



## CAPITULO III

En este capítulo se presentan los métodos y algoritmos utilizados para el desarrollo del sistema de procesamiento de imágenes para controlar un robot manipulador y se describen en la forma como serán utilizados dentro del proyecto.

### 3.1 PROCESAMIENTO DE IMÁGENES.

Un área muy ligada a la de visión computacional es la de procesamiento de imágenes. Aunque ambos campos tienen mucho en común, el objetivo final es diferente.

#### **Objetivo de Procesamiento de Imágenes**

El objetivo de procesamiento de imágenes es mejorar la calidad de las imágenes para su posterior utilización o interpretación. [Escalera 01]

### 3.2 VISIÓN COMPUTACIONAL

*Su función principal es reconocer y localizar objetos en el ambiente mediante el procesamiento de las imágenes.*

La visión computacional es el estudio de los procesamientos de imágenes, para entenderlos y construir máquinas con capacidades similares a las humanas. [Sucar 03]

#### **Objetivo de la Visión Computacional**

*El objetivo de la visión computacional es extraer características de una imagen para su descripción e interpretación por la computadora.*

La base del software de un sistema de visión es la interpretación y análisis de los píxeles. El resultado final puede ser, desde la medida de una partícula, a la determinación o lectura de una serie de caracteres (OCR), pasando por cualquier otro proceso que podamos imaginar sobre las imágenes.

Dependiendo de si la aplicación se realiza en entorno industrial o científico los pasos a seguir en un sistema de visión serán algo distintos.

### **Aplicación Industrial**

- Captura de la Imagen
- Definición de la región de interés donde se realizarán las medidas
- Inicialización de las tolerancias para determinar si la pieza a determinar es o no correcta
- Ejecutar las medidas
- Generar una salida apropiada

### **Aplicación Científica**

- Capturar la Imagen
- Hacer un proceso de mejora
- Determinar los elementos a medir
- Ejecutar la medida
- Almacenar las medidas y realizar procesos gráficos o estadísticos

Mientras que en las aplicaciones industriales la velocidad a la que se realizan las medidas es fundamental, ya que se deben evaluar todas las piezas producidas en tiempo real, en las aplicaciones científicas se busca la determinación de los resultados en imágenes más complejas.

### 3.3 COMPONENTES DE UN SISTEMA DE VISIÓN

El buen desempeño de un sistema de visión depende en gran parte de sus componentes que lo forman, existiendo 6 partes primordiales para que el sistema funcione adecuadamente.

**Captación:** 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

**Segmentación:** Es el proceso que divide a un imagen en objetos que sean de nuestro interés.

**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 asocia un significado a un conjunto de objetos reconocidos.

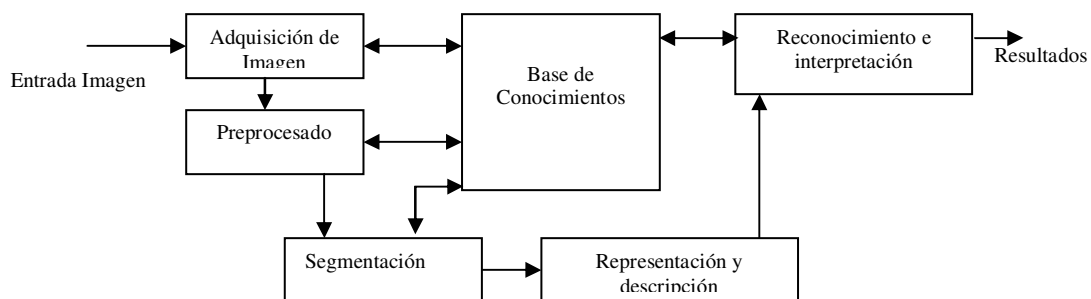


Figura 3.1: Etapas fundamentales de un sistema de Visión Artificial. [Chávez 2001]

Un sistema de visión artificial toma información útil acerca de una escena a partir de su proyección en dos dimensiones.

