

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE ELECTRÓNICA



**"SISTEMA DE INSPECCION BASADO EN VISION
ARTIFICIAL APLICADO A UN PROCESO INDUSTRIAL"**

ALUMNOS:

PEDRO ERNESTO ORTEGA MARROQUIN
EMERSON FRANCISCO POCASANGRE LEON
FERNANDO ANTONIO VALLE FIGUEROA

ASESOR:

ING. EDGARDO CRUZ ZELEDÓN

TUTOR:

ING. OSCAR WENCESLAO RIVAS

CIUDADELA DON BOSCO, OCTUBRE DE 2004.

TABLA DE CONTENIDO

<i>TABLA DE CONTENIDO</i>	2
<i>INTRODUCCION</i>	4
<i>1. GENERALIDADES DE UN SISTEMA DE VISION ARTIFICIAL</i>	6
1.1.1 INTRODUCCIÓN	6
1.1.2 ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL	7
1.1.3 APLICACIONES DE UN SISTEMA DE VISION ARTIFICIAL	8
<i>1.2 ADQUISICION DE LA IMAGEN</i>	9
1.2.1 GENERALIDADES	9
<i>1.3 TECNICAS DE ILUMINACION</i>	10
1.3.1 ILUMINACIÓN DIRECCIONAL	10
1.3.2 ILUMINACIÓN DIFUSA	11
1.3.3 ILUMINACIÓN A CONTRA LUZ	12
1.3.4 ILUMINACIÓN ESTRUCTURADA	13
1.3.5 ILUMINACIÓN OBLICUA	14
<i>1.4 RESULTADOS OBTENIDOS SOBRE ILUMINACIÓN</i>	14
<i>1.5 TOMA DE IMAGEN DESDE MATLAB</i>	17
<i>2. PREPROCESAMIENTO DE IMAGEN</i>	19
2.1 INTRODUCCION	19
2.2 ELIMINACION DE RUIDO	21
2.3 FILTROS NO LINEALES	21
2.4 EXTRACCION DE CARACTERÍSTICAS	24
2.5 BORDES	26
2.5.1 PERFILADO	26
2.5.2 EL OPERADOR GRADIENTE	27
2.6 AJUSTE DE LA IMAGEN	30
2.7 BINARIZACION DE LA IMAGEN	32
2.8 CONTINUIDAD A PARTIR DEL PREPROCESAMIENTO	33
2.9 BASES DE DISEÑO DEL SISTEMA DE INSPECCIÓN POR VISIÓN ARTIFICIAL	34
2.10 FLUJOGRAMA DE LA SECUENCIA DE CAPTURA Y PREPROCESAMIENTO	36
<i>3. SEGMENTACION DE IMÁGENES</i>	37
3.1 OBTENCIÓN DE CARACTERISTICAS DE UNA IMAGEN BINARIA	37
3.2 REDES NEURONALES ARTIFICIALES	41
<i>4. CLASIFICACION DE LA IMAGEN</i>	51
4.1 TEORIA DE LA RESONANCIA ADAPTATIVA (ART1s)	51
4.2 ARQUITECTURA Y FUNCIONAMIENTO	52
4.3. ENTRENAMIENTO DE LA RED ART1S PARA EL SISTEMA DE INSPECCIÓN BASADO EN VISION ARTIFICIAL	60

4.4.FUNCIONES UTILIZADAS POR EL ALGORITMO PRINCIPAL.	67
4.5 MEDICIONES DE DESEMPEÑO REAL DEL SISTEMA	70
<i>ANEXO A</i>	71
A.1 MANUAL DE USUARIO.....	71
<i>ANEXO B</i>	77
B.1 DESCRIPCIÓN Y MANEJO DEL TOOLBOX.	77
<i>ANEXO C</i>	82
C.1 EL PUERTO PARALELO.....	82
<i>ANEXO D</i>	86
D.1 Hardware Utilizado (Diagramas del prototipo)	86
<i>CONCLUSIONES</i>	94
<i>ALCANCES Y RECOMENDACIONES PARA SUPERAR LAS LIMITANTES</i> ..	94
<i>LIMITACIONES</i>	95
<i>REFERENCIAS BIBLIOGRAFICAS</i>	96
<i>REFERENCIAS DE INTERNET</i>	96

INTRODUCCION

En la actualidad la exigencia del sector de consumo ha crecido grandemente en cuanto al nivel de calidad de los productos, es debido a esto que las industrias han decidido modernizarse en cuanto a sus niveles de control de calidad. Y a sabiendas de que debido a que la mano de obra es afectada por diferentes factores (cansancio, temperatura, etc.) su confiabilidad no es óptima, es que se presentó la necesidad de utilizar tecnología para realizar el análisis de los productos finales, esta tecnología se denomina visión artificial y se inicio en la década de 1960, desde entonces ha avanzado grandemente hasta convertirse en nuestros días en la herramienta más utilizada en el control de calidad final de los productos.

Para las relaciones de intercambio comercial que son tan comunes hoy en día, existen requerimientos respecto a la calidad del envasado de los productos alimenticios; por cuanto en una búsqueda por brindar una solución económicamente viable en la línea de envasado de alimentos en vidrio, se ha optado por la realización de un “sistema de inspección” que pueda realizar la detección de anomalías como son rajaduras y cuerpos extraños en dichos frascos.

Esta inspección debe realizarse de tal manera que los resultados no puedan verse interferidos por los múltiples factores que existen en un ambiente industrial. Para la elaboración del sistema de inspección, se ha optado por la utilización de técnicas de visión artificial.

En el capítulo uno se presentan las generalidades de un sistema de visión artificial, y otros factores de importancia en donde se involucra la captura de una imagen y las técnicas de iluminación adecuadas para obtenerla.

La detección de defectos superficiales es de suma importancia, y para realizarla, será necesaria la implementación de algoritmos que sean capaces de ejecutar dicha función en el menor tiempo posible puesto que el sistema debe ser dinámico dentro del proceso.

El capítulo dos trata sobre el preprocesamiento que debe darse a una imagen para su posterior clasificación, es decir para que se convierta en información útil para un sistema digital de análisis de imágenes.

En el tercer capítulo se presentan los algoritmos utilizados para la extracción de características en una imagen binaria, así como las generalidades de las Redes Neuronales.

Finalmente en el último capítulo se profundiza acerca de la Red Neuronal utilizada (ART1S), el porque la utilización de esta red, su funcionamiento, su entrenamiento. Además el Algoritmo principal y el rendimiento del sistema.

CAPITULO 1

1. GENERALIDADES DE UN SISTEMA DE VISION ARTIFICIAL

1.1.1 INTRODUCCIÓN

La visión humana es crítica en muchos puestos de trabajo ya que la precisión del ojo humano es inalcanzable para cualquier sistema de visión artificial. Sin embargo, el ojo humano está sujeto a fuentes de error que disminuyen notablemente su eficacia en el proceso productivo de la empresa. La fatiga visual y la monotonía de las tareas de verificación son los puntos débiles del ser humano en el proceso productivo.

Los sistemas de visión artificial son inmunes a la fatiga visual y a la monotonía de las tareas, rindiendo tanto más cuanto más monótonas sean las tareas que supervisa. La visión artificial controlada por computadora tiene innumerables aplicaciones para el control de procesos industriales con una considerable mejora de la eficacia productiva.

La Visión Artificial es una disciplina que tiene como finalidad, la extracción de información del entorno físico a partir de imágenes, utilizando para ello una computadora. Se trata de un objetivo ambicioso y complejo que actualmente se encuentra en una etapa de desarrollo.

Un sistema de Visión Artificial actúa sobre una representación de una realidad que le proporciona información sobre brillo, colores, formas, etcétera. Estas representaciones suelen estar en forma de imágenes estáticas, escenas tridimensionales o imágenes en movimiento.

1.1.2 ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL

La Visión Artificial, puede ser dividida en seis áreas principales:

- Adquisición de Imagen: Es el proceso a través del cual se obtiene una imagen visual.
- Preprocesamiento: Incluye técnicas tales como la reducción de ruido y realce de detalles. Aquí se incluyen las técnicas de Detección de bordes. Además de filtros y transformaciones geométricas, que eliminan partes indeseables de la imagen o se realzan partes interesantes de la misma.
- Segmentación: Es el proceso que divide a una imagen en objetos que sean de utilidad y consiste en aislar los elementos que interesan de una escena para comprenderla
- Descripción: Es el proceso mediante el cual se obtienen características convenientes para diferenciar un tipo de objeto de otro, por ejemplo tamaño y forma.
- Reconocimiento: Es el proceso que identifica los objetos de una escena, por ejemplo diferentes tipos de piezas en un tablero de juego.
- Interpretación: Es el proceso que asocia un significado a un conjunto de objetos reconocidos.

El objetivo de un Sistema de Visión Artificial, es crear un modelo del mundo real a partir de imágenes.

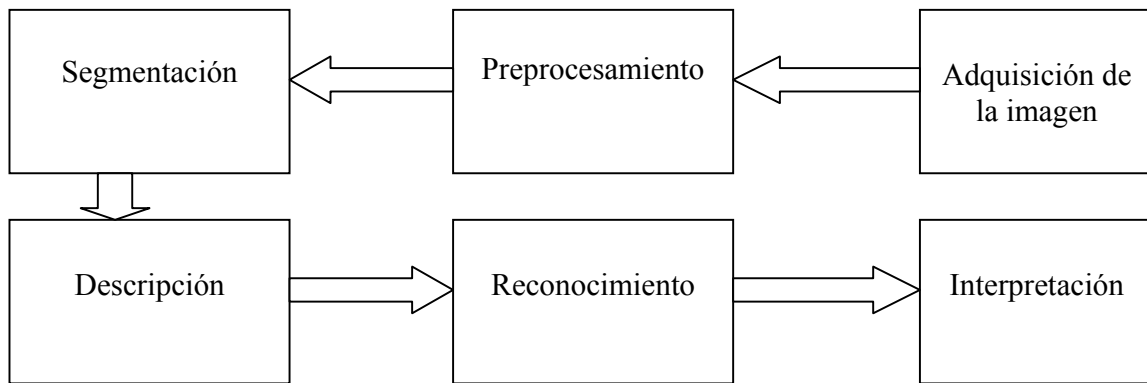


Figura 1.1 Etapas de un sistema de visión artificial

1.1.3 APLICACIONES DE UN SISTEMA DE VISION ARTIFICIAL

El amplio espectro de aplicaciones cubierto por la visión artificial, se debe a que permite extraer y analizar información espectral, espacial y temporal de los distintos objetos.

La información espectral incluye frecuencia (color) e intensidad (tonos de gris). La información espacial se refiere a aspectos como forma y posición (una, dos y tres dimensiones). La información temporal comprende aspectos estacionarios (presencia y/o ausencia) y dependientes del tiempo (eventos, movimientos, procesos).

Según el tipo de aplicación, así serán el tipo de imagen que será necesario adquirir (imágenes de rayos X, infrarrojo, etc.) y el análisis que se aplicará. La mayoría de las aplicaciones de la visión artificial pueden clasificarse por el tipo de tarea en inspección (medición, calibración, detección de fallas), verificación, reconocimiento, identificación y análisis de localización (posición, guía).

La **detección de fallas** es un análisis cualitativo que involucra la detección de defectos o artefactos no deseados, con forma desconocida en una posición desconocida. Por ejemplo, encontrar rajaduras en frascos de vidrio, hallar defectos en la pintura de un auto nuevo, o agujeros en hojas de papel, etc.

La **medición** o **calibración** se refiere a la correlación cuantitativa con los datos del diseño, asegurando que las mediciones cumplan con las especificaciones del diseño. Por ejemplo, el revisar que un cable tenga el espesor recomendado.

La **verificación** es el chequeo cualitativo de que una operación de ensamblaje ha sido llevada a cabo correctamente. Por ejemplo, que no falte ninguna tecla en un teclado, o que no falten componentes en un circuito impreso.

El **reconocimiento** involucra la identificación de un objeto con base en descriptores asociados con el objeto. Por ejemplo, la clasificación de cítricos (limones, naranjas, mandarinas, etc.) por color y tamaño. Otro ejemplo de reconocimiento podría ser aplicado a células, por área y forma.

Identificación es el proceso de identificar un objeto por el uso de símbolos en el mismo. Por ejemplo, el código de barras.

El **análisis de localización** es la evaluación de la posición de un objeto. Por ejemplo, determinar la posición donde debe insertarse un circuito integrado ("chip").

Guía significa proporcionar adaptativamente información posicional de retroalimentación para dirigir una actividad. El ejemplo típico es el uso de un sistema de visión para guiar un brazo robot mientras suelda o manipula partes.

1.2 ADQUISICION DE LA IMAGEN

1.2.1 GENERALIDADES

En un sistema de inspección visual la primera tarea a realizar es la captura de la escena a analizar, para luego ser almacenada en memoria; para cumplir dicha función en el presente proyecto se utilizará el siguiente dispositivo electrónico:

WebCam Intel CS-430

Resolución mínima de 160 x 120 píxeles.

Resolución máxima de 640 x 480 píxeles.

Una correcta adquisición de la imagen supone un paso muy importante para que el proceso de reconocimiento tenga éxito. Dentro de esta etapa se han encontrado múltiples factores que afectan directamente al proceso de captura de la imagen, formados fundamentalmente por el sistema de visión artificial (cámara, óptica utilizada, computadora y software de desarrollo) y el entorno y posicionamiento de los elementos (la iluminación, el fondo, posición correcta de la cámara, ruido eléctrico-óptico externo, etc.).

1.3 TECNICAS DE ILUMINACION

La iluminación juega un papel importante dentro de la visión artificial, pues simplifica de manera considerable el análisis y posterior interpretación de la escena captada, es un factor que suele afectar de forma considerable a la complejidad de los algoritmos de visión. La iluminación existente en el entorno no suele ser aceptable ya que se obtienen imágenes con bajo contraste, sombras, etc.

El tipo de iluminación dependerá de cada caso concreto, resultando difícil generalizar, sin embargo, pueden destacarse los diferentes tipos de iluminación, así como de fuentes de luz que a continuación se analizarán:

1.3.1 ILUMINACIÓN DIRECCIONAL

Consiste en aplicar una iluminación orientada al objeto usando un haz altamente direccional (en el caso de un haz láser), y medir el grado de dispersión.

Se utiliza en localización y reconocimiento de piezas, inspección de la superficie de los objetos, etc. La forma del haz luminoso depende de la aplicación y del objeto a analizar.

Se pueden destacar los siguientes casos:

- Punto luminoso
- Plano de luz
- Corona circular luminosa

En la figura 1.2 se representa un ejemplo de iluminación mediante planos de luz. Este tipo de iluminación es de fácil uso pero presenta principalmente como problema la aparición de sombras que pueden oscurecer atributos del objeto iluminado.

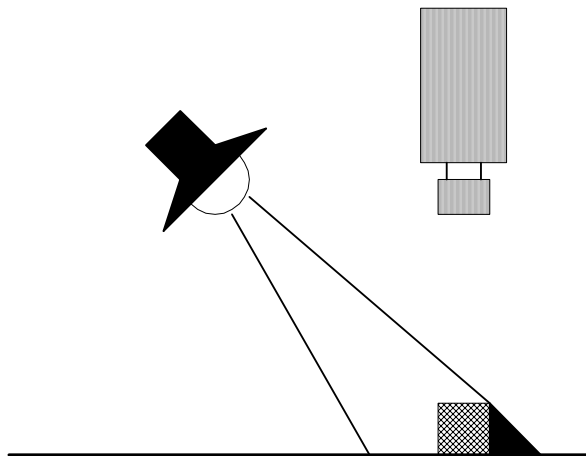


Figura 1.2 Iluminación mediante planos de luz

1.3.2 ILUMINACIÓN DIFUSA

En ella se intenta que los haces luminosos incidan sobre el objeto desde todas las direcciones y no sólo desde la fuente luminosa. Se utiliza cuando los objetos a analizar presentan superficies suaves y regulares.

Esta forma de iluminación se suele emplear en aplicaciones donde las características de la superficie son importantes. Este tipo de iluminación no proporciona el máximo contraste del objeto.

Como fuentes de luz que se pueden utilizar para obtener este tipo de iluminación tenemos el fluorescente y las fuentes halógenas con haces de fibra óptica.

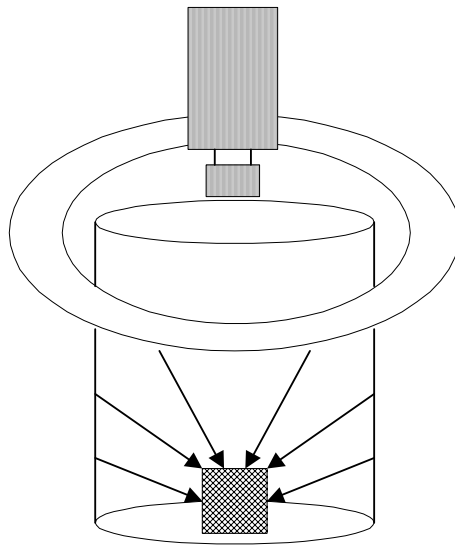


Figura 1.3 Iluminación difusa

1.3.3 ILUMINACIÓN A CONTRA LUZ

Consiste en iluminar el objeto por detrás de forma que la fuente luminosa, el objeto y la cámara formen una línea recta. Produce imágenes con sólo dos niveles de gris (imágenes en blanco y negro).

Esta técnica se adapta muy bien en aplicaciones para la localización de piezas y análisis dimensional, en las cuales la silueta de los objetos es suficiente para su reconocimiento.

El principal inconveniente es la pérdida de todos los detalles, así como la imposibilidad de detectar fisuras en el objeto. En la figura 1.4 se muestra un ejemplo del uso de este tipo de iluminación.

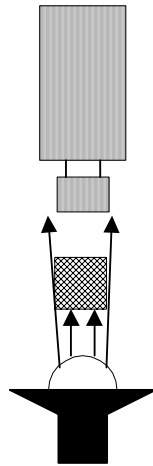


Figura 1.4 Iluminación a contra luz

1.3.4 ILUMINACIÓN ESTRUCTURADA

Este tipo de iluminación se sirve de la proyección de puntos, franjas, o rejillas sobre la superficie de trabajo.