Existen herramientas que ayudan a tomar una decisión de los métodos y algoritmos que se aplican a los sistemas de procesamiento de imágenes, debido a que la información grafica no siempre responde a resultados de métodos o algoritmos definidos, esto es por que cada sistemas de procesamiento es distinto y las aplicaciones son variadas. Los métodos analizados fueron tomados en base a una propia herramienta de procesamiento de imágenes, sin embargo existen herramientas comerciales que ayudan a tomar una decisión como Wit, NeatVisión, etc. Para mayo información consulta el apéndice A.

### **3.4 ANÁLISIS DE MÉTODOS DE PROCESAMIENTO DE IMÁGENES PARA CONTROLAR UN ROBOT MANIPULADOR**

Inicialmente se conoce que se trabaja ahora directamente en la imagen a color, el objetivo es mejorarla para su interpretación visual o para alimentarla a los niveles superiores de visión. Normalmente se pueden aplicar las técnicas para imágenes monocromáticas a imágenes a color, aplicándose a cada componente. Sin embargo, hay casos en que si se hace esto directamente, pueden existir problemas y no obtenerse exactamente el efecto deseado. Ejemplos de esto son la aplicación de ecualización por histograma y la detección de orillas en imágenes a color.

Los algoritmos implementados en este proyecto están definidos dentro del área de visión activa, debido a que trabajan sobre escenas activas del ambiente desarrollado por el robot. A continuación se presenta los algoritmos que se utilizan dentro del sistema de procesamiento de Imágenes para controlar un robot manipulador.

La secuencia como se van presentando los métodos de procesamiento de imágenes son la forma en que se van aplicando al sistema de control del manipulador. Se inicia con un método de preprocesamiento de imagen.

**Método de Preprocesamiento de imagen.**

**Segmentación:** Es el proceso en que se divide una escena percibida en sus partes para extraer objetos para su posterior reconocimiento y análisis. Por lo que se ha una subdivisión del espacio de parámetros llamados células de acumulación donde  $(a_{max}, a_{min})$  y  $(b_{max}, b_{min})$  son los intervalos.

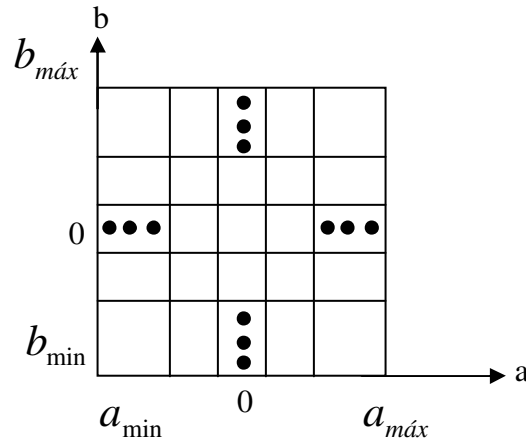


Figura 3.2: División del plano de parámetros en células. [Gonzalez y Woods 92]

La imagen es dividida en una matriz de 3x3, inicialmente se desarrolla este tipo de segmentación, para disminuir problemas de confusión de piezas. Normalmente este uso de segmentación en células acumulativas se implementa en algoritmos básicos de segmentación de tipo similitud como es la transformada de Hough. El tipo de segmentación utilizado es Segmentación General.

**Métodos de procesamiento de imágenes**

La mayoría de procesamiento de imágenes se aplica directamente sobre imágenes en escala de grises, debido al bajo consumo de cómputo de éste; la mayoría de métodos matemáticos deterministas que se utilizan en su procesamiento, están basados en la diferencia de niveles

de grises, por lo que no existen muchos métodos para procesamiento de imágenes en color, aun cuando estos pueden ser utilizados en este tipo de formato de imagen.

En este proyecto plantea utilizar métodos matemáticos estadísticos para el procesamiento de imágenes en color, con el fin de ser demostrables para ser reutilizados en otras investigaciones.

### Obtención de características del objeto por Color

Existen varias representaciones o modelos de color. Estos modelos se dividen en dos clases de modelos. Unos son modelos sensoriales y otros se denominan modelos perceptuales.