Posee dos ventajas con respecto a los anteriores tipos de iluminación:

- En primer lugar establece un patrón de luz conocido sobre la superficie de trabajo y las diferencias con este patrón indican la presencia de un objeto, simplificándose así el problema de detección de un objeto.
- En segundo lugar, analizando la forma en que el patrón de luz es distorsionado, es posible obtener información de las características tridimensionales del objeto.

1.3.5 ILUMINACIÓN OBLICUA

La iluminación oblicua creará sombras encargadas de aumentar el contraste de las partes tridimensionales. Este tipo de iluminación es utilizada para generar sombras sobre objetos cuyo contraste es pequeño respecto del fondo. La figura 1.5 muestra una posible aplicación de este tipo de iluminación.

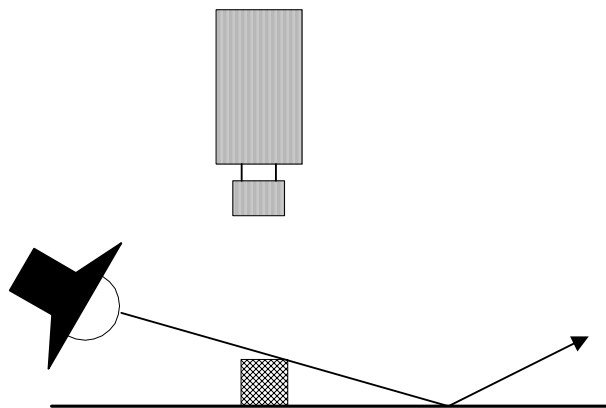


Figura 1.5 Iluminación Oblicua

1.4 RESULTADOS OBTENIDOS SOBRE ILUMINACIÓN

Para una buena captura de la escena es necesario hacer uso de una apropiada técnica de iluminación; ya que de ésta depende la calidad con la que es tomada la imagen.

En las primeras pruebas de captura de imagen que se realizaron se utilizaron las siguientes *técnicas de iluminación*:

Iluminación Direccional: para este tipo de iluminación, la imagen captada por la cámara presentó sombras que se mezclaban con la imagen binarizada, por lo cual generaba información que no es útil para los propósitos de reconocimiento de defectos en

envases de vidrio, pues dicha información adicional pudiera ser tomada por el sistema como un defecto.

La *figura 1.6* muestra una imagen de un envase de vidrio captada por el sistema utilizando iluminación direccional.



Figura 1.6 Imagen tomada utilizando técnica de iluminación direccional

Iluminación Difusa: cuando se hicieron pruebas con esta técnica se introdujo el envase de vidrio a un espacio donde las paredes permitieron reflejar la luz de la fuente hacia el envase, recibiendo luz desde varias direcciones, el inconveniente al capturar la imagen son los reflejos generados por el mismo envase, que tienen tal brillantez que no es posible detectar defectos minúsculos en la superficie del envase.

La *figura 1.7* muestra una imagen cuando el envase es iluminado de forma difusa.



Figura 1.7 escena captada usando iluminación difusa

Iluminación a contra luz: usando esta técnica se obtuvieron resultados que de alguna manera dieron indicios de que una iluminación posterior permitía presentar detalles en el envase que no eran captados cuando se utilizaban las técnicas anteriores, puesto que los detalles del envase eran más evidentes en las zonas donde había mayor cantidad de luz, sin embargo existieron zonas que debido a la discrepancia en la iluminación no se pudieron observar tales detalles.

En la *figura 1.8* se observan las características mencionadas anteriormente en el envase.



Figura 1.8 imagen tomada con iluminación a contra luz

La técnica de iluminación más apropiada para la extracción de todas las características de un envase de vidrio es la *iluminación posterior difusa*, dicha técnica permite resaltar los defectos presentes en un envase de vidrio para luego tomarlos y someterlos a un análisis por parte de la aplicación.

Como prueba a lo anterior se presenta la *figura 1.9* donde se observa el envase con un gran porcentaje de sus características.



Figura 1.9 imagen captada con la técnica de iluminación posterior difusa

1.5 TOMA DE IMAGEN DESDE MATLAB

Al iniciar con el proceso de investigación de la mejor aplicación para manipular archivos de imagen, entre las opciones que se han tomado en cuenta son: “VISUAL BASIC” y “MATLAB”. Sin embargo, luego de evaluar las diferentes funciones y atributos de cada aplicación, relacionados con el control y diversificación de parámetros en una imagen cualquiera que sea, se ha decidido darle prioridad a la aplicación “MATLAB”, y se decidió descartar a “VISUAL BASIC”, debido a la mayor dificultad que presenta; pues no cuenta con las suficientes herramientas implícitas comparado con la otra aplicación antes mencionada.

Para la captura de imágenes desde el entorno de programación de Matlab es necesario un algoritmo que relacione el dispositivo de toma de imagen (webcam) y almacene esta información como una matriz en formato digital.

Investigando sobre las posibles maneras de obtener una imagen con un algoritmo propio de Matlab se encontró una solución que puede utilizar el driver de una cámara, obtener la información que esta recibe y poder ser operada por Matlab, el nombre de este algoritmo es Visión for Matlab por sus siglas en inglés *VFM*².

² Ver anexo B acerca del toolbox *Vision for Matlab (VFM)*

Entre las primeras pruebas que se realizaron con VFM estuvo la toma de una imagen con todas sus propiedades, es decir de acuerdo a la capacidad de la cámara utilizada, que en este caso es de 256 colores. A continuación se muestra el algoritmo de captura básica de la imagen así como la imagen obtenida en la *figura 1.10*.

```

%Programa para la captura de imagen utilizando VFM
%vfm puede ser utilizado dentro de un imwrite para
%indicar la fuente desde la cual sera tomada la informacion
%y posteriormente le asigna el nombre con el cual sera
%almacenada en forma de archivo. Luego se hace la presentacion
%de la imagen con el comando imshow
close all
imwrite(vfm('grab',1), 'imagen_prueba.jpg','jpg');
figure(1)
imshow('imagen_prueba')

```



Figura 1.10 Algoritmo de captura de imagen y la imagen obtenida

CAPITULO 2

2. PREPROCESAMIENTO DE IMAGEN

2.1 INTRODUCCION

El procesamiento de imágenes es un conjunto de algoritmos cuya finalidad es conseguir una mejora en la apariencia de la imagen original. Esta mejora en la apariencia consistirá en que determinadas características de la imagen queden resaltadas o en eliminar aquello que las oculta. Por razones históricas esta etapa de la visión artificial se inicia en un campo relacionado con ella, el procesamiento o tratamiento de imágenes. Sin embargo aquí el enfoque será distinto ya que lo que se pretende es que el análisis último sea llevado a cabo por una PC por lo que la imagen resultante puede ser más confusa ante el operador que la inicial no siendo así para el computador y viceversa. No se trata por tanto de mejorar la imagen de manera que quede lo más ideal posible sino que pueda ser analizada de la forma más simple. Es por tanto una etapa orientada hacia el problema específico que se quiera solucionar por lo que los algoritmos que funcionen bien en unos casos no lo harán en otros.

Los algoritmos pueden dividirse en dos grandes grupos según se trabaje el dominio del espacio o de la frecuencia. En el primero se trabajará directamente con los píxeles de la imagen. Serán algoritmos locales que transformarán o bien el valor de cada píxel tomado individualmente o el de un pequeño conjunto de ellos. El segundo caso, basado en la frecuencia, cambiará la transformada de Fourier de la imagen. Ambos enfoques no se excluyen mutuamente y de hecho en algunas aplicaciones debe usarse una combinación de ellos.

Se empezará viendo las transformaciones que cambian el contraste de una imagen. Cada píxel modifica su valor en función del nivel de gris que tenga, independientemente del valor del resto de los píxeles de la imagen. El segundo grupo se basa en la modificación del histograma ya que de él puede desprenderse si la imagen es clara u oscura. Otro problema que se estudia es la eliminación de ruido, que siempre se

hallará presente en una imagen. Se verán técnicas en el dominio del espacio y en el dominio de la frecuencia. Ambos enfoques se usarán también para el realce de bordes de la imagen, en donde se trata de resaltar aquellos puntos donde existe un cambio de niveles de gris.

Debido a que es necesario luego de realizar la captura de la imagen, iniciar con el proceso de tratamiento y obtención de las características útiles y minimizar la cantidad de información que será procesada por el sistema a fin de agilizarlo, se iniciará con la investigación de las técnicas más apropiadas para realizar este objetivo.

De la necesidad de convertir la imagen RGB a niveles de gris.

- Se reduce la cantidad de información, pues una imagen RGB ocupa una matriz de $M \times N \times 3$, mientras que una imagen en niveles de gris solamente ocupa una de $M \times N$.
- La aplicación de reconocimiento de anomalías (defectos) en envases de vidrio, se reduce al análisis de existencia de éstas, por lo que no se requiere una clasificación de los defectos, sino solamente su identificación, por lo que una imagen en escala de grises es suficiente para el proceso.
- El tipo de filtros necesarios para simplificar una imagen tiene la limitante de permitir únicamente imagen $M \times N$, es decir dos dimensiones por lo que debe reconfigurarse el formato de RGB a Grises.

Se realizaron las siguientes pruebas de conversión a escala de grises con algoritmos que a continuación se presentan:

```
%Conversion de una imagen RGB a escala de grises.  
% se incluye la funcion VFM para la captura de la imagen  
%Tambien se utilizan otros comandos para conversion de imagen a matriz  
%Compatible y procesable por MATLAB  
vfm('preview',1);  
imwrite(vfm('grab',1),'Imagen.tif','tif');  
A=imread('Imagen.tif');  
B=rgb2gray(A);  
figure(1)  
imshow(B)
```

Fig.2.1 Algoritmo para la captura y conversión de una imagen a escala de grises.



Fig. 2.2 Imagen RGB y su equivalente en escala de grises.

2.2 ELIMINACION DE RUIDO

Todas las imágenes tienen una cierta cantidad de ruido, valores distorsionados, bien debidos al sensor de la cámara o al medio de transmisión de la señal. El ruido consiste generalmente en píxeles aislados que toman un valor de gris diferente del de sus vecinos. Las maneras de filtrado que se estudiarán a continuación se basan en esta característica. El ruido puede clasificarse en dos tipos: gaussiano e impulsional. En el primer caso el valor final del píxel sería el ideal mas una cantidad correspondiente al error, pudiéndose describir este error como una variable gaussiana. En el ruido impulsional el valor que toma el píxel no tiene relación con el valor ideal sino con el valor del ruido que toma valores altos.

2.3 FILTROS NO LINEALES

Outlier

Cada píxel es comparado con la media de sus ocho vecinos, si esta diferencia es superior a un valor preestablecido se considera ruido y se sustituye por el valor de esa media.

Mediana

En una secuencia impar de números $x_1 \dots x_N$, la mediana es aquel valor que cumple que $(N-1)/2$ elementos tiene un valor menor o igual y que $(N-1)/2$ tiene un valor mayor o igual. En el caso de imágenes se pueden aplicar dos soluciones:

- Definir una ventanas de dimensión $N \times 1$ y se aplica el mismo criterio
- Definir ventanas de dimensión $N \times N$ y elegir la mediana entre ellas.

Dependiendo del ruido o distorsión que posea la imagen se aplica un tipo de filtro que mejor soluciona este problema, por ejemplo:

- El tipo de ruido sal y pimienta es eliminado de mejor manera por el filtro de mediana.

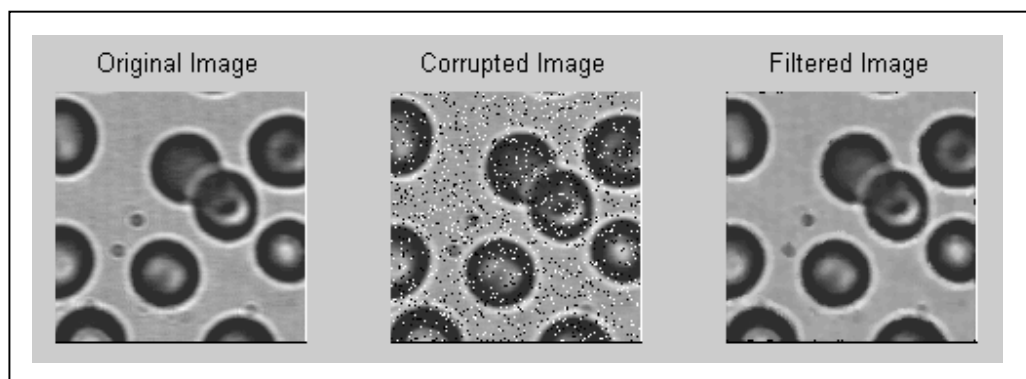


Fig. 2.3 Imagen con ruido tipo Salt & Pepper filtrada por el método de mediana.

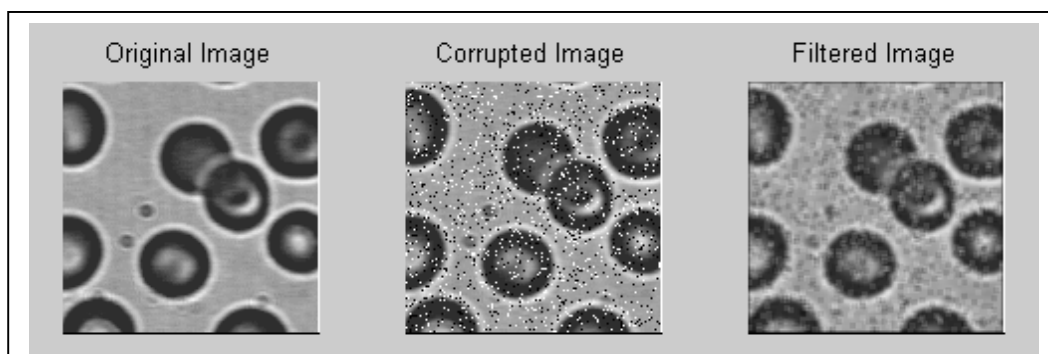


Fig. 2.4 Imagen con ruido Salt & Pepper tratada con filtro tipo averaging.

- Al tener un tipo de ruido Gaussiano es conveniente según las pruebas realizadas, utilizar un filtro Averaging como mejor opción para eliminarlo.

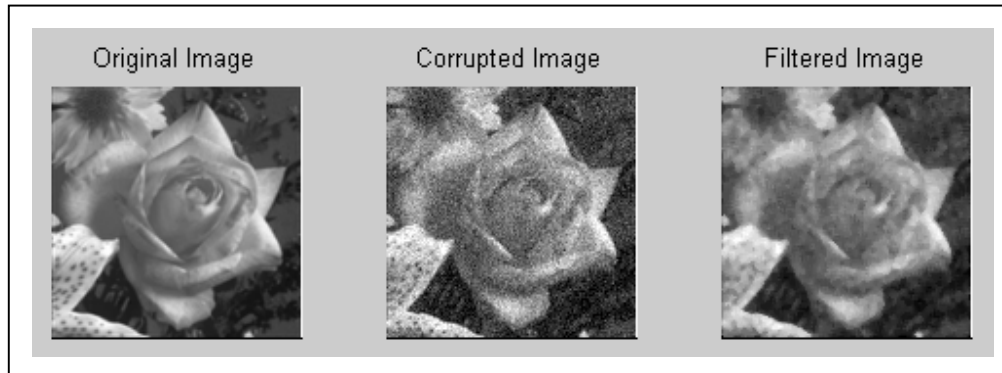


Fig. 2.5 Imagen con ruido Gaussiano procesada con un filtro de Mediana.

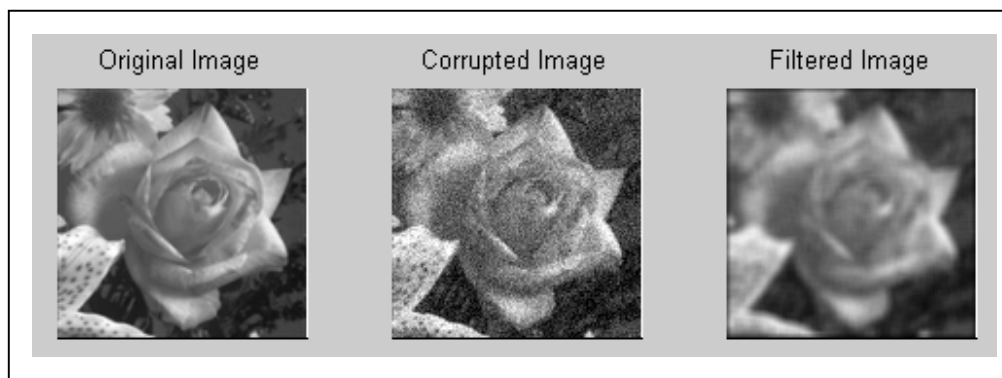


Fig. 2.6 Imagen con ruido Gaussiano filtrada mediante Averaging.

Se presenta a continuación el algoritmo utilizado para efectuar la captura de la imagen, la transformación a escala de grises y el filtrado utilizando el método de mediana.

Dentro del método de mediana es necesario especificar una matriz o vecindario con el fin de aumentar el nivel de filtrado.

```

%Conversion de una imagen RGB a escala de grises.
% se incluye la funcion VFM para la captura de la imagen
%Tambien se utilizan otros comandos para conversion de imagen a matriz
%Compatible y procesable por MATLAB

vfm('preview',1);

A=0;
B=0;
C=0;
inwrite(vfm('grab',1),'Imagen1.tif','tif');
A=imread('Imagen1.tif');
B=rgb2gray(A);
C=medfilt2(B,[4 4]);
figure(1)
imshow(B)
figure(2)
imshow(C)

```

Fig. 2.7 Algoritmo para tratamiento de imagen con filtro de mediana.