[Sucar 03]

La idea principal de Obtener Características por Color es sacar datos relevantes en formato RGB de objetos, para crear características que puedan ser utilizadas por otros procesos para llegar a un resultado.

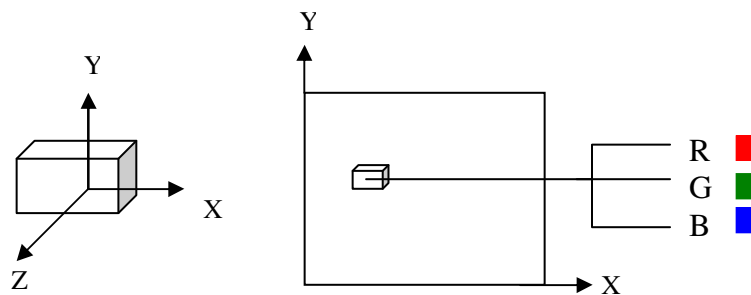


Figura 3.3: Obtención de características del objeto por color [Starostenko 2003]

Por la naturaleza de la captura de la imagen, se trabaja en formato RGB, por lo cual se trabaja con un formato sensorial en lugar del un formato de percepción como el HSI.

Además se pueden aplicar las mismas técnicas que se utilizan en imágenes monocromáticas. Para esto se toma cada componente (R,G,B) como una imagen

monocromática y se aplica algún operador o método a cada una independientemente. Después se combinan todas (considerando normalmente el máximo o el promedio).

### ***Modelo Sensorial***

El modelo RGB es el modelo básico que utiliza las componentes primarias rojo, verde y azul, normalizadas. En este modelo se basan las cámaras y receptores de televisión. Sin embargo se tienen problemas al aplicarlo al procesamiento de imágenes (ecualización).

### ***Modelo Perceptual.***

Modelo HSI (Hue, Saturation, Intensity) se puede ver como una transformación del espacio RGB al espacio perceptual. En principio los modelos perceptuales deben ser mejores ya que nosotros detectamos los cambios en estas componentes. Sin embargo, es compleja la implementación de la detección de orillas en croma por no ser lineal.

Otra alternativa es definir técnicas especiales para detección de orillas en imágenes a color.

El procesamiento de imágenes a color es relativamente reciente, por los altos requerimientos de memoria y cómputo necesarios.

### **Procesos para Detección de Bordes:**

Los contornos caracterizan los límites de un objeto. Un contorno se caracteriza por presentar una transición de claro a oscuro o viceversa. Por esto se presenta un método para detectar bordes como Sobel.

### ***Sobel***

Se detecta el borde, calculando el gradiente de la imagen en dos direcciones ortogonales.

El operador utilizado es Sobel, el cual se presenta de la siguiente manera.



$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
--	--

Tabla: Operador Gradiente Sobel

Una plantilla detecta contornos horizontales y la otra verticales, obteniendo dos imágenes de gradientes  $I_{g1}$ ,  $I_{g2}$ .

Para determinar si en un píxel determinado hay un contorno o no, se define un umbral a partir del cual se considera la existencia de contorno:

La imagen de contornos  $I_c(x,y)$  se puede formar a partir de:

$$I_c(x,y) = \begin{cases} 1 & \text{si } |I_{g1}(x,y)| + |I_{g2}(x,y)| > \text{umbral} \\ 0 & \text{otro caso} \end{cases} \quad (2.13)$$

**Detección de bordes en color**

Este método se aplica a cada una de las componentes básicas de RGB, como si se tratara de una imagen monocromática, y se hace una combinación de todas ellas, considerándose el máximo o el promedio, sin embargo puede presentar orillas que no impliquen un cambio fuerte en ninguna componente, pero si son notables en color o saturación. Además este método se plantea para detectar bordes de piezas no convexas y convexas.

Una forma alternativa para calcular el borde de objetos en imágenes en color, es hacer una transformación de formato RGB a HSI, y aplicar cualquier método matemático estadístico para la detección de bordes, como Sobel o Robert, sin embargo se aplica a cada componente de color HSI, y después se combinan todas las orillas detectadas, considerando normalmente el máximo o promedio.