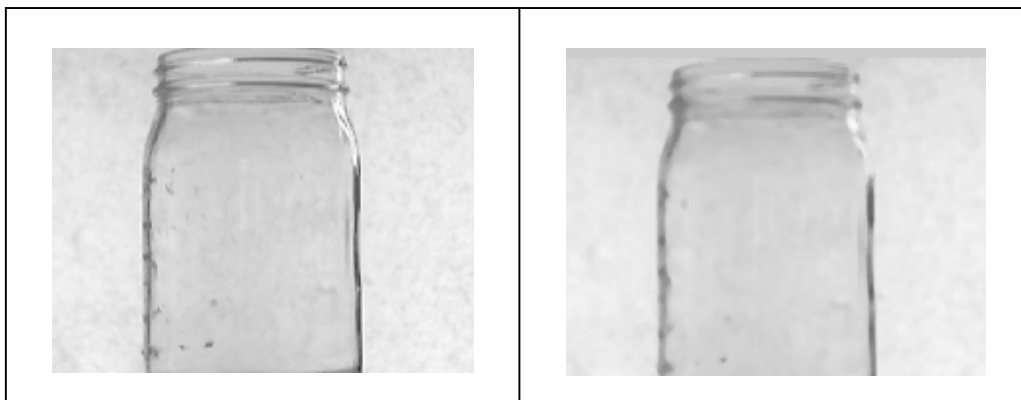


Fig. 2.8 Imagen normal y filtrada por el método de mediana.

2.4 EXTRACCION DE CARACTERÍSTICAS

Una de las ideas mas importantes del procesamiento de imágenes es la extracción de características de la imagen, con la intención de describir, interpretar o comprender la escena que se está recogiendo.

Los sistemas de visión más sofisticados son capaces de interpretar los resultados del análisis efectuado y describir los distintos objetos encontrados, así como su relación dentro de la escena. El análisis de la imagen está principalmente vinculado al estudio de las características obtenidas, la segmentación de la imagen y finalmente su clasificación.

En un sistema de visión artificial, la imagen captada es primero preprocesada con el fin de mejorarla, pudiendo tratar su nivel de ruido, resolución, contraste, textura, color, etc. Es entonces cuando se podrán extraer algunas características con el fin de poder segmentar la imagen en sus componentes, por ejemplo, separar los diferentes objetos mediante la extracción de sus bordes.

La imagen segmentada será tratada por un clasificador o un sistema de interpretación cuya misión será determinar la relación entre los diferentes objetos en una escena con la intención de proporcionar su descripción.

Las características espaciales de un objeto deben venir dadas por sus niveles de gris, su distribución conjunta de probabilidad y su distribución espacial.

Lo normalmente empleado en la mayoría de los casos son los valores de las propiedades físicas del objeto tales como su capacidad de reflexión, transmisión, color, espectro, etc.

Puesto que parece que uno de los principales objetivos de un sistema de visión es clasificar los objetos individuales presentes en una imagen, parece lógico delimitar la frontera (el contorno) de los objetos como paso obligado para el cálculo de los atributos determinantes.

2.5 BORDES

Uno de los problemas más importantes que se encontró en el análisis de imágenes es la detección de bordes. Los bordes van a definir los límites del objeto, pudiendo ser utilizados posteriormente en tareas de segmentación, registro e identificación de objetos en la escena.

Las técnicas empleadas en la detección de bordes tienen por objeto la determinación de los puntos en que se produce una variación de intensidad, empleándose para ello métodos basados en los operadores derivada.

Estas técnicas logran realzar los contornos de los objetos que aparecen en una imagen, dando mayor nitidez a estos objetos y creando una imagen más agradable para el observador.

Además realzando exageradamente los bordes (perfilado), se puede conseguir que un procesamiento posterior de la imagen, normalmente una detección de umbral, lleve a la obtención de una imagen en la que sólo aparezcan los contornos presentes en la imagen primitiva. El principal inconveniente de esta técnica es la inclusión de ruido o reforzamiento del ya existente en la imagen original.

2.5.1 PERFILADO

Se emplean técnicas basadas en el cálculo de gradientes o bien en los operadores laplacianos para efectuar el realce y la extracción de contornos.

Para poder utilizar operadores de este tipo sobre una imagen muestreada es necesario obtener una aproximación del concepto de derivada para espacios discretos. La generalización comúnmente usada se basa en el cálculo de diferencias entre píxeles vecinos. Estas diferencias, según la relación entre los píxeles considerados, pueden dar

lugar a derivadas unidimensionales o bidimensionales, así como aplicarse en alguna dirección determinada de la imagen o en todas las direcciones de forma global.

Existen otras aproximaciones diferenciales que han probado ser de gran utilidad para la extracción de contornos, es el caso de la de Roberts, la de Sobel, Canny, etc.

2.5.2 EL OPERADOR GRADIENTE

El operador gradiente G aplicado sobre una imagen f(x, y) puede definirse como:

$$G[f(x, y)] = |G_x| + |G_y| \quad \text{Ec. 2.1}$$

Siendo:

$$G_x = \frac{\delta \cdot f}{\delta \cdot x} = f(x, y) - f(x-1, y) \quad \text{Ec. 2.2}$$

$$G_y = \frac{\delta \cdot f}{\delta \cdot y} = f(x, y) - f(x, y-1) \quad \text{Ec. 2.3}$$

La representación grafica del vector gradiente a lo largo de una dirección r correspondiente al borde de un objeto es la mostrada en la figura 2.9.

El operador gradiente puede representarse por un par de mascarar H₁ y H₂ correspondientes a los ejes x, y respectivamente:

$$H_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Tabla 2.1. Operador gradiente mascara H₁

$$H_2 = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

Tabla 2.2. Operador gradiente máscara H₂

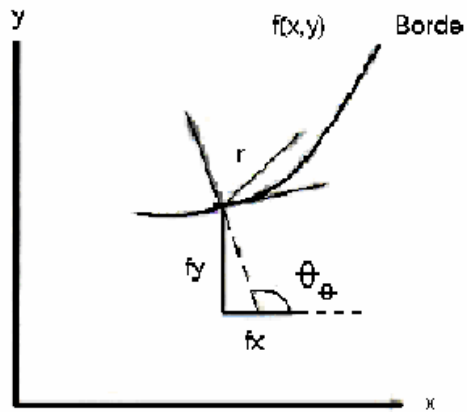


Fig. 2.9 Vector Gradiente

A continuación se mostrará una lista de los operadores gradiente mas utilizados:

- Gradiente de Roberts.

-1	1
1	-1

Tabla 2.3. Gradiente de Roberts.

- Detector Horizontal.

-1	-1	-1
0	0	0
1	1	1

Tabla 2.4. Detector Horizontal.

- Detector Vertical

-1	0	1
-1	0	1
-1	0	1

Tabla 2.5. Detector Vertical

- Operador de Prewitt. Es una combinación de los dos anteriores al computarse las sumas de las diferencias locales.

- Operador de Sobel. Es una combinación de:

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

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

Tabla 2.6. Operador de Sobel

Ambas mascarar, computando las diferencias de las sumas locales de ambas, lo cual reduce el efecto de ruido en la imagen, al igual que en el operador de Prewitt.

Este operador proporcionará un valor nulo en regiones uniformes.

- Operador Laplaciano. Suele aplicarse en aquellos casos en los cuales las Transiciones de los niveles de gris son bastante abruptas.

La definición del operador Laplaciano que de forma más frecuente se utiliza es:

$$L[f(x, y)] = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \quad \text{Ec. 2.4}$$

-1	-1	-1
-1	8	-1
-1	-1	-1

Tabla 2.7. Operador Laplaciano.

- Operador Isotrópico.

-1	0	1
-√2	0	√2
-1	0	1

-1	-√2	-1
0	0	0
1	√2	1

Tabla 2.8. Operador Isotrópico.

- El operador Compass (brújula). Permite realizar una medida del gradiente en las direcciones seleccionadas.

A continuación se presenta una imagen y la respectiva extracción de sus bordes

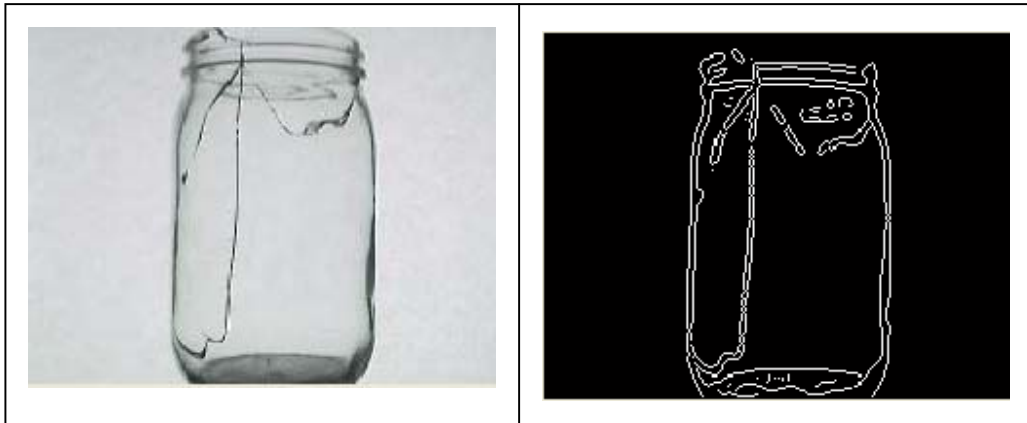


Fig. 2.10. Imagen original y sus bordes usando el método de Canny.

2.6 AJUSTE DE LA IMAGEN

Cuando se ha realizado el proceso de filtrado de la imagen es necesario mejorarla mucho más que el resultado obtenido, pues esta necesita que exista uniformidad en cuanto a brillantez de los píxeles en las diferentes regiones que componen la imagen y muchas veces el proceso de filtrado no proporciona la calidad para que una imagen se traslade por métodos de binarización.

Es necesario que la imagen cuente con rasgos definidos, pues a la hora de binarizar, si es que no se mantienen los niveles de iluminación promedio en un área resulta una binarización con sectores definidos de manera no adecuada para los propósitos de análisis y exclusión de los envases de vidrio.

```

%Conversion de una imagen RGB a escala de grises.
% se incluye la funcion VFM para la captura de la imagen
%Tambien se utilizan otros comandos para conversion de imagen a matriz
%Compatible y procesable por MATLAB

vfm('preview',1);

A=0;
B=0;
C=0;
D=0;
imwrite(vfm('grab',1),'Imagen1.tif','tif');
A=imread('Imagen1.tif');
B=rgb2gray(A);
C=medfilt2(B,[4 4]);
D=imadjust(A,[0.3 0.7],[]);
figure(1)
imshow(B)
figure(2)
imshow(C)
figure(3)
imshow(D)

```

Fig. 2.11. Algoritmo que contiene comandos de ajuste de imagen.

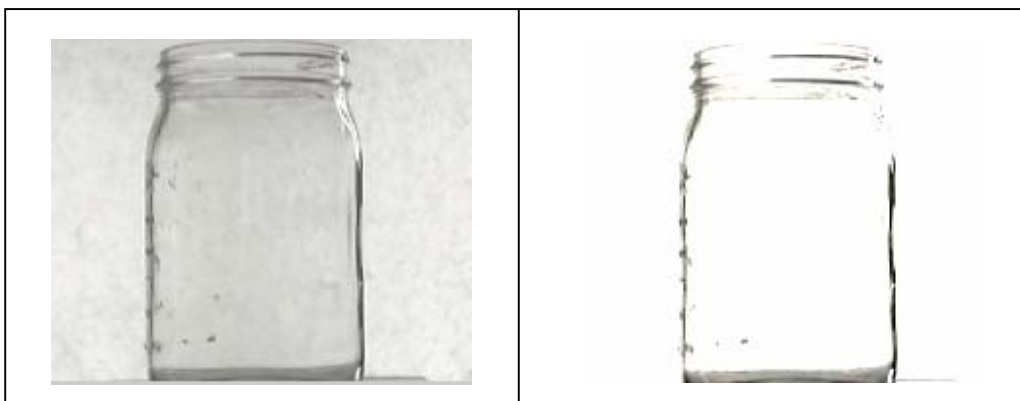


Fig. 2.12 Imagen luego de aplicarle un ajuste.

2.7 BINARIZACION DE LA IMAGEN.

Cuando la imagen se ha hecho pasar por los distintos procedimientos de preprocesamiento, finalmente se debe llevar a un punto en el cual exista la posibilidad de obtener solamente las características básicas para definir si existe o no alguna anomalía dentro del envase. Se hace necesario entonces aplicar dentro del algoritmo comandos que sean capaces de dar como resultado una imagen en dos niveles de gris, máximo y mínimo (blanco y negro).

A continuación se presenta el algoritmo que realiza la función y da como resultado una imagen en blanco y negro.

```
%Conversion de una imagen RGB a escala de grises.
% se incluye la funcion VFM para la captura de la imagen
%Tambien se utilizan otros comandos para conversion de imagen a matriz
%Compatible y procesable por MATLAB

vfm('preview',1);

A=0;
B=0;
C=0;
D=0;
imwrite(vfm('grab',1),'Imagen1.tif','tif');
A=imread('Imagen1.tif');
B=rgb2gray(A);
C=medfilt2(B,[4 4]);
D=imadjust(A,[0.3 0.7],[]);
F=im2bw(C);
figure(1)
imshow(B)
figure(2)
imshow(C)
figure(3)
imshow(D)
figure(4)
imshow(F)
```

Fig. 2.13 Algoritmo para obtener imágenes en blanco y negro.

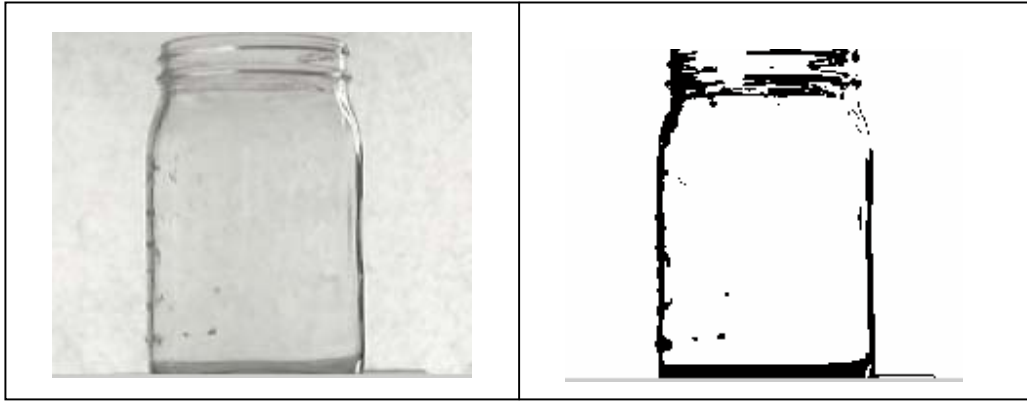


Fig. 2.14 Imagen en blanco y negro obtenida con el programa.

2.8 CONTINUIDAD A PARTIR DEL PREPROCESAMIENTO

Se ha logrado realizar el preprocesamiento de la imagen utilizando las técnicas descritas anteriormente, desde la toma de imagen, conversión a escala de grises, obtención de las características y se obtiene al final una imagen binarizada con las características finales anteriores a la etapa de análisis. Sin embargo se estudian las posibilidades y factibilidades de realizar este análisis mediante técnicas de identificación estadística de los parámetros que definen una anomalía en el frasco de vidrio, así también se ha tomado en cuenta la posibilidad de realizar el estudio de las características desfavorables o no, utilizando redes neuronales entrenadas para tal fin, (teóricamente con una base de datos de imágenes apropiadas para ser aprobados los frascos y otra base conteniendo información de imágenes de frascos dañados, rotos o con los defectos que se desean identificar para ser considerado un envase a rechazar de la línea de embotellado).

2.9 BASES DE DISEÑO DEL SISTEMA DE INSPECCIÓN POR VISIÓN ARTIFICIAL.

1. El sistema será capaz de identificar fallas en frasco cilíndricos de vidrio con un borde diferente al cuerpo del frasco.



Fig. 2.15 Modelo de Frasco a analizar.

2. Los tipos de falla que se identificarán, son básicamente daños o anomalías de aspecto cosmético en la superficie del bote, como vidrios pegados, rajaduras y defectos del tipo ‘hilos de teléfono’.

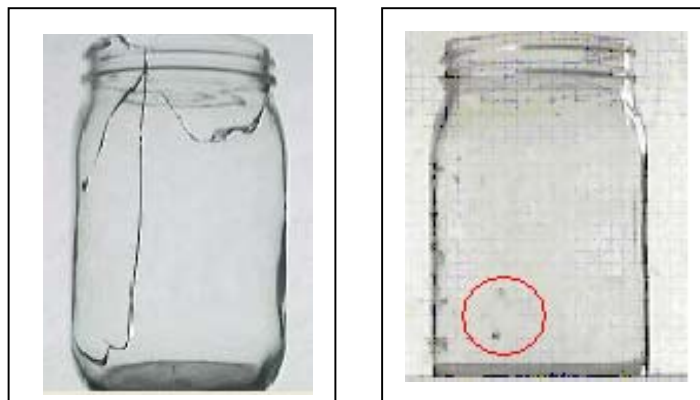


Fig. 2.16 Falla identificable por el sistema.

3. Los botes serán detectados por un sensor reflexivo que conectado a un puerto indicará al sistema la presencia de un frasco en la posición necesaria para capturar su imagen y ser procesada para su posterior aceptación o desecho.
4. Una banda desplazará al bote, haciéndolo pasar justo delante del sistema de iluminación posterior, que ayudará a capturar una imagen con los detalles o defectos capaces de ser identificados por el sistema.