La transformación de RGB a HSI se realiza mediante la siguiente ecuación:

$$H = \cos^{-1} \left( \frac{\frac{1}{2}(R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right) \quad (2.14)$$

$$S = 1 - \left( \frac{3 \min(R, G, B)}{R + G + B} \right) \quad (2.15)$$

$$I = \frac{1}{3}(R + G + B) \quad (2.16)$$

Si  $B > G : H = 2\pi - H$

La intensidad (I) y saturación (S) están normalizada (entre cero y uno) y el croma (H) esta entre 0 y 360 grados.

***Distancia de Color entre Píxeles***

Se toma la distancia de cada píxel a sus vecinos (máscara de 3x3), se suman y se normalizan (dividir entre 8). De esta forma la “magnitud” de la orilla aumenta al aumentar la diferencia en intensidad, croma o saturación.

$$d = \left[ (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2 \right]^{1/2} \quad (2.17)$$

Ó

$$d = \left[ \text{abs}(R_1 - R_2)^2 + \text{abs}(G_1 - G_2) + \text{abs}(B_1 - B_2) \right] \quad (2.18)$$

***Convex Hull:***

La envolvente convexa, también denominada cierre convexo o *convex hull*, es uno de los más fundamentales constructores geométricos. El problema de computar un *convex hull* no

sólo está centrado en aplicaciones prácticas, sino también es un vehículo para la solución de un número de cuestiones aparentemente sin relación con él, que surgen en la Geometría Computacional y tiene aplicaciones, como por ejemplo, en el procesado de imágenes y en localización.

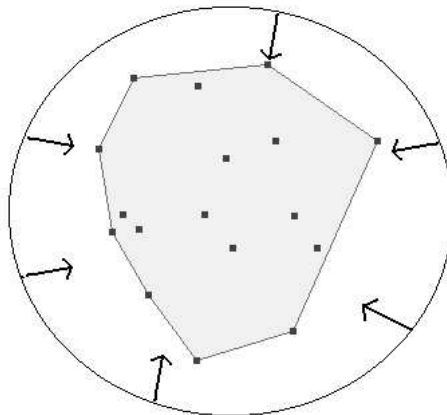


Figura 3.4: Idea intuitiva del significado del cierre convexo [Cormen 2001]

Desafortunadamente, no es posible construir la definición intuitiva de cierre convexo de forma natural, por lo que hay que definir un algoritmo, que en este caso es el QuikHull. [Cormen 2001]

**Algoritmo Quick Hull.**

La idea del Quick Hull es ir descartando lo más pronto posible los puntos que no formarán parte de la frontera del cierre convexo, que suelen ser los más interiores de la nube de puntos. El algoritmo Quick Hull calcula la envolvente convexa en tiempo  $O(n^2)$ .

Se aplica este método debido a que las piezas con las que se trabaja, son convexas y se utiliza para generar una figura conexa dentro del software de esta forma se tiene una figura definida, y se puede calcular el centro de masa de figura

**Etiquetamiento:**

Cuando se hace una búsqueda de un objeto por color se busca su valor de umbral y una vez encontrado, se modifica cambiándolo a un valor definido. De esta forma aparecen valores homogéneos definidos en la imagen.

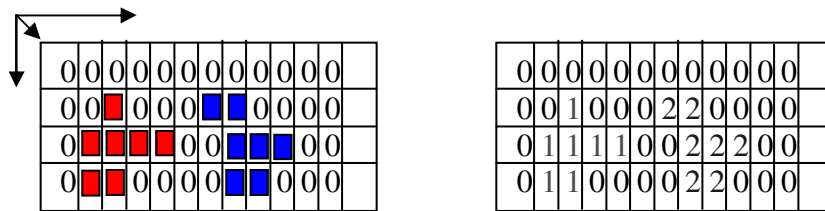


Figura 3.5: Algoritmo de Etiquetamiento.