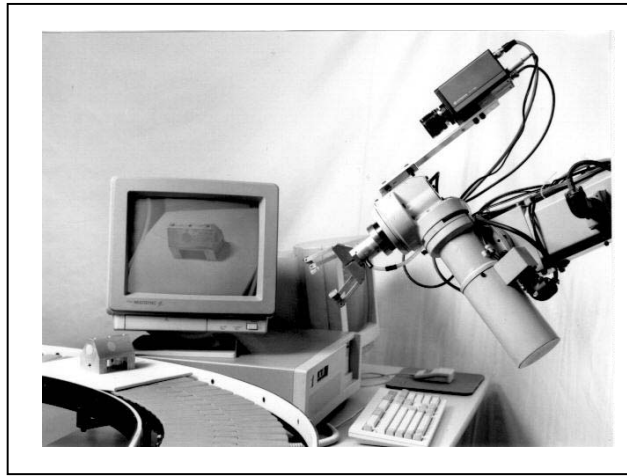
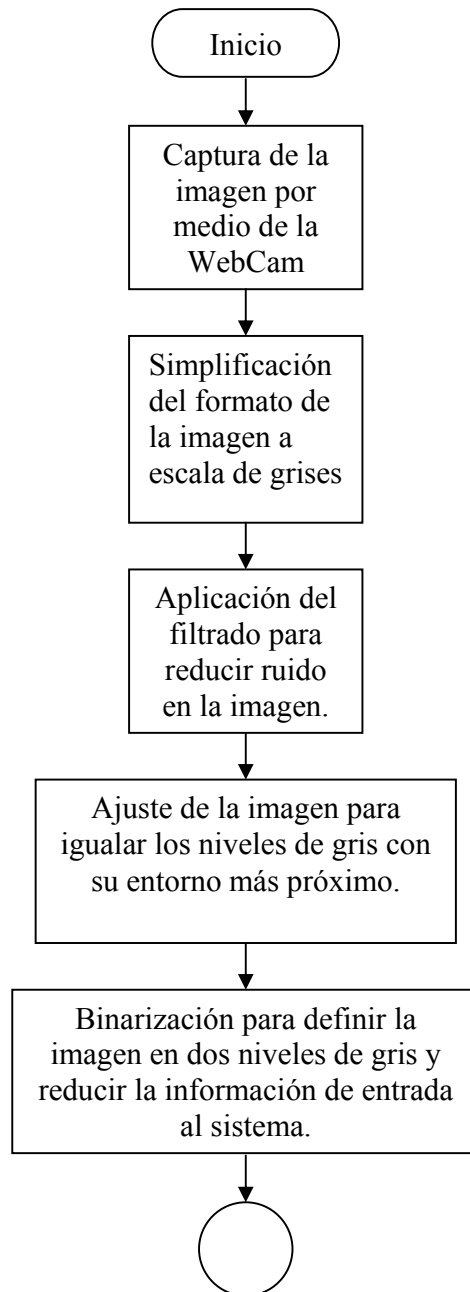


Fig. 2.17 Sistema de inspección (software, cámara, banda, objeto en análisis).

5. La posición de la cámara estará combinada con el diseño de la banda para obtener un espacio óptimo de captura de imagen sin obstáculos que pudieran ser tomados como defectos del bote.

2.10 FLUJOGRAMA DE LA SECUENCIA DE CAPTURA Y PREPROCESAMIENTO.



CAPITULO 3

3. SEGMENTACION DE IMÁGENES

En cualquier imagen se encontrarán presentes uno o varios objetos localizados en un entorno. El objetivo de la segmentación es separar dichos objetos del medio en el que se encuentran y distinguirlos entre si. Después de esta separación estará el proceso de interpretación y reconocimiento.

La segmentación se basa en tres propiedades:

- **Similitud.** Cada uno de los elementos tiene propiedades parecidas de color, textura, etc.
- **Discontinuidad.** Los objetos destacan del entorno y tienen por tanto, uno de sus bordes definidos. Lo mismo ocurre si existe un ocultamiento parcial de un objeto por parte del otro.
- **Conectividad.** Los píxeles tienen que estar agrupados.

Estas suposiciones no son fáciles de cumplir y sin embargo son fundamentales para una buena segmentación. Respecto a la primera los objetos deberían presentar una apariencia uniforme, por ejemplo no debería haber brillos en la imagen. No obstante se encuentra lejos de poder analizar imágenes complejas y que a menudo se tienen que realizar fuertes simplificaciones. Esta limitación, sin embargo, no es un impedimento para que la visión artificial tenga un gran rango de aplicaciones.

Las técnicas para obtener la segmentación se basan en la búsqueda de las partes uniformes de la imagen o justo lo contrario, aquellas partes donde se produce un cambio.

3.1 OBTENCIÓN DE CARACTERÍSTICAS DE UNA IMAGEN BINARIA

A partir de la imagen binaria pueden obtenerse varias características de los objetos que permitan su posterior análisis e identificación.

TAMAÑO:

El tamaño de un objeto está definido por la suma de sus píxeles.

POSICIÓN:

El conocimiento de la posición de un objeto es importante ya que en la mayoría de aplicaciones industriales los objetos se encuentran situados sobre una superficie plana (una mesa , una cinta transportadora); y se conocen los parámetros de la cámara: distancia focal, y la distancia y orientación de la cámara respecto a las superficies sobre la que se encuentran los objetos, por lo que a partir de la posición en la imagen pueden obtenerse las coordenadas del objeto que pueden ser utilizadas por otros elementos del sistema de inspección.

ORIENTACIÓN:

Para que un objeto tenga una orientación debe ser alargado (un círculo, no tiene orientación). El eje que minimice la distancia al de todos los puntos de la imagen será aquel que determine la orientación del objeto.

A continuación se muestran los algoritmos que se han utilizado para el proceso de segmentación, así como los resultados obtenidos:

ALGORITMO PARA SEGMENTAR LA IMAGEN AL OBJETO DE ANALISIS.

```
function respuesta=pixel_pivot2(A)

MATRIZ_EVAL=A;
TAM_MAT=size(MATRIZ_EVAL);
FILAS=TAM_MAT(1,1);
COLUMNAS=TAM_MAT(1,2);
CONTENEDOR(1,1)=1;
CONTENEDOR(1,2)=1;
FILASN=1;

SALIR=0;

FILASE2=FILAS/2;

%---Encuentra punto de referencia para corte izquierdo
for CICLO_COLUMNAS=25:COLUMNAS
    for CICLO_FILAS=1:FILASE2
        PIVOT=A(CICLO_FILAS,CICLO_COLUMNAS);
        if PIVOT<1
            CONTENEDOR(FILASN,1)=CICLO_FILAS;
            CONTENEDOR(FILASN,2)=CICLO_COLUMNAS;
            FILASN=FILASN+1;
            SALIR=1;
            break
        end
    end
    if SALIR>0
        break
    end
end

%PARTE DE CORTE DE IMAGEN

%----- respuesta=CONTENEDOR;|
INICIO_CORTE_X=CONTENEDOR(1,2);

%FIN_CORTE_X=CONTENEDOR2(1,2)-CONTENEDOR(1,2);
FIN_CORTE_X=109;

IMAGEN_CORTADA=imcrop(A,[INICIO_CORTE_X 1 FIN_CORTE_X 180]);
respuesta=IMAGEN_CORTADA;
```

Fig. 3.1 Algoritmo para aislamiento de la imagen.

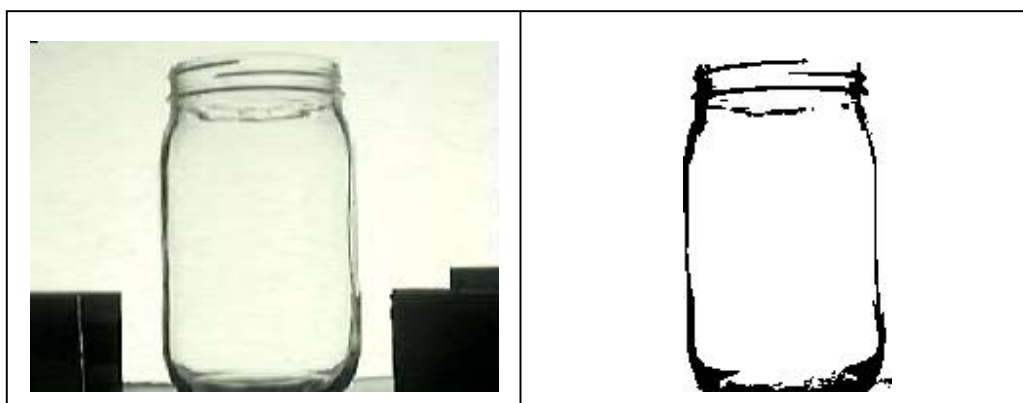


Fig.3.2 a) Imagen normal RGB

b) Imagen Segmentada

ALGORITMO PARA MEJORAR LOS BORDES DEL OBJETO DE ANÁLISIS.

```
function OPERAC=pixel_isolated(A)
I=uint8(A);
%I=imread('fotol.jpg');
%figure,imshow(I),title('original image');
Ic=imcomplement(I);
BW=im2bw(Ic,graythresh(Ic));
%figure,imshow(Ic),title('complement of image');
%figure,imshow(BW);
%title('thresholding the image to show small structures');
se=strel('disk',6);
BWc=imclose(BW,se);
BWco=imopen(BWc,se);
%figure,imshow(BWc),title('closing the thresholded image');
%figure,imshow(BWco);
%title('opening the image to show large objects');
mask=BW&BWco;

INVERTED_MASK=imcomplement(mask);
complem=imcomplement(BW);
OPERAC=not(xor(INVERTED_MASK,complem));
```

Fig. 3.3 Algoritmo para la supresión de bordes.

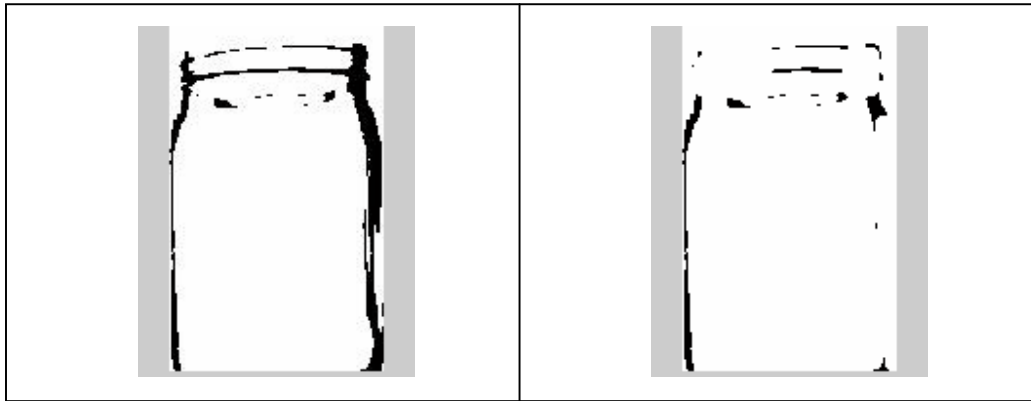


Fig. 3.4 Resultado de simplificación de bordes.

3.2 REDES NEURONALES ARTIFICIALES

Con las redes neuronales artificiales se busca la solución de problemas complejos, no como una secuencia de pasos, sino como la evolución de unos sistemas de computación inspirados en el cerebro humano, y dotados de cierta “inteligencia”, los cuales no son sino la combinación de elementos simples de proceso interconectados, que operando de forma paralela en varios estilos, consiguen resolver problemas relacionados con el reconocimiento de formas o patrones, predicción, codificación, control y optimización entre otras aplicaciones.

Las Redes Neuronales Artificiales son una simulación abstracta de un sistema nervioso real que está formado por un conjunto de unidades neuronales conectadas unas con otras por medio de conexiones del tipo axón. Estas conexiones tienen una gran semejanza con las dendritas y los axones en los sistemas nerviosos biológicos.

El Modelo Biológico

La teoría y modelado de redes neuronales está inspirada en la estructura y funcionamiento de los sistemas nerviosos, donde la neurona es el elemento fundamental. En general, una neurona consta de un cuerpo celular más o menos esférico, de 5 a 10 micras de diámetro, del que salen una rama principal, el axón, y varias ramas más cortas, llamadas dendritas.

Una de las características de las neuronas es su capacidad de comunicarse. En términos generales las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina e integra y emite señales de salida. El axón transmite dichas señales a los terminales axónicos, que distribuyen información o un nuevo conjunto de neuronas, se calcula que en el cerebro humano existen del orden de 10^{15} conexiones.

Las señales que se utilizan son de dos tipos: eléctrica y química. La señal generada por la neurona y transportada a lo largo del axón es un impulso eléctrico, mientras que la señal que se transmite entre los terminales axónicos de una neurona y las dendritas de la otra es de origen químico.

Para establecer una similitud directa entre la actividad sináptica y la analogía con las redes neuronales artificiales se puede considerar: Las señales que llegan a la sinapsis son las entradas a la neurona; estas son ponderadas (atenuadas o simplificadas) a través de un parámetro, denominado peso asociado a la sinapsis correspondiente.

Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo).

El efecto es la suma de las entradas ponderadas. Si la suma es igual o mayor que el umbral de la neurona, entonces la neurona se activa (da salida). Esta es una situación de todo o nada; cada neurona se activa o no se activa. La facilidad de transmisión de señales se altera mediante la actividad del sistema nervioso. Las sinapsis son susceptibles a la fatiga, deficiencia de oxígeno y la presencia de anestésicos, entre otros. Esta habilidad de ajustar señales es un mecanismo de aprendizaje.

Elementos de una Red Neuronal Artificial

Las redes neuronales artificiales (RNA) son modelos que intentan reproducir el comportamiento del cerebro. Como tal modelo, realiza una simplificación, averiguando cuáles son los elementos relevantes del sistema, bien porque la cantidad de información de que se dispone es excesiva o bien porque es redundante. Una elección adecuada de sus características, más una estructura conveniente, es el procedimiento convencional utilizado para construir redes capaces de realizar determinada tarea.

1. Unidad de proceso: La neurona Artificial. Existen tres tipos de unidades en cualquier sistema: entradas, salidas y ocultas. Las unidades de entrada reciben señales desde el entorno; las de salida envían la señal fuera de la red, y las unidades ocultas son aquellas cuyas entradas y salidas se encuentran dentro del sistema. Se conoce como capa o nivel a un conjunto de neuronas cuyas entradas provienen de la misma fuente y cuyas salidas se dirigen al mismo destino.
2. Estado de Activación. Los estados del sistema en un tiempo t se representan por un vector $A(t)$. Los valores de activación pueden ser continuos o discretos, limitados o ilimitados. Si son discretos, suelen tomar un conjunto discreto de valores binarios, así un estado activo se indicaría con un 1 y un estado pasivo se representaría por un cero. En otros modelos se considera un conjunto de estados de activación, en cuyo valor entre $[0,1]$, o en el intervalo $[-1,1]$, siendo una función sigmoideal.
3. Función de Salida o de Transferencia. Asociada con cada unidad hay una función de salida, que transforma el estado actual de activación en una señal de salida. Existen cuatro funciones de transferencia típicas que determinan distintos tipos de neuronas:
 - ❑ Función Escalón
 - ❑ Función Lineal y Mixta
 - ❑ Sigmoideal
 - ❑ Función Gaussiana
4. Conexiones entre neuronas. Las conexiones que unen a las neuronas que forman una RNA tiene asociado un peso, que es el que hace que la red adquiera conocimiento. Se considera que el efecto de cada señal es aditivo, de tal forma que la entrada neta que recibe una neurona es la suma del producto de cada señal individual por el valor de la sinapsis que conecta ambas neuronas y es lo que se conoce como red de propagación. Se utiliza una matriz W con todos los pesos, Si w_{ji} es positivo indica que la relación entre las neuronas es excitadora, es decir, siempre que la neurona i esté activada, la neurona j recibirá una señal que tenderá

- a activarla. Si w_{ji} es negativo, la sinapsis será inhibitoria. En este caso si i está activada, enviará una señal que desactivará a j . Finalmente si w_{ji} es 0 se supone que no hay conexión entre ambas.
5. Función o Regla de Activación. Se requiere una regla que combine las entradas con el estado actual de la neurona para producir un nuevo estado de activación. Esta función F produce un nuevo estado de activación en una neurona a partir del estado que existía y la combinación de las entradas con los pesos de las conexiones. Esa F es denominada función de activación, y las salidas que se obtienen en una neurona para las diferentes formas de F serán:
- ❑ Función de Activación Escalón
 - ❑ Función de Activación Identidad
 - ❑ Función de Activación Lineal -Mixta
 - ❑ Función de Activación Sigmoidal
6. Regla de Aprendizaje. El aprendizaje puede ser comprendido como la modificación de comportamiento inducido por la interacción con el entorno y como resultado de experiencias conduce al establecimiento de nuevos modelos de respuesta a estímulos externos. En el cerebro humano el conocimiento se encuentra en la sinapsis. En el caso de las RNA el conocimiento se encuentra en los pesos de las conexiones entre neuronas. Todo proceso de aprendizaje implica cierto número de cambios en estas conexiones. En realidad, puede decirse que se aprende modificando los valores de los pesos de la red.

Estructura de una Red Neuronal Artificial

La distribución de neuronas dentro de la red se realiza formando niveles o capas de un número determinado de neuronas cada una, y que existen capas de entrada, de salida, y ocultas. Cuando ninguna salida de las neuronas es entrada de neuronas del mismo nivel o de niveles precedentes, la red se describe como de propagación hacia adelante.

Cuando las salidas pueden ser conectadas como entradas de neuronas de niveles previos o del mismo nivel, incluyéndose ellas mismas, la red es de propagación hacia atrás.

Características de las Redes Neuronales

Existen cuatro aspectos que caracterizan una red neuronal: su topología, el mecanismo de aprendizaje, tipo de asociación realizada entre la información de entrada y salida, y la forma de representación de estas informaciones:

1. Topología de las Redes Neuronales. La arquitectura de las redes neuronales consiste en la organización y disposición de las neuronas formando capas más o menos alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Redes Monocapa: Se establecen conexiones laterales, cruzadas o autorrecurrentes entre las neuronas que pertenecen a la única capa que constituye la red. Se utilizan en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para generar informaciones de entrada, que se presentan a la red, incompletas o distorsionadas.

Redes Multicapa: Son aquellas que disponen de conjuntos de neuronas agrupadas en varios niveles o capas. Una forma de distinguir la capa a la que pertenece la neurona, consiste en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Según el tipo de conexión, se distinguen las redes feedforward, y las redes feedforward/feedback.

2. Mecanismo de Aprendizaje. El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero, una conexión

se destruye cuando su peso pasa a ser cero. Se puede afirmar que el proceso de aprendizaje ha finalizado (la red ha aprendido) cuando los valores de los pesos permanecen estables ($dw_{ij} / dt = 0$).

Un criterio para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual, o si el aprendizaje supone la desconexión de la red.

Otro criterio suele considerar dos tipos de reglas de aprendizaje: las de aprendizaje supervisado y las correspondientes a un aprendizaje no supervisado, estas reglas dan pie a una de las clasificaciones que se realizan de las RNA: *Redes neuronales con aprendizaje supervisado y redes neuronales con aprendizaje no supervisado*. La diferencia fundamental entre ambos tipos consiste en la existencia o no de un agente externo (supervisor) que controle el aprendizaje de la red.

Redes con Aprendizaje Supervisado:

El proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada. El agente externo comprueba la salida de la red y en el caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida se aproxime a la deseada.

Se consideran tres formas de llevar a cabo este tipo de aprendizaje:

- Aprendizaje por corrección de error: Consiste en ajustar los pesos en función de la diferencia entre los valores deseados y los obtenidos en la salida de la red; es decir, en función del error.
- Aprendizaje por refuerzo: Se basa en la idea de no indicar durante el entrenamiento exactamente la salida que se desea que proporcione la red ante una determinada entrada. La función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito=+1 o fracaso=-1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades.

- Aprendizaje estocástico: Consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

Redes con Aprendizaje No Supervisado:

Estas redes no requieren influencia externa para ajustar los pesos de las conexiones entre neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada es o no correcta, así que existen varias posibilidades en cuanto a la interpretación de la salida de estas redes.

En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado en el pasado. En otro caso podría realizar una codificación de los datos de entrada, generando a la salida una versión codificada de la entrada, con menos bits, pero manteniendo la información relevante de los datos, o algunas redes con aprendizaje no supervisado lo que realizan es un mapeo de características, obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de entrada, de tal forma que si se presentan a la red informaciones similares, siempre sean afectadas neuronas de salidas próximas entre sí, en la misma zona del mapa.

En general en este tipo de aprendizaje se suelen considerar dos tipos:

- Aprendizaje Hebbiano: Consiste básicamente en el ajuste de los pesos de las conexiones de acuerdo con la correlación, así si las dos unidades son activas (positivas), se produce un reforzamiento de la conexión. Por el contrario cuando una es activa y la otra pasiva (negativa), se produce un debilitamiento de la conexión.
- Aprendizaje competitivo y cooperativo: Las neuronas compiten (y cooperan) unas con otras con el fin de llevar a cabo una tarea dada. Con este tipo de aprendizaje se pretende que cuando se presente a la red cierta información de entrada, solo una de las neuronas de salida se active (alcance su valor de

respuesta máximo). Por tanto las neuronas compiten por activarse, quedando finalmente una, o una por grupo, como neurona vencedora.

3. Representación de la Información de Entrada y Salida. Las redes neuronales pueden también clasificarse en función de la forma en que se representan las informaciones de entrada y las respuestas o datos de salida. Así un gran número de redes, tanto los datos de entrada como de salida son de naturaleza analógica, cuando esto ocurre, las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoideal. Otras redes sólo admiten valores discretos o binarios a su entrada, generando también unas respuestas en la salida de tipo binario. En este caso, las funciones de activación de las neuronas son de tipo escalón. Existe también un tipo de redes híbridas en las que las informaciones de entrada pueden ser valores continuos, aunque las salidas de la red son discretas.

Ventajas de las Redes Neuronales

1. **Aprendizaje Adaptativo:** Es una de las características más atractivas de las redes neuronales, es la capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.
En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan unos resultados específicos. Una RNA no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de los pesos de los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado el periodo inicial de entrenamiento.
2. **Autoorganización:** Las redes neuronales usan su capacidad de aprendizaje adaptativo para organizar la información que reciben durante el aprendizaje y/o la operación. Una RNA puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje. Esta autoorganización provoca la facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a los que no habían sido expuestas anteriormente.

3. **Tolerancia a Fallos:** Comparados con los sistemas de computadora tradicionales, los cuales pierden su funcionalidad en cuanto sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un pequeño número de neuronas, aunque el comportamiento del sistema se ve influenciado, sin embargo no sufre una caída repentina.
4. **Operación en Tiempo Real:** Los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
5. **Fácil inserción dentro de la tecnología existente:** Debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo costo, es fácil insertar RNA para aplicaciones específicas dentro de sistemas existentes (chips, por ejemplo). De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas de forma incremental, y cada paso puede ser evaluado antes de acometer un desarrollo más amplio.

Aplicaciones de las Redes Neuronales

Biología:

Aprender más acerca del cerebro y otros sistemas.

Obtención de modelos de la retina.

Empresa:

Reconocimiento de caracteres escritos.

Explotación de bases de datos.

Síntesis de voz desde texto.

Manufacturación:

Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.)

Control de producción en líneas de proceso.

Inspección de calidad.

Filtrado de señales.

Militares:

Clasificación de las señales de radar.

Creación de armas inteligentes.

Optimización del uso de recursos escasos.

CAPITULO 4

4. CLASIFICACION DE LA IMAGEN

4.1 TEORIA DE LA RESONANCIA ADAPTATIVA (ART1s)

Una de las características de la memoria humana consiste en:

Habilidad para aprender nuevos conceptos sin necesitar de olvidar otros aprendidos en el pasado.

Tarea de Clasificación de Patrones.

Problema bien definido ----- Conjunto de Entrenamiento.

Aprendizaje Correcto ----- Matriz de Pesos (finales)

Limitaciones:

- ❑ Problemas no están bien definidos.
- ❑ Añadir nuevos patrones de interés una vez que se ha realizado el entrenamiento con unos iniciales puede suponer el olvido de lo aprendido en la primera fase.

Lo que se ha intentado mostrar en esta descripción es lo que Grossberg denomina el Dilema de la Estabilidad y Plasticidad del Aprendizaje. Este dilema plantea los siguientes interrogantes:

- ❑ Poder aprender nuevos patrones (*Plasticidad del aprendizaje*).
- ❑ Poder retener los patrones previamente aprendidos (*Estabilidad del aprendizaje*)

Conseguir una red que pueda dar respuesta a uno de estos interrogantes es sencillo, no lo es, si se pretende diseñarla para que solucione ambos.

En respuesta a este dilema, Grossberg y Carpenter desarrollaron la denominada Teoría de la Resonancia Adaptativa (ART).

- ❑ Se aplica a sistemas competitivos.
- ❑ Lo que se pretende es categorizar (clusterizar).
- ❑ Aprendizaje Sin Supervisión.

IDEA: Para solucionar el dilema de la plasticidad y estabilidad, el modelo ART propone añadir a las redes un mecanismo de realimentación entre las neuronas competitivas de la capa de salida de la red y la capa de entrada. Este mecanismo facilita el aprendizaje de nueva información sin destruir la ya almacenada.

La teoría de la resonancia adaptativa se basa en la idea de hacer resonar la información de entrada con los representantes o prototipos de las categorías que reconoce la red.

- ❑ Si entra en resonancia con alguno, es suficientemente similar, la red considera que pertenece a dicha categoría y únicamente realiza una pequeña adaptación del prototipo almacenado representante de la categoría para que incorpore algunas características del dato presentado.
- ❑ Cuando no resuena con ninguno, no se parece a ninguno de los existentes, recordados por la red hasta ese momento, la red se encarga de crear una nueva categoría con el dato de entrada como prototipo de la misma.

Dos operaciones: **Entrenamiento y Adaptación.**

4.2 ARQUITECTURA Y FUNCIONAMIENTO

RED ARTIS

Una red ART básicamente consta de dos capas entre las que se establecen conexiones hacia adelante y hacia atrás y dos tipos de señales de control.

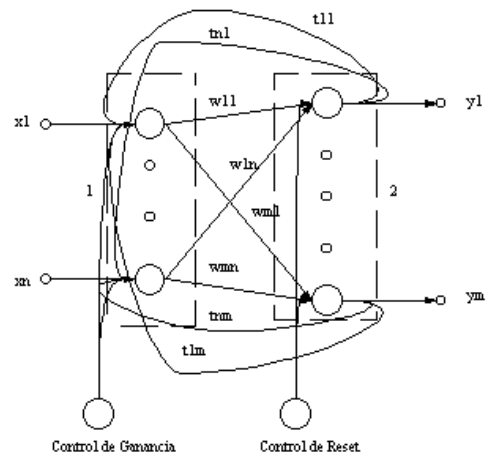


Fig. 4.1 Capas y conexiones de la red ART.

Cada una de las capas existentes de la red realiza diferentes tareas:

Capa 1 (Competitiva)

Operaciones:

Entrada y Comparación.

Capa 2

Operaciones:

Salida y Reconocimiento

OPERACIÓN

Se puede describir la acción de la red en términos de la actividad de cada capa para una serie de fases:

- ❑ Fase de Inicialización.
- ❑ Fase de Reconocimiento.
- ❑ Fase de Comparación.
- ❑ Fase de Búsqueda.

FASE DE INICIALIZACIÓN

Esta fase consiste en inicializar los diferentes parámetros de la red y determinar como serán las señales control.

Los pesos se inicializan siguiendo el siguiente criterio:

$$w_{ji} = 1/(1+N), \text{ en donde } N = n^{\circ} \text{ entradas (conexiones hacia adelante).}$$
$$t_{ij} = 1 \text{ (conexiones hacia atrás).}$$

Señales de Control

Las señales de control se utilizan para manejar el flujo de datos a través de la red y conmutar entre las diferentes fases:

- Control de Ganancia: controla el flujo para la capa de entrada y conmuta entre dos estados: Entrada de Datos y Comparación.

Esta señal toma valores binarios y dichos valores suelen depender de varias señales:

- Salidas de las Neuronas de la capa de Salida ponderadas por pesos (-N)
- Vector de entrada ponderado por pesos (1).

Señal de Ganancia (SG)

SG es uno siempre que se le presenta a la red un patrón de entrada válido.

Si en la capa competitiva existe alguna neurona activa SG es forzado a 0

$$SG = \begin{cases} 1 & \sum_{i=1}^N xi - N \sum_{j=1}^M y_j \geq 0.5 \\ 0 & \sum_{i=1}^N xi - N \sum_{j=1}^M y_j < 0.5 \end{cases}$$

Ec. 4.1 Señal de Ganancia

Control de Reset: Es el más simple de los dos. Controla el flujo de los datos para la capa de salida. Sirve para conmutar también entre dos estados: Reconocimiento y Búsqueda.

Utiliza valores binarios y estos valores activan o desactivan los nodos en la capa de reconocimiento (capa de salida). Lo mismo que en el caso anterior estos valores suelen depender de diferentes señales. Salidas de las neuronas, capa de entrada ponderadas por (-1). Vector de entrada ponderado por ρ (parámetro de vigilancia). De tal manera la señal de Reset será de la siguiente forma:

$$SR = \begin{cases} 1 & \sum_{i=1}^N \rho x_i - \sum_{i=1}^N neta_i^1 > 0 \quad \text{Reset} \\ 0 & \sum_{i=1}^N \rho x_i - \sum_{i=1}^N neta_i^1 \leq 0 \end{cases}$$

Ec. 4.2 Señal de Reset

La señal de reset tiene como misión desactivar la neurona ganadora en la capa de salida cuando un vector de entrada no es suficientemente similar al prototipo de una clase. Esta medida de similaridad depende del factor o parámetro de reset.

FASE DE RECONOCIMIENTO

En la fase de reconocimiento un vector de entrada es presentado a la red, activando una neurona en la capa de salida.

Las neuronas en la capa de entrada reciben 3 entradas:

- Vector de entrada.
- Salidas de la capa 2.
- Control de ganancia.

En esta fase la señal de ganancia es 1 ya que ninguna neurona de la capa de salida está activada. Las neuronas de la capa de entrada se activan utilizando la regla 2/3. Es decir si

dos o más entradas están activadas la neurona se activa produciendo uno, en otro caso produce un cero.

Para las neuronas de la capa de salida la operación que se realiza es la siguiente:

$Y_j = 1$, si $\sum W_{ji} * X_i$ es máximo.

$Y_j = 0$, en otro caso.

Entradas Binarias (distancia mínima) (Operación AND).

FASE DE COMPARACION

Dos vectores son presentados en la capa de entrada: vector de entrada y vector producido por la capa de salida (una neurona activada).

Se conmuta entre dos operaciones: Reconocimiento (entrada) y Comparación.

Señal de control de ganancia es cero.

Al estar la señal de ganancia puesta a cero, la salida de la capa de entrada es un AND entre el vector de entrada y el vector producido por la capa de salida, ponderado por los pesos 'feedback'.

Test de Vigilancia

El resultado obtenido en la capa de entrada después de la fase de comparación es enviado al control de Reset. Simplemente significa una medida de similaridad entre el vector de entrada y el vector prototipo obtenido por la capa de salida.

La comparación se hace valorando la siguiente relación de semejanza:

$$\text{relación de semejanza} = \frac{\|X * Y\|}{\|X\|} = \frac{\|X * T_j\|}{\|X\|}$$

$$\|X * Y\| = \sum_{i=1}^N x_i * y_{ij} = \sum_{i=1}^N x_i * t_{ij} \quad j = \text{neurona ganadora}$$

$$\|X\| = \sum_{i=1}^N x_i$$

Ec. 4.3. Relación de semejanza entre el vector de entrada y el vector prototipo.

Al trabajar con valores binarios (0/1), el producto aritmético equivale al lógico (AND), y por tanto, lo que se representa es: el número de componentes binarios con valor 1 que coinciden entre el vector de entrada y el de salida.

En el caso de que fueran completamente iguales, este valor coincidiría con el denominador, y la relación de semejanza sería la unidad (representando el 100%). Si no coincidieran en ningún bit, entonces sería 0 (representado el 0%).

Una vez calculada la relación de semejanza entre ambas informaciones, se compara dicha relación con el parámetro de vigilancia ρ , cuyo valor será fijado por el usuario en un rango de 0 a 1 y que influirá en el número de clases que establecerá la red, ya que cuando mayor sea su valor, se está pidiendo al sistema que discrimine con mayor precisión, de tal forma que si el valor asignado es 1 se estará indicando que a una clase sólo pueden pertenecer patrones que sean exactamente iguales.

Si se cumple que *Relación de Semejanza* $< \rho$, entonces la neurona vencedora j , en la capa de salida no representa la apropiada categoría a la que pertenece la información de entrada y es desestimada, se elimina del conjunto de posibles neuronas vencedoras, y se vuelve a presentar el patrón a la estructura.

En el caso contrario se ha encontrado la categoría apropiada al vector de entrada y se deben actualizar los pesos para confirmar este hecho.

Actualización de pesos:

Los pesos de forward (W) tienen el mismo valor que los feedback (T), pero normalizado:

$$W = T / (\gamma + \sum T) \quad \text{Ec.4.4 Pesos de forward.}$$

γ es un valor pequeño, normalmente se utiliza 0.5, aunque en la inicialización de los pesos se recomienda como valor la unidad (Lippman).

Si la neurona de salida j no es la vencedora, el valor de su salida es cero, con lo que no debe producirse variación en el peso correspondiente. Por el contrario, si la neurona es la ganadora si se debe producir variación en los pesos asociados a ella.

La regla de cambio de pesos es la siguiente:

$$t_{ij}(t+1) = t_{ij}(t) * X_i \quad \text{Ec.4.5 Regla de cambio de Pesos}$$

Los pesos W_{ij} se obtienen normalizando los

$$W_{ji}(t+1) = \frac{t_{ij}(t) * X_i}{\gamma + \sum_{i=1}^N t_{ij}(t) * X_i} \quad \text{Ec.4.6 Pesos } W_{ij}$$

FASE DE BÚSQUEDA

Si en un primer momento se ha determinado que la primera neurona ganadora no representa la categoría del vector de entrada, se debe empezar la búsqueda por diferentes categorías establecidas por la red.

Al no representar la categoría del vector de entrada la señal de control de reset, desactiva dicha neurona ganadora.

Esta acción tiene un doble efecto: por una parte fuerza su salida a 0 y la excluye para las sucesivas búsquedas y por otro lado la señal de control de ganancia es puesta a 1 (ninguna neurona de la capa de salida es ganadora).

El vector de entrada es entonces de nuevo aplicado a la estructura repitiéndose entonces los pasos previamente expuestos. En estas circunstancias se puede llegar a 2 situaciones completamente diferentes:

- Si todavía quedan neuronas de las salidas que no hayan sido desestimadas, se repite el proceso de búsqueda de la categoría a la que pertenece la entrada.
- Si no queda ninguna neurona candidata porque ya se ha repetido el proceso, entonces se produce una *situación de saturación* de la red, ya que no se puede

crear una nueva categoría para el patrón, al no existir ninguna neurona de salida que pueda asociarse a la nueva clase.

Podría solucionarse el problema ampliando el número de neuronas de salida de la red de forma dinámica, ya que los nuevos pesos no afectarían a representantes ya almacenados en el resto de conexiones que componen la memoria a largo plazo de la red.

ALGORITMO GENERAL

1. Se inicializan pesos y parámetro de vigilancia.
2. Se aplica una nueva entrada.
3. Se determina la neurona ganadora en la capa de salida.
4. Se compara la categoría que representa la neurona ganadora con la entrada.
Si pertenece a la categoría, se va al paso 5.
Si no pertenece a la categoría se desactiva la ganadora y se vuelve al paso 2.
5. Se adaptan los pesos.
6. Se desactivan todas las neuronas de la capa de salida.
7. Se vuelve al paso 2.