De esta forma a cada valor homogéneo se le asigna un valor o etiqueta, de esta forma se puede conocer el número de objetos existentes en cada célula y de esta forma se identifica dentro del sistema, siendo identificados todos los objetos totalmente.

Una vez en funcionamiento los métodos de procesamiento de imágenes, generan una salida específica en una matriz de 3x3. Esta hace una interacción con el juego del Tic Tac Toe, el cual toma la matriz y actualiza su resultado.

A continuación se presenta el análisis del juego del Tic Tac Toe, el cual genera una salida, la cual es convertida a un conjunto de reglas o configuraciones que son enviadas para controlar el robot manipulador Lego MindStorms.

**3.5 TIC TAC TOE**

Una vez que se actualiza la matriz, se pasa a un modulo de juego, que es la implementación del tan conocido tic-tac-toe, este solamente responde como si tuviera un jugador normal.

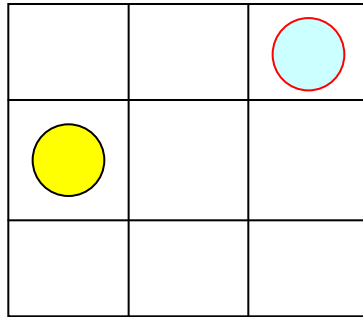


Figura 3.6: Ejemplo de Juego Tic Tac Toe

La implementación de este tipo de juegos puede ser desarrollada para que se ejecute en forma ingenua o en forma inteligente.

#### **Algoritmo Ingenuo.**

Cuando se implementa el juego del tic tac toé en forma ingenua, se generan los movimientos de la computadora en forma aleatoria, y solamente se checa quien gano, las probabilidades de que la computadora gane de esta forma son mínimas.

#### **Algoritmo inteligente.**

En este caso se implementó en forma inteligente, para lo cual se llevo a cabo el desarrollo de un algoritmo que utiliza un árbol de juego Minimax, este algoritmo es utilizado para encontrar las mejores soluciones a este tipo de juego.

#### ***Minimax***

El algoritmo Minimax consiste en establecer el compromiso para las mejores jugadas de ambos contendientes. Esto es, en una visión realista no se deben esperar situaciones

favorables siempre que el oponente pueda evitarlo, ni tampoco desfavorables, si el jugador puede rehuirlas.

El procedimiento es como sigue:

1. Si un nodo P es terminal, su valor es el de la función de evaluación para el jugador para el que se resuelve el árbol de búsqueda.
2. En otro caso, su valor será:
  - El máximo de los valores de sus hijos si es un nodo alfa.
  - El mínimo de los valores de sus hijos si es un nodo beta.
3. El algoritmo finaliza cuando se dispone de la puntuación del nodo inicial. La mejor jugada será la que devolvió el valor elegido por el nodo raíz.

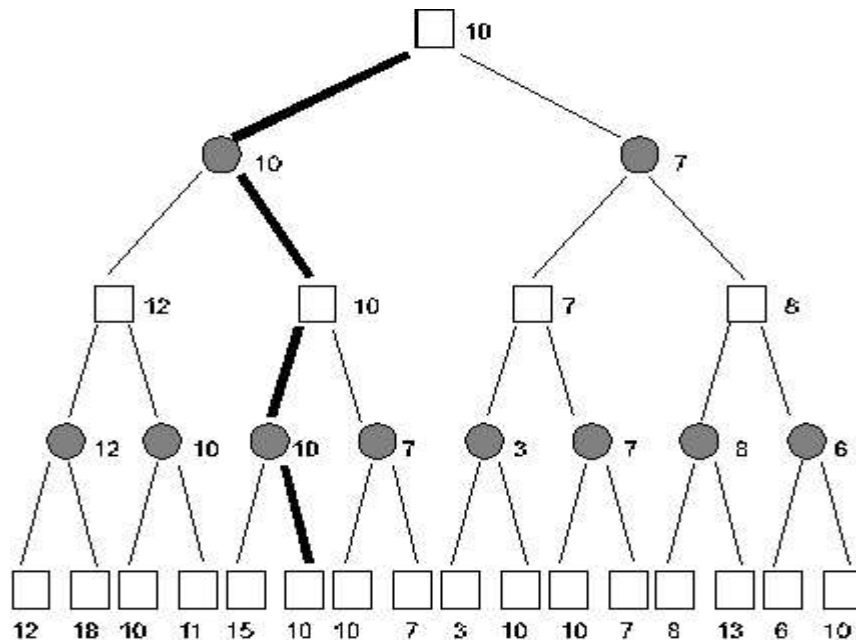


Figura 3.7: Se presenta un ejemplo de aplicación de este algoritmo. Junto a cada nodo no terminal se representa su valor Minimax.