LIMITACIONES DE LA RED ART

Aunque la red ART es una de los más potentes modelos con capacidad autoorganizativa, sin supervisión, existen algunas limitaciones importantes a tener en cuenta que dificultan la tarea de clasificación realizada por la red. Estas limitaciones se refieren a:

1. Su dependencia del tipo y orden de las informaciones aprendidas.

2. La influencia del parámetro de vigilancia en el número de categorías creadas por la red, ya que pequeños cambios de este parámetro pueden originar un gran número de categorías.
3. Se trata de una red con una gran sensibilidad ante el ruido o distorsión, también ante el desplazamiento de los datos de entrada, que puede dar lugar a una gradual degradación de los prototipos correspondientes, debido al carácter destructivo (operación AND de datos de entrada y prototipo) de la regla de aprendizaje que realiza la adaptación de los pesos.
Sin embargo, si estos datos de entrada son sometidos a un adecuado preprocesamiento, la red ART puede ser útil para realizar una posterior tarea de clasificación o categorización.
4. También este modelo presenta el problema de la ineficiencia en cuanto a las necesidades requeridas para el almacenamiento de los pesos de las conexiones entre neuronas. Hay que tener en cuenta que se precisan $2*N$ para representar cada categoría de N bits.

4.3. ENTRENAMIENTO DE LA RED ART1S PARA EL SISTEMA DE INSPECCIÓN BASADO EN VISION ARTIFICIAL

En la etapa de entrenamiento de la red, se han debido manejar ciertos parámetros propios de la ART1S, deben especificarse entradas binarias, en formato de matrices de una sola columna y con la misma cantidad de elementos. Para el caso del sistema de inspección, se han utilizado columnas conteniendo más de 10,000 elementos, debido al tamaño de las imágenes que se están adquiriendo mediante la cámara.

Las entradas deben tener el mismo tamaño, pues en caso contrario la red responde de manera errónea. Las categorías de salida de la red van a estar determinadas de acuerdo a la cantidad de entradas y sujetas al parámetro de vigilancia ρ , pues un valor de ρ bajo, permitirá una menor cantidad de categorías como salidas de la red y viceversa.

Algoritmo para el entrenamiento de la red una vez definidas las imágenes en formato de una sola columna.

Si 'A' y 'B', son matrices binarias representativas de imágenes de frascos, deben trasladarse a formato de matrices de una sola columna de la siguiente manera:

$C = \text{vec2mat}(A, 1);$

$D = \text{vec2mat}(B, 1);$

En donde C y D contienen información lista para entrar a la red y lograr su entrenamiento.

$P = [C, D];$

$[V, W] = \text{art1s}(P, \rho);$ en donde W son los pesos de la red y V el vías

Resultado = $\text{compet}(W * \text{Im});$ en donde Im es la imagen a comparar en formato de una columna.

Salida = $\text{find}(\text{Resultado} == 1);$ En donde Salida debe corresponder a una de las categorías Previamente definidas en el entrenamiento.

JUSTIFICACION DE LA ELECCION DEL TIPO DE RED

La principal propiedad de una red neuronal artificial es la capacidad de aprender del entorno en el que opera y de mejorar su funcionamiento. El aprendizaje es el proceso mediante el cual, los parámetros de una red neuronal artificial se adaptan como consecuencia de un proceso de estimulación llevado a cabo por el entorno en el que la red opera. El tipo de aprendizaje vendrá determinado por la forma en que cambie la configuración de la red. Para el caso, la red neuronal ART1S presenta un tipo de aprendizaje no supervisado, se utilizo este tipo de aprendizaje debido a que sólo se utilizan vectores de entrada para entrenar la red. Los valores de las salidas son determinados por la red durante el curso del aprendizaje. Se utiliza para construir modelos internos que capturan similitudes en los valores de los vectores de entrada, sin recibir información adicional.

En esta forma de aprendizaje la red sólo se entrena con vectores de entrada. La RNA decide cuales son las figuras que determinan el agrupamiento de los vectores de

entrada y así genera la salida. Por tanto, no hay valores deseados (o supervisión) ni realimentación basándose en las valoraciones aportadas por un “crítico” (o fortalecimiento). Esta red se caracteriza por su capacidad de clasificar conjuntos de datos de entrada. Poseen un solo nivel con muchas conexiones cuyos pesos necesitan inicializarse. Las neuronas compiten de forma que las triunfadoras son las únicas que modifican los pesos asociados a sus conexiones.

Cabe mencionar, que se utilizó la red neuronal ART1S debido a que este tipo de red trabaja solamente con valores binarios; además que este tipo de red realiza un análisis de la imagen píxel a píxel. Lo cual era una ventaja debido a que la imagen con la que trabaja el sistema está en formato binario; esto gracias a que luego de la etapa de adquisición de imagen y pre-procesamiento se obtiene una imagen con las características esenciales de la imagen a analizar.

La red neuronal ART1S tiene la opción de realizar un entrenamiento con patrones; la cual presentará una respuesta en función de estos patrones; esta respuesta se tendrá luego de realizar un análisis píxel a píxel de la imagen patrón con la imagen a analizar; y aquella que presente o active la función ganadora presentará una respuesta en función de una de las categorías con la que fue entrenada; por tanto resulta conveniente debido a que el objetivo del sistema es presentar una respuesta de bueno o malo según el tipo de frasco.

Algoritmo principal para la clasificación de objetos

```
function varargout = PHOTOVERS4(varargin)
% PHOTOVERS4 M-file for PHOTOVERS4.fig
%   PHOTOVERS4, by itself, creates a new PHOTOVERS4 or raises the
existing
%   singleton*.
%
%   H = PHOTOVERS4 returns the handle to a new PHOTOVERS4 or the
handle to
%   the existing singleton*.
%
%   PHOTOVERS4('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in PHOTOVERS4.M with the given input
arguments.
%
%   PHOTOVERS4('Property','Value',...) creates a new PHOTOVERS4 or
raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before PHOTOVERS4_OpeningFunction gets
called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to PHOTOVERS4_OpeningFcn via
varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help PHOTOVERS4

% Last Modified by GUIDE v2.5 04-Aug-2004 14:53:01

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @PHOTOVERS4_OpeningFcn, ...
                  'gui_OutputFcn',  @PHOTOVERS4_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```

else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before PHOTOVERS4 is made visible.
function PHOTOVERS4_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to PHOTOVERS4 (see VARARGIN)

% Choose default command line output for PHOTOVERS4
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes PHOTOVERS4 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = PHOTOVERS4_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'))
;
end

function edit1_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as
a double

density = str2double(get(hObject, 'String'));
entero=round(density);

if (density-entero)~=0
    set(hObject, 'String', 1);
    errordlg('No se permiten decimales','Error');
end

if isnan(density)
    set(hObject, 'String', 1);
    errordlg('Ingrese solamente numeros enteros','Error');
end

data = getappdata(gcf, 'metricdata');
data.density = density;
setappdata(gcf, 'metricdata', data);

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%def={'9'};
%numero_imagenes=inputdlg('INTRODUZCA EL NUMERO DE BOTES','CANTIDAD DE
BOTES',[1 20],def);
%numero_imagenes=cell2mat(numero_imagenes);

%   if isempty(numero_imagenes)==1,
%       numero_imagenes='9';
%   end
data = getappdata(gcf, 'metricdata');
density=data.density;

%-----Cargando la variable que contiene los pesos-----

Wanda=Wendolin_Trans(1);

ion='VOID';
set(handles.text8,'string',ion);

CONTADOR_BOTES=0;
Entrada=1;
SENSOR=0;
set(handles.text4,'string',CONTADOR_BOTES);

```

```

for BOTES=1:density,
    DIO = digitalio('parallel',1);
    hline=addline(DIO, 0, 'Out');
    putvalue(DIO, 0);

    DIO1 = digitalio('parallel',1);
        hline1=addline(DIO1, 10,'In');

while SENSOR<1
    SENSOR=getvalue(DIO1);
end
CONTADOR_BOTES=CONTADOR_BOTES+1;
set(handles.text4,'string',CONTADOR_BOTES);

%-----Retraso para desplazamiento-----
tic
for RETRASO=1:4333333*30*2.8
end
toc
DIO2 = digitalio('parallel',1);
hline2=addline(DIO2, 0, 'Out');
putvalue(DIO2, 1);

%-----Etapa de toma de Imagen-----
%for RETRASO11=1:100000

%end

Qbert=TakeP(Entrada);
MOCHADO=pixel_pivot4(Qbert);
imwrite(MOCHADO,'picture.jpg','jpg');
CUTTED=imread('picture.jpg');
KRONOS=pixel_isolated(CUTTED);
CUTTED2=pixel_pivot3(KRONOS);

size(CUTTED2);

%axes(handles.axes3)
%imshow(CUTTED)

%----- Convierte matriz CUTTED a una matriz M x 1-----
TRANSFORMER=vec2mat(CUTTED2,1);
size(TRANSFORMER);
axes(handles.axes3)
imshow(CUTTED)
axes(handles.axes4)
imshow(CUTTED2)
drawnow
Entrada_Red=compet(Wanda*TRANSFORMER);
Salida_Red=find(Entrada_Red==1)

```

```

if Salida_Red>1
    ion='Frasco Malo';
    set(handles.text8,'string',ion);
else
    ion='Frasco Bueno';
    set(handles.text8,'string',ion);
end

%-----Retraso para reinicio de movimiento de banda-----
for RETRASO2=1:10000
    B=1;
end
SENSOR=0;
end

DIO3 = digitalio('parallel',1);
hline3=addline(DIO3, 0, 'Out');
putvalue(DIO3, 1);

% --- Executes on button press in pushbutton1.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

close(PHOTOVERS4);

```

4.4. FUNCIONES UTILIZADAS POR EL ALGORITMO PRINCIPAL.

Función de adquisición de imagen más preprocesamiento.

```

function salida=TakeP(entrada);
%-----Toma de imagenes y almacenamiento temporal-----
imwrite(vfm('grab',1),'Fkbotecito.jpg','jpg');
Imagen=imread('Fkbotecito.jpg');

%-----Comandos de preprocesamiento de la imagen -----

IMAGENRGB=rgb2gray(Imagen);
IMAGENMEJOR=imadjust(IMAGENRGB,[0.3 0.7],[]);
IMAGENNEGRO=im2bw(IMAGENMEJOR,0.9999999);
salida=IMAGENNEGRO;

```

Algoritmo Para Mejorar Los Bordes Del Objeto De Análisis.

```
function OPERAC=pixel_isolated(A)
I=uint8(A);
%I=imread('fotol.jpg');
%figure,imshow(I),title('original image');
Ic=imcomplement(I);
BW=im2bw(Ic,graythresh(Ic));
%figure,imshow(Ic),title('complement of image');
%figure,imshow(BW);
%title('thresholding the image to show small structures');
se=strel('disk',6);
BWc=imclose(BW,se);
BWco=imopen(BWc,se);
%figure,imshow(BWc),title('closing the thresholded image');
%figure,imshow(BWco);
%title('opening the image to show large objects');
mask=BW&BWco;

INVERTED_MASK=imcomplement(mask);
complem=imcomplement(BW);
OPERAC=not(xor(INVERTED_MASK,complem));
```

Fig. 3.3 Algoritmo para la supresión de bordes.

ALGORITMO PARA SEGMENTAR LA IMAGEN AL OBJETO DE ANALISIS.

```
function respuesta=pixel_pivot2(A)

MATRIZ_EVAL=A;
TAM_MAT=size(MATRIZ_EVAL);
FILAS=TAM_MAT(1,1);
COLUMNAS=TAM_MAT(1,2);
CONTENEDOR(1,1)=1;
CONTENEDOR(1,2)=1;
FILASN=1;

SALIR=0;

FILASE2=FILAS/2;

%---Encuentra punto de referencia para corte izquierdo
for CICLO_COLUMNAS=25:COLUMNAS
    for CICLO_FILAS=1:FILASE2
        PIVOT=A(CICLO_FILAS,CICLO_COLUMNAS);
        if PIVOT<1
            CONTENEDOR(FILASN,1)=CICLO_FILAS;
            CONTENEDOR(FILASN,2)=CICLO_COLUMNAS;
            FILASN=FILASN+1;
            SALIR=1;
            break
        end
    end
    if SALIR>0
        break
    end
end

%PARTE DE CORTE DE IMAGEN

%----- respuesta=CONTENEDOR;|
INICIO_CORTE_X=CONTENEDOR(1,2);

%FIN_CORTE_X=CONTENEDOR2(1,2)-CONTENEDOR(1,2);
FIN_CORTE_X=109;

IMAGEN_CORTADA=imcrop(A,[INICIO_CORTE_X 1 FIN_CORTE_X 180]);
respuesta=IMAGEN_CORTADA;
```

4.5 MEDICIONES DE DESEMPEÑO REAL DEL SISTEMA

Al realizar algunas pruebas sobre el rendimiento del prototipo implementado para demostrar el funcionamiento de un sistema de Vision Artificial se obtuvieron mediciones respecto a la cantidad de frascos que el sistema es capaz de analizar en un tiempo determinado; se obtuvieron los porcentajes de error y de aciertos que el sistema es capaz de proporcionar. Los datos son los siguientes:

Cantidad de frascos por minuto = 14

De las mediciones de desempeño realizadas se obtuvo la siguiente tabla:

Población	Frascos buenos	Frascos Malos	Resultados correctos	Resultados erróneos	Rendimiento	% error
50	25	25	46	4	92%	8%

Tabla 4.1. Estadísticas porcentuales de rendimiento del sistema

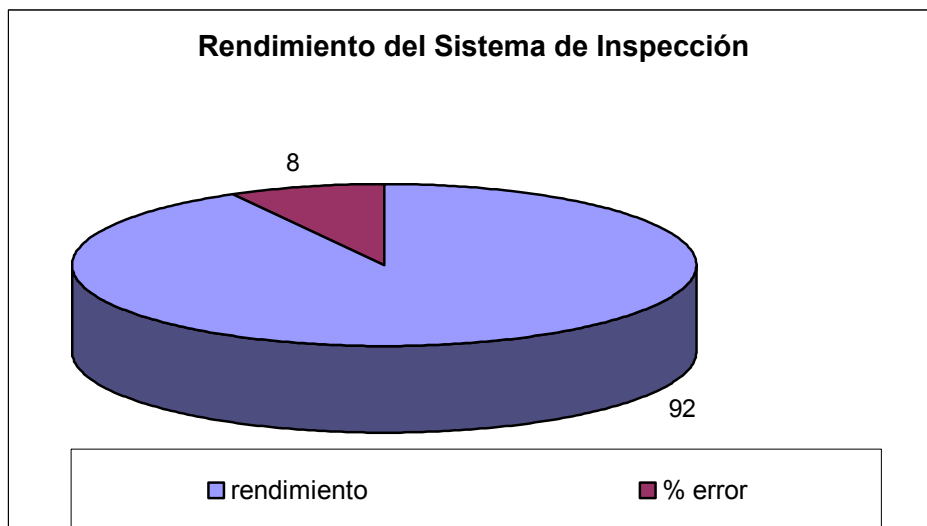


Gráfico 4.1 Distribución de Rendimiento del Sistema de Inspección

ANEXO A

A.1 MANUAL DE USUARIO

**MANUAL DE USUARIO
PARA
SISTEMA DE INSPECCION
BASADO EN VISION ARTIFICIAL**

Desarrollado por:

Pedro Ernesto Ortega Marroquín
Emerson Francisco Pocasangre León
Fernando Antonio Valle Figueroa

INTRODUCCION

El presente manual está diseñado para facilitar la utilización de la interfase gráfica del sistema de inspección basado en Visión Artificial, comprende la descripción de las distintas funciones del software y las capacidades de respuesta de la aplicación.

La aplicación está perfilada para facilitar la entrada de datos y la percepción de la información por parte del usuario, por lo que cuenta únicamente con funciones definidas en pantalla sin la necesidad de ingresar a menús que puedan incrementar el tiempo de respuesta de dicho usuario a una solicitud del sistema.

La aplicación cuenta con botones de tamaño considerable debido a que en entornos industriales se recomienda la utilización de 'touch screen' a fin de evitar la interacción con el sistema mediante un periférico (ratón, lápiz luminoso, teclado, etc.).

La aplicación cuenta con diferentes campos de información para mantener al usuario actualizado en cuanto a lo que está sucediendo durante el análisis de las imágenes de los frascos de vidrio.

CONDICIONES PARA LA INSTALACION DEL SOFTWARE

- Windows 2000 o Superior
- 512 MB de memoria RAM
- Microprocesador de 1.8 GHz o superior.
- Touch Screen (Recomendado)
- Matlab 6.5 o superior
- Toolbox de procesamiento de Imágenes
- Toolbox de redes neuronales
- Toolbox VFM (Vision For Matlab)
- Puerto USB
- Web cam
- Sistema de iluminación posterior difusa (Construido de acuerdo a las necesidades de la operación).

PERFIL DE USUARIO

- Conocimiento básico del sistema operativo Windows.
- Manejo del ratón
- Manejo de touch screen (opcional)
- Haber recibido capacitación para operar el sistema de inspección.

Nota: En caso de que el sistema presente alguna falla, esta será resuelta por los desarrolladores del software.

DESCRIPCION GENERAL DEL SISTEMA



- El sistema permite la entrada manual de la cantidad de frascos máxima a ser analizada.
- El sistema admite un conteo en tiempo real de los frascos analizados que es presentado en pantalla, incluye un conteo general en el campo del mismo nombre y un conteo específico de cantidad de producto analizado que se encuentra en buen estado indicado en el campo 'Frascos Buenos' y en otro campo llamado 'Frascos malos', el producto desechado.
- Se Cuenta con un botón de vista previa 'P', para visualizar el frasco anteriormente al proceso.

- Se cuenta con un botón de inicio de operaciones ‘I’, el cual permite iniciar el sistema de análisis, activando los sensores y la cámara además de las herramientas de manejo de imágenes del software.
- Existe también el botón de paro del sistema ‘F’, el cual resulta en la finalización de la operación inmediatamente a su activación, permitiendo que en un caso de error el sistema sea detenido.
- Puede observarse en un campo nombrado ‘IMAGEN ANALIZADA’, la toma en blanco y negro de la imagen que el sistema se encuentra analizando, permitiendo una consideración visual además de las consideraciones del sistema.