Siendo la idea principal Maximizar las tiradas considerando que el oponente va a minimizar

Para decidir que jugada hacer, el árbol se divide por niveles:

- *max*: al primer jugador (nivel) y todas las posiciones (niveles) donde juega

- *min*: al oponente y todas las posiciones en donde juega

Las hojas se etiquetan con *gana*, *pierde*, *empata* desde el punto de vista de *max*

Si podemos etiquetar todas las hojas, podemos etiquetar todo el árbol

La respuesta generada por el juego, se toma y se convierte en una instrucción de control, la cual se envía al robot Lego MindStorms.

### 3.6 RESPUESTA EN TIEMPO REAL

La idea central para definir este proyecto como sistema con respuesta en Tiempo Real, es la noción de tiempo de respuesta o tiempo que tarda el sistema en generar una salida asociada a una entrada.

Debe tomarse en cuenta que en un sistema de Tiempo Real se debe distinguir de otros en donde el tiempo de respuesta es importante pero no crítico. Un sistema de Tiempo Real depende no sólo de los resultados de la computadora, si no también del tiempo que tarda en producirse dichos resultados.

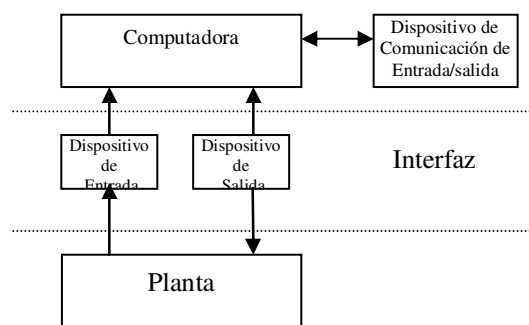


Fig. Esquema general de un sistema de control por computadora

En la figura se representa el esquema general de un sistema de control por computadora. Una característica común en los sistemas de tiempo real es que la computadora se encuentra conectado en un entorno dentro del cual está trabajando para un gran número de dispositivos de los que recibe y a los que da una amplia variedad de estímulos. Las entradas

y salidas y las señales de comunicación tienen una característica común. Éstas están conectadas a dispositivos que comunican el computador con procesos externos a él. Todos estos procesos externos operan en su propia escala de tiempo y es el computador quien se encarga de producir las salidas, de tal forma, que el instante de tiempo en que se producen, esté de acuerdo con las escalas de tiempos de los procesos externos.

Atendiendo a la forma de sincronizar los procesos externos con las acciones internas realizadas por el computador, los sistemas de tiempo real se pueden clasificar en:

**Sistemas basados en reloj.** La sincronización se define en términos de paso de tiempo.

**Sistemas basados en sensores.** La sincronización se define en términos de sucesos, por ejemplo el cierre de un interruptor.

**Sistemas interactivos.** La relación entre las acciones del computador y la evolución del sistema se define de forma más amplia que en los casos anteriores

En este proyecto se plantea utilizar la clasificación de Sistema interactivo, debido al conjunto de acciones que tiene que desarrollar la computadora con relación con sus entradas externas como la cámara y la lectura de sensores del Robot Lego MindStorms

## CONCLUSIÓN

En este capítulo se presentaron los medios para desarrollar un sistema de procesamiento de imágenes el cual tomará una decisión en base al resultado del algoritmo del tic-tac-toe, para manipular un robot manipulador.

Sin embargo los detalles sobre el desarrollo del proyecto se explican en el siguiente capítulo.