PROCEDIMIENTO PARA LA OPERACIÓN CON EL SISTEMA

1. Debe ejecutarse el acceso directo del software denominado ‘INSPECTOR’, aparecerá entonces la interfase gráfica desde la cual pueden manipularse las funciones de la aplicación.
2. Si se desea puede cambiarse la cantidad máxima de frascos a ser analizados en el campo ‘CANTIDAD MAXIMA’, debe escribirse específicamente números enteros, de lo contrario aparecerán mensajes de error indicando que no se ha escrito una cadena de entrada válida.
3. Seguidamente debe presionarse el botón ‘I’ de inicio de proceso, se observará entonces que el sistema inicia su operación activando la cámara, las señales de entrada y de salida para los dispositivos.

4. Si se comete un error durante el proceso, puede reiniciarse el sistema mediante el botón 'F', quedando inactiva la aplicación, la cual puede reanudarse si se presiona de nuevo el botón 'I'.
5. El sistema continúa con su operación hasta completar con el número de frascos indicado, al presionar el botón 'F', al cerrar la aplicación.

PROTOTIPO ENSAMBLADO



ANEXO B

EL TOOLBOX VISION FOR MATLAB

B.1 DESCRIPCIÓN Y MANEJO DEL TOOLBOX.

Este toolbox permite a Matlab, tener acceso a la fuente de video conectada e instalada en la computadora. Está implementado para su uso en el sistema operativo Microsoft Windows 95, 98, NT y XP, y utiliza el driver de Microsoft WDM Image Capture (WIN32) para obtener el control del dispositivo de video conectado al ordenador.

VFM permite una gran facilidad en el manejo de la cámara, incluso permite la modificación de sus propiedades de configuración como son:

- Resolución de la imagen
- Ajuste automático o manual de los parámetros de Imagen
- Determinar el dispositivo desde el que se adquirirá la imagen

En la figura B.1 se puede apreciar la pantalla principal que aparece al ejecutar este toolbox.

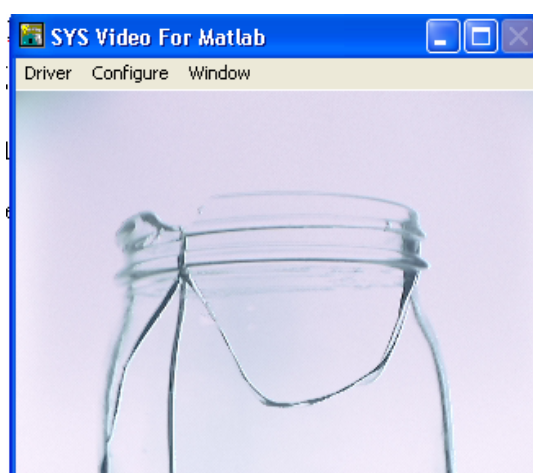


Figura B.1 Pantalla principal del Toolbox VFM.

Desde la barra de menús se puede acceder a las distintas opciones de configuración del dispositivo que se detallan a continuación:

- Driver (Controlador de video) (figura B.2): Selección del driver que se va a utilizar para la captura (en el caso de que hubiera varios instalados).

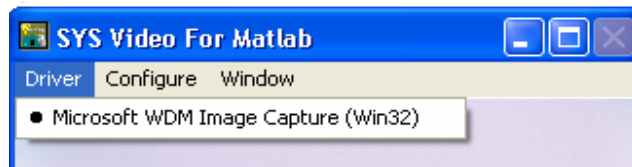


Figura B.2 Selección de Driver.

- Configure (Configuración)(figura 1.3): Este submenú consta de las siguientes:
 - Source: Abre el cuadro de diálogo de configuración de la fuente de video. En éste se podrá: seleccionar que entrada de video se quiere utilizar (en el caso de que se dispusiera de varias, tales como S-video, video compuesto,...); configurar la imagen procedente de esa fuente de video (brillo, matiz, saturación,...); fijar el número de fotogramas por segundo con que se va a realizar la captura; etc.
 - Format: Abre el cuadro de diálogo de configuración del formato de captura. En este cuadro se establece el tamaño de la imagen con que se va a trabajar y la profundidad y compresión en píxeles de la misma.

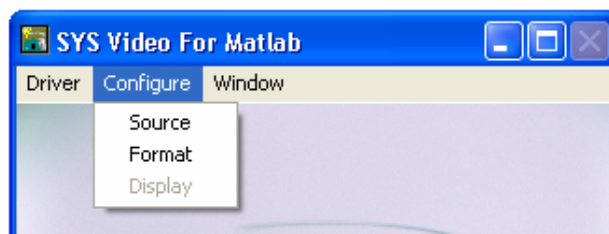


Figura B.3 Opciones de formato y configuración de la cámara.

- Window (Ventana) (figura B.4): Opciones de configuración de la ventana de visualización de video, tales como: modificar su relación de tamaño, seleccionar el modo *Preview* y ocultarla o mostrarla.

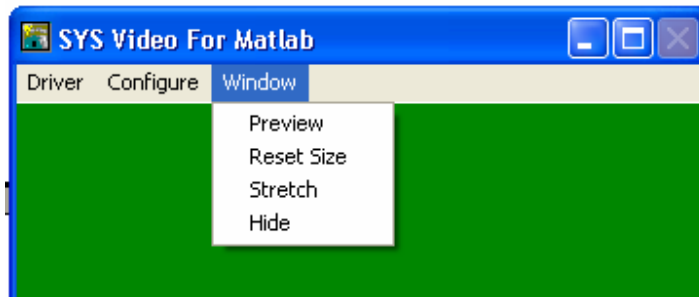


Figura B.4 Configuración de la ventana del toolbox.

El programa ofrece la posibilidad de acceder a sus funciones a través de una serie de comandos. A continuación se muestra una lista detallada, con su correspondiente funcionamiento, de todas las instrucciones que ofrece:

- **VFM('grab', N° imágenes)**

Devuelve una matriz $M \times N \times 3 \times N$ imágenes formado por elementos del tipo uint8 (enteros sin signo de 8 bit), donde M y N son el alto y ancho respectivamente que se han configurado para las imágenes capturadas. Las imágenes que devuelve están en formato RGB.

- **VFM('preview', bPreview)**

Alterna entre el modo de preview del driver de captura, dependiendo del valor de bPreview. Por defecto, bPreview toma el valor 1.

- **VFM('show', bShow)**

Muestra o esconde la ventana de captura dependiendo del valor de bShow. Esta ventana se muestra si el valor de bShow es 1 (valor que tiene por defecto).

- **VFM('configsource')**

Muestra el cuadro de diálogo de configuración de la fuente de video.

- **VFM('configformat')**

Muestra el cuadro de diálogo de la configuración del formato de captura.

- **VFM('configdisplay')**

Muestra el cuadro de diálogo de configuración de visualización.

- **VFM('drivername', índice)**

Devuelve el nombre de un driver del sistema para un índice dado. El índice debe estar en el rango 1-10. Si existe un driver para el índice introducido, se devuelve una cadena que contiene el nombre de dicho driver.

- **VFM('setdriver', índice)**

Establece el driver de video a utilizar por VFM dependiendo del valor del índice. El índice debe estar en el rango 1-10. Si un driver no existe para un índice dado se mostrará el correspondiente mensaje de advertencia.

1.- Ejemplo de realización de captura de una imagen:

Además de las instrucciones detalladas anteriormente, se utilizarán las que ofrece el Matlab para la lectura y almacenamiento de imágenes: *imread* y *imwrite*, respectivamente.

Así, a continuación se expone un ejemplo de captura y almacenamiento de una imagen y de lectura de la misma:

- **Captura y almacenamiento de una imagen**

```
>> Imwrite (vfm ('grab',1),'prueba.tif','tif')
```

Con esta instrucción se captura una imagen de la fuente de video y se almacena en disco en formato *tif* y con el nombre *prueba*.

- **Lectura de una imagen**

```
>>imagen = imread('prueba.tif','tif')
```

Con esta instrucción se lee del disco la imagen con formato *tif* y nombre *prueba* y se almacena en la variable *imagen*. Esta variable resultará una matriz de dimensión $M \times N \times 3$, donde M y N son las dimensiones (alto y ancho) de la imagen *prueba*.

ANEXO C

C.1 EL PUERTO PARALELO

Los puertos de comunicación de una computadora son de particular interés que le permiten utilizar una computadora personal para controlar todo tipo circuitos electrónicos utilizados, principalmente, en actividades de automatización de procesos, adquisición de datos, tareas repetitivas y otras actividades que demandan precisión. Debido a su facilidad de comunicación de éste será utilizado en el sistema de inspección basado en Vision Artificial.

Conceptos básicos

Existen dos métodos básicos para transmisión de datos en las computadoras modernas. En un esquema de transmisión de datos en **serie** un dispositivo envía datos a otro a razón de un bit a la vez a través de un cable. Por otro lado, en un esquema de transmisión de datos en **paralelo** un dispositivo envía datos a otro a una tasa de n número de bits a través de n número de cables a un tiempo. Sería fácil pensar que un sistema en *paralelo* es n veces más rápido que un sistema en *serie*, sin embargo esto no se cumple, básicamente el impedimento principal es el tipo de cable que se utiliza para interconectar los equipos. Si bien un sistema de comunicación en paralelo puede utilizar cualquier número de cables para transmitir datos, la mayoría de los sistemas paralelos utilizan ocho líneas de datos para transmitir un byte a la vez.

Descripción del conector físico

La conexión del puerto paralelo al mundo exterior se realiza mediante un conector hembra **DB25**.

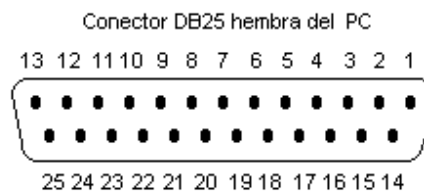


Fig. C.1 Conector DB25

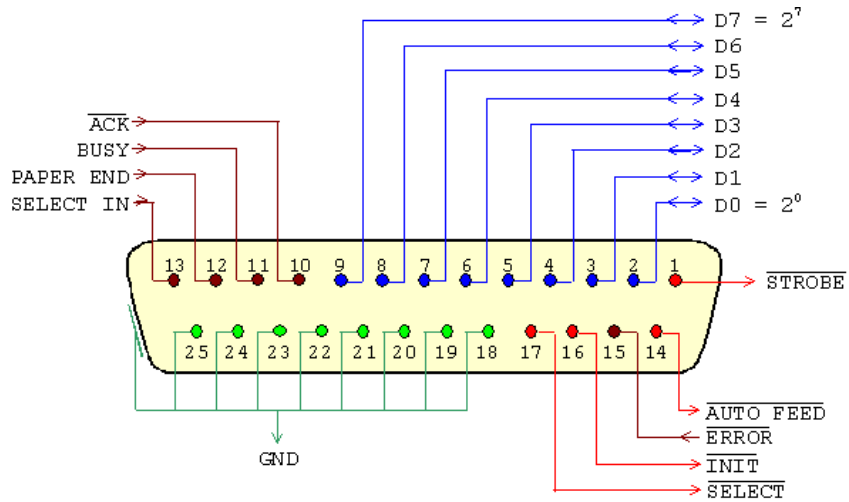


Fig. C.2 Descripción de los pines de salida del puerto paralelo

- **8 líneas (pines)** son para salida de datos (bits de **DATOS**). Sus valores son únicamente modificables a través de software, y van del pin 2 (dato 0, D0) al pin 9 (dato 7, D7).
- **5 líneas** son de entrada de datos (bits de **ESTADO**), únicamente modificables a través del hardware externo. Estos pines son: 11, 10, 12, 13 y 15, del más al menos significativo.
- **4 líneas** son de control (bits de **CONTROL**), numerados del más significativo al menos: 17, 16, 14 y 1. Habitualmente son salidas, aunque se pueden utilizar también como entradas y, por tanto, se pueden modificar tanto por software como por hardware.
- las **líneas de la 18 a la 25** son la tierra.

En la siguiente tabla se detallan la nomenclatura y descripción de cada línea. La columna "Centronics pin" se refiere a las líneas del conector tipo Centronics usado en las impresoras. La columna E/S se refiere al dato visto desde el lado del PC.

DB25 pin	Centronics pin	Tipo (E/S)	Señal	Descripción
1	1	S	Strobe	Si está bajo más de 0.5 μ s, habilita a la impresora para que reciba los datos enviados.
2	2	S	D0	Bit 0 de datos, bit menos significativo (LSB)
3	3	S	D1	Bit 1 de datos
4	4	S	D2	Bit 2 de datos
5	5	S	D3	Bit 3 de datos
6	6	S	D4	Bit 4 de datos
7	7	S	D5	Bit 5 de datos
8	8	S	D6	Bit 6 de datos
9	9	S	D7	Bit 7 de datos, bit más significativo (MSB)
10	10	E	Ack	Un pulso bajo de $\sim 11 \mu$ s indica que se han recibido datos en la impresora y que la misma está preparada para recibir más datos.
11	11	E	Busy	En alto indica que la impresora está ocupada.
12	12	E	PaperEnd	En alto indica que no hay papel.
13	13	E	SelectIn	En alto para impresora seleccionada.
14	14	S	AutoFeed	Si está bajo, el papel se mueve una línea tras la impresión.
15	32	E	Error	En bajo indica error (no hay papel, está fuera de línea, error no det.).
16	31	S	Init	Si se envía un pulso en bajo $> 50 \mu$ s la impresora se reinicia.
17	36	S	Select	En bajo selecciona impresora (en gral. no se usa, ya que SelectIn se fija a alto).
18-25	19-30,33		GND	Masa retorno del par trenzado.
18-25	16			Masa lógica
18-25	17			Masa chasis

Tabla C.1 Nomenclatura y Descripción de las líneas

El nombre de cada señal corresponde a la misión que cumple cada línea con relación a la impresora, el periférico para el que fue diseñado el puerto paralelo. Las señales activas a nivel bajo aparecen con la barra de negación (por ejemplo, Strobe).

Cuando se indica alto o bajo se refiere a la tensión en el pin del conector. Alto equivale a ~5V en TTL y bajo a ~0V en TTL.

ESPECIFICACION DE PINES UTILIZADOS EN EL HARDWARE DEL SISTEMA DE INSPECCION

En total se utilizan cinco pines de los disponibles del puerto paralelo que fueron detallados anteriormente, a continuación se presenta la función de cada pin usado:

- Pin 2 (D0): este es utilizado para indicar que la respuesta del sistema al clasificado actual ha sido “FRASCO MALO” y es representado por la activación de un LED rojo.
- Pin 3 (D1): este es utilizado para indicar que la respuesta del sistema al clasificado actual ha sido “FRASCO BUENO” y es representado por la activación de un LED verde.
- Pin 4 (D2): este se utiliza para la activación o desactivación del motor paso que dirige la banda transportadora de frascos. Esta función se realiza cambiando el estado del pin de 0 (activación) a 1 (desactivación).
- Pin 10 (ACK): Este es el único pin que es utilizado como entrada al sistema y realiza la función de indicarle a éste la presencia del frasco, esto se realiza por medio de un sensor óptico que cuando presenta un 1 existe presencia de frasco y cuando cambia al estado 0 indica que no hay presencia de frasco.
- Pin 18 – 25 (GND): se puede utilizar cualquier pin del 18 al 25 para representar la señal de tierra de las señales del puerto paralelo.

ANEXO D

D.1 Hardware Utilizado (Diagramas del prototipo)

Motores paso a paso

Un motor paso a paso es un tipo especial de motor sincrónico diseñado para rotar un determinado número de grados por cada pulso eléctrico que recibe por su unidad de control. Los pasos habituales son 7.5° o 15° por pulso. Estos motores se usan en muchos sistemas de control, puesto que la posición de un eje o de otra pieza mecánica puede controlarse de modo preciso mediante ellos. En la figura D.1 aparece un motor sencillo paso a paso junto con su unidad de control. Conviene observar la figura D.2 para comprender el funcionamiento de un motor de este tipo; allí aparece un estator trifásico bipolar con rotor de imán permanente. Si se aplica un voltaje de cc a la fase a del estator sin aplicar ningún voltaje a las fases b y c, entonces en el rotor se inducirá un momento de torsión que lo hace alinearse con el campo magnético BS, del estator, como se observa en la figura D.2-b.

Ahora supóngase que se apaga la fase a y que se aplica en la fase c un voltaje de cc negativo. El nuevo campo magnético del estator ha girado 60° con respecto al campo magnético previo, y el rotor del motor lo imita. Continuando este modelo, es posible construir una tabla que muestre la posición del rotor en función del voltaje aplicado al estator. Si el voltaje producido por la unidad de control cambia con cada pulso de entrada, según aparece en la Tabla D.1, entonces el motor paso a paso avanzará 60° con cada pulso de entrada.

Es fácil construir un motor paso a paso con intervalo de paso mejor mediante el aumento del número de polos del motor. La cantidad de grados mecánicos que corresponde a determinada cantidad de grados eléctricos es

$$\theta_m = (2/P) \theta_e \quad \text{Ec. D.1}$$

Puesto que cada paso de la Tabla D.1 corresponde a 60 grados eléctricos, la cantidad de grados mecánicos movidos por paso disminuye al aumentar el número de polos. Por ejemplo, si el motor paso a paso tiene 8 polos, entonces el ángulo mecánico del eje del motor cambiará a 15° por paso.

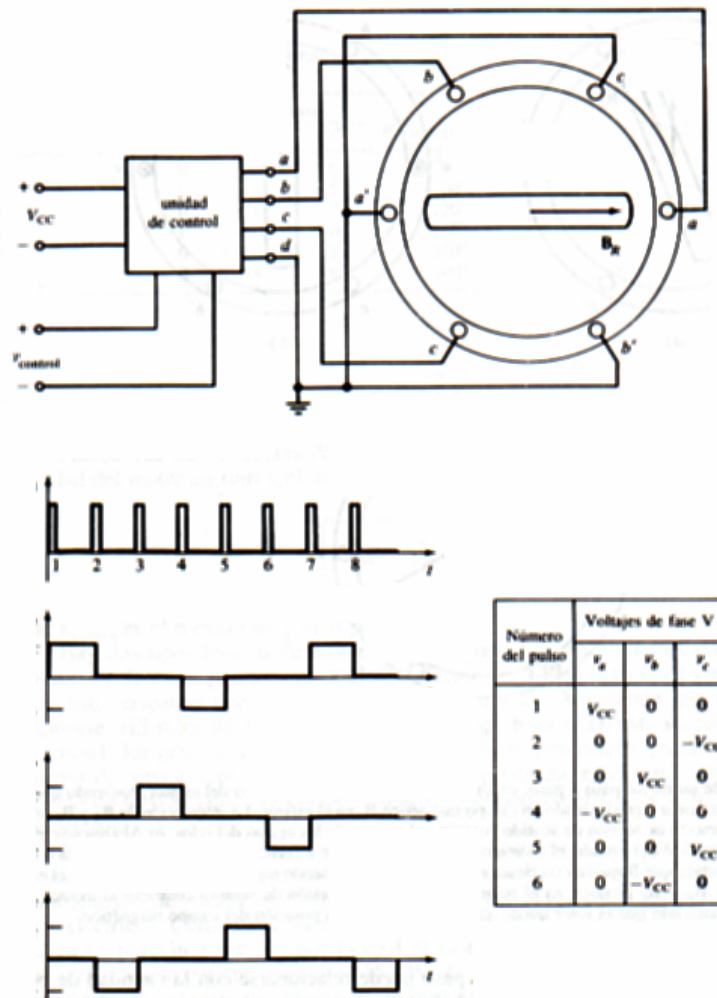


Fig. D.1

a) Motor sencillo de paso trifásico con su respectiva unidad de control. Las entradas a la unidad de control constan de una fuente de potencia de cc y una señal de control que consiste en una sucesión de pulsos. b) Muestra del voltaje de salida de la

unidad de control en función de una serie de pulsos de control. c) Tabla que muestra el voltaje de salida de la unidad e control en función del número de pulsos.

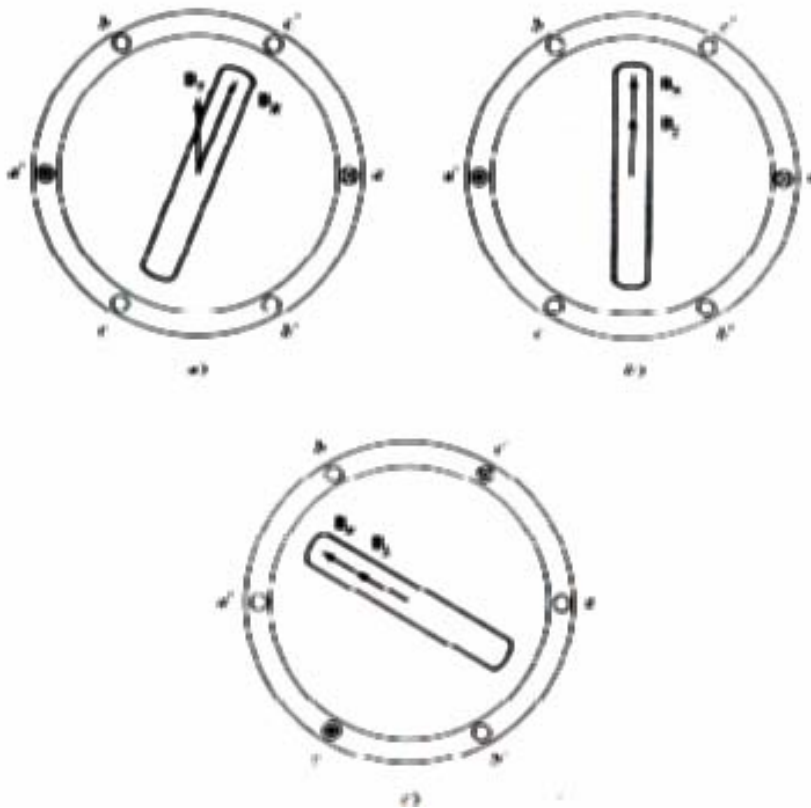


Fig. D. 2

Funcionamiento de un motor paso a paso: a) Un voltaje V se aplica a la fase a del estator, haciendo que fluya una corriente en la fase a y produciendo un campo magnético BS en el estator. La interacción de BR y BS produce en el rotor un momento de torsión de sentido contrario al avance de las agujas del reloj. b) Al alinearse el rotor con el campo magnético del estator, el momento de torsión neto cae a cero. c) Un voltaje $-V$ se aplica a la fase c del estator, haciendo que fluya una corriente en la fase c y produciendo un campo magnético BS en el estator. La interacción de BR y BS produce en el rotor un momento de torsión de sentido contrario al avance de las agujas del reloj, haciendo que el rotor

quede alineado con la nueva posición del campo magnético.

La velocidad de un motor paso a paso puede relacionarse con la cantidad de pulsos que hay en su unidad de control por unidad de tiempo utilizando la ecuación D.1; esta ecuación da el ángulo mecánico de un motor paso a paso en función del ángulo eléctrico. Si los dos lados de esta ecuación están diferenciados con respecto al tiempo, entonces tenemos una relación entre las velocidades eléctrica y mecánica de rotación del motor:

$$\omega_m = (2/P) \omega_e \quad \text{Ec. D.2-a}$$

TABLA D. 1

Posición del rotor en función del voltaje en un motor paso a paso bipolar

Numero de pulso de entrada	Voltajes de Fase			Posición del rotor
	a	b	c	
1	V	0	0	0°
2	0	0	-V	60°
3	0	V	0	120°
4	- V	0	0	180°
5	0	0	V	240°
6	0	- V	0	300°

O

$$n_m = (2/P) n_e \quad \text{Ec. D.2-b}$$

Puesto que hay 6 pulsos de entrada por revolución eléctrica, la relación entre la velocidad del motor en rpm y el número de pulsos por minuto es

$$n_m = (1/3P) n_{\text{pulsos}} \quad \text{Ec. D.3}$$

donde n_{pulsos} es el número de pulsos por minuto.

Hay dos tipos básicos de motores paso a paso, que sólo difieren en la disposición del rotor: el de *imán permanente* y el de *reluctancia*. El primero tiene un rotor de imán

permanente, mientras que el segundo tiene un rotor ferromagnético que no es un imán permanente. (El rotor del motor de reluctancia antes descrito en esta sección es el de tipo reluctancia). En general, el motor paso a paso de imán permanente puede producir mayor momento de torsión que el de tipo reluctancia, puesto que el de imán permanente tiene momento de torsión tanto del campo magnético permanente del rotor como de los efectos de reluctancia.

Generalmente los motores paso a paso de tipo reluctancia se construyen con un embobinado de estator de cuatro fases en vez del embobinado trifásico descrito anteriormente. El embobinado de estator de cuatro fases reduce los pasos entre los pulsos, de 60 a 45 grados eléctricos. Como ya se mencionó, el momento de torsión de un motor de reluctancia varía en proporción a $\text{Sen } 2\delta$, por lo cual el momento de torsión de reluctancia entre los pasos será máximo para un ángulo de 45° . Por tanto, un determinado motor paso a paso de tipo reluctancia puede producir mayor momento de torsión con un embobinado de estator de cuatro fases que con uno trifásico.

La ecuación D.3 puede generalizarse con el fin de aplicarla a todos los motores paso a paso, independientemente del número de fases de los embobinados de estator. En general, si un estator tiene N fases, se requieren $2N$ pulsos por revolución eléctrica en tal motor. Por consiguiente, la relación entre la velocidad del motor y rpm y el número de pulsos por minuto es

$$n_m = (1/NP) n_{\text{pulsos}} \quad \text{Ec. D.4}$$

Los motores paso a paso se utilizan frecuentemente en sistemas de control y posición porque el computador que efectúa el control puede saber la *velocidad* y la *posición* exactas del motor paso a paso sin necesitar información de retomo del eje del motor. Por ejemplo, si un sistema de control envía 1,200 pulsos por minuto al motor bipolar paso a paso de la figura D.1, entonces la velocidad del motor será exactamente

$$\begin{aligned} n_m &= (1/3P) n_{\text{pulsos}} \\ &= (1/6)(1200 \text{ pulsos/ min}) \\ &= 200 \text{ rpm} \end{aligned}$$

Además, si se conoce la posición inicial del eje, entonces el computador puede

determinar el ángulo exacto del eje del rotor en cualquier momento futuro mediante el simple conteo del número total de pulsos que éste haya enviado a la unidad de control del motor paso a paso.

Secuencia Lógica Del Motor.

Como se había explicado anteriormente, un motor paso a paso necesita una secuencia lógica para realizar el giro de su eje, dicha secuencia puede ser de tres distintas configuraciones, las cuales son: velocidad, fuerza y velocidad-fuerza; para el presente proyecto cualquiera de las tres configuraciones es válida; pero la que se utilizara será la primera ya que su diseño es más fácil y su circuito es mucho más sencillo.

La secuencia lógica de velocidad se muestra a continuación:

0001	1000
0010	0100
0100	0010
1000	0001

En la cual se observa que para que el motor realice el respectivo giro lo que se necesita es rotar un bit de derecha a izquierda o viceversa.

Esta secuencia lógica es lograda con el uso de un dé multiplexor; el IC 74LS139, cuyo funcionamiento es mostrado por su tabla de verdad:

G	B	A	Y3	Y2	Y1	Y0
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

Se observa que la secuencia:

1110
1101
1011
0111

no es la vista anteriormente que necesita el motor; sino lo inverso, lo cual no es ningún problema ya que se puede invertir estas señales con la lógica TTL.

Explicada la salida de dicho IC; se debe mencionar que para que este realice lo anterior; sus entras (A, B) deben poseer en conteo binario de cero a tres. Para obtener este conteo se hará uso del IC 74 LS 191; el cual es un contador binario de cuatro bits up-down.

Pero como se observa dicho dispositivo de cuatro bits, su respectivo conteo es de cero a quince; y, como solo se requiere el conteo de cero a tres se limitará a utilizar los dos bits menos significativos.

Para que este dispositivo realice lo explicado anteriormente es necesario que en su entrada de reloj se le proporcione un tren de pulsos, el cual es producido por un multivibrador astable (IC 555).

Dicho circuito se muestra a continuación:

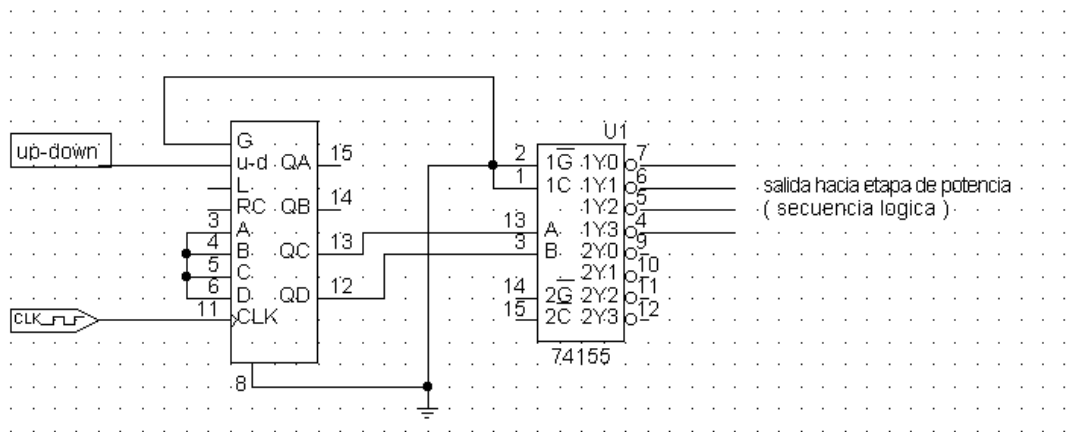


Fig. D.3 Circuito que realiza secuencia lógica del motor paso a paso.

Etapa De Potencia.

Una vez obtenida la secuencia lógica que se necesita para hacer que el motor gire; es importante darse cuenta que un motor paso a paso necesita mucho mas corriente que la que el circuito controlador maneja, es por esto que se requiere utilizar un circuito especial que amplifique la corriente de la salida de la secuencia lógica a la que se necesita para poder excitar las respectivas bobinas del motor.

Esta amplificación es lograda por una configuración de transistor; la cual se denomina Par Darlington y se interconecta de la siguiente manera:

Es importante saber que las bobinas del motor debido a la inducción magnética producen corrientes en sentido opuesto a la de circulación en el circuito, es por esto que es necesario conectar un diodo polarizado en reversa con la bobina del motor, así se asegura que ninguna corriente dañe a la demás circuiteria interna.

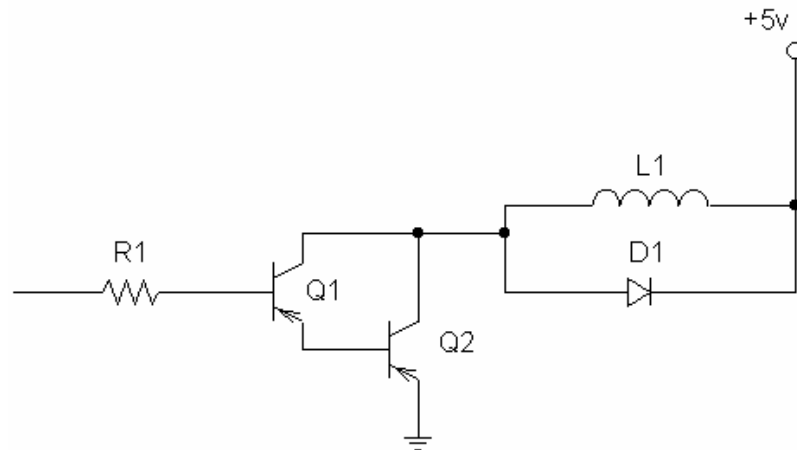


Fig. D.4 Circuito amplificador de corriente de los pulsos para el motor paso.

CONCLUSIONES

- El proyecto ha dado la pauta para implementar sistemas de captura y tratamiento de información utilizando la visión artificial.
- Se construyó un sistema capaz de identificar anomalías en un frasco a nivel superficial y por medio de Visión Artificial clasificarlo como bueno o malo.
- El sistema construido posee una facilidad de manejo para el usuario pues básicamente el operario solo debe dar inicio al proceso y obtener al final los resultados del conteo y clasificación de los frascos.
- El sistema elimina la posibilidad de errores humanos durante la verificación de los frascos.

ALCANCES Y RECOMENDACIONES PARA SUPERAR LAS LIMITANTES

- La velocidad del sistema depende del hardware utilizado, es decir que al utilizar dispositivos como cámaras, Microprocesador de mayor velocidad y bandas más especializadas, pueden optimizarse los resultados finales.
- Al utilizar versiones superiores a la versión 6.5 de MATLAB, las cuales incluyen toolbox de adquisición de imágenes, se supera la utilización obligatoria del toolbox 'Vision For Matlab' y se aumenta la velocidad de procesamiento de la imagen produciendo un rendimiento mayor del sistema.
- Este sistema ha sido diseñado para analizar un tipo de frasco, sin embargo podría llegar a realizar el análisis de diversos modelos, implementando entrenamientos previos y agregando campos de selección de tipo de frasco
- Se podría incrementar la velocidad del sistema agregando luz estroboscópica para evitar efectos de sombra en la imagen adquirida, eliminando la necesidad de detener el frasco para tomar su imagen.
- El sistema puede reprogramarse con la finalidad de analizar una determinada área del frasco.

LIMITACIONES

- El sistema no funciona para analizar frascos opacos.
- La rosca del frasco de vidrio no puede ser analizada debido a sus multiformes características.
- El rendimiento del sistema depende directamente del hardware utilizado.
- La cámara de adquisición de la imagen, no puede ser afectada por rayos de luz que impacten de forma directa sobre el lente, debido a la sensibilidad que debe existir hacia la luz proveniente de las lámparas de iluminación del sistema.
- El sistema necesita al menos 3 cámaras para cubrir los 360° de circunferencia del frasco.
- El sistema está diseñado para analizar frascos en serie.

REFERENCIAS BIBLIOGRAFICAS

- ***Introducción A la visión por computador***

Universidad Carlos III de Madrid

Arturo de la Escalera Hueso

José María Armingol Moreno

Área de ingeniería de Sistemas y Automática.

2ª Edición, MCMXCVIII, Madrid, España.

- ***Tratamiento Digital de Voz e Imagen y Aplicación a la Multimedia***

Marcos Faúndez Zanuy

Alfaomega Grupo Editor

Marcombo S. A. Barcelona, España.

December 1998, Release 0.1 1998/12/16

REFERENCIAS DE INTERNET

- <http://www2.cmp.uea.ac.uk/~fuzz/vfm/default.html>

En este sitio se encuentra el link para bajar el Toolbox VFM para MATLAB;

además se encuentra una breve reseña de las principales funciones de este interesante Toolbox.

- <http://mailweb.udlap.mx/~sp098974/tesislic/cap3.pdf>

Este sitio contiene diferente información acerca de los componentes principales de un sistema de visión artificial. Acá se puede encontrar la aplicación práctica de un brazo robótico.

- <http://www.lfcia.org/~cipenedo/cursos/scx/Tema6/nodo6-2.html>

En este sitio se puede encontrar la descripción y funcionamiento en general de la red neuronal ART1S

<http://www.lfcia.org/~cipenedo/cursos/scx/Tema1/nodo1-1.html>

<http://www.monografias.com/trabajos/redesneuro/redesneuro.shtml>

Diferentes links donde se puede obtener la información de los principios y características básicas de las redes neuronales artificiales, así como también los principios de funcionamiento de estas.

<http://www.modelo.edu.mx/univ/virtech/circuito/superior#superior>

Breve reseña del puerto paralelo.