



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

“Control remoto mediante visión artificial”

TESIS

Que para obtener el título de:

Licenciado en Ciencias de la Computación

Presenta:

María Antonia Ruíz Díaz

Asesor: Dr. Luis Carlos Altamirano Robles

Coasesor: Lic. Jorge Eduardo Xalteno Altamirano

H. Puebla de Zaragoza, Febrero de 2009.

Agradecimientos

*Porque hay gente que no consigues olvidar jamás,
no importa el tiempo que eso dure...*

Gracias por todo abuelitos

Francisco Díaz Pérez †
Plácida Anzúres Vargas †



A mi mamá María Eugenia Díaz Anzures, por que no hay manera de agradecer tu amor, tus consejos, tus enseñanzas, por tu gran entereza y esfuerzo por educarme, y por las tantas cosas más, que son imposibles de describir en unas pocas líneas. Gracias mamá te quiero mucho.

A mi tía María del Rosario Díaz Anzures y a mi hermana María Guadalupe Rojas Díaz, gracias por su compañía, su cariño, por escucharme siempre, por animarme, y por ser mi apoyo en los buenos y malos momentos.

A mis papás Antonio Ruíz Domínguez y Marco Antonio Rojas Díaz, a mis tíos José Díaz Anzures, José Francisco Díaz Anzures †, Patricia Sánchez Mendoza y José Enrique Díaz Anzures, gracias por su apoyo y ejemplo.

A Jorge Eduardo Xalteno Altamirano por todas las cosas compartidas, por ser mi compañero, confidente y gran amigo, por tu cariño, por el apoyo incondicional que me has dado desde que te conocí y por ser uno de los motores que me impulsa a seguir adelante día a día.

A mis amigos: Pedro Tecuanhuehue Vera, Augusto Meléndez Teodoro, José Alfredo Salazar Romero, Felipe Serrano Martínez, Martha Angélica Solano Hernández, Karina Meneses Flores, Roberto Ángel Meléndez Armenta, Jorge Alejandro Ramírez Suárez, Fabricio Otoniel Pérez Pérez y Marco Arturo Ramírez Cadena, gracias por brindarme su amistad, y por los momentos y aventuras compartidos, porque con ustedes el trayecto siempre es más placentero.

A la Profa. Consuelo Vargas Romero gracias por su valiosa amistad, sus enseñanzas y por ser un ejemplo de lucha continua, y gracias igualmente a Enrique Aurelio Hernández Meza por apoyar a mi familia sin condiciones.

Y al Dr. Luis Carlos Altamirano Robles, gracias por su paciencia y por haberme asesorado para poder ver concluido este trabajo.

Resumen

En la actualidad existen inmensidad de aplicaciones de carácter tecnológico que van desde el entretenimiento hasta áreas como la telemedicina, la industria, pasando por la educación y el control remoto de dispositivos electrónicos de naturaleza variada, tales como electrodomésticos y robots.

Hoy en día surgen aplicaciones basadas en visión artificial que hasta hace poco se consideraban imposibles, y en un futuro no muy lejano surgirán aplicaciones avanzadas que podrán ser usadas en acciones de nuestra vida diaria.

En este trabajo se realiza el diseño y construcción de un sistema prototipo de control remoto para dispositivos electrónicos basado en visión artificial, y desarrollado en MATLAB así mismo se presenta una solución en hardware para dicho problema.

La organización de este trabajo de tesis es la siguiente:

En el capítulo I se presentan los conceptos preliminares necesarios para fundamentar la propuesta, así como para dar un mejor entendimiento, en el capítulo II se presentan algunos de los antecedentes de la visión artificial, y se describe a groso modo la propuesta de trabajo, en el capítulo III se hace el planteamiento del problema en forma más detallada y se presenta la solución en software del sistema, en el capítulo IV se muestran las conexiones utilizadas para la implementación en hardware, en el capítulo V se presentan las pruebas realizadas, y finalmente en el capítulo VI las conclusiones y el trabajo futuro.

Índice general

| | |
|--|-----|
| Agradecimientos | i |
| Resumen | iii |
| Índice general | iv |
| Índice de figuras | v |
| Índice de Tablas | vii |
| Introducción | 1 |
| Capítulo I. Conceptos preliminares | 4 |
| Capítulo II. Planteamiento del problema | 18 |
| Capítulo III. Solución en Software | 23 |
| Capítulo IV. Solución en Hardware | 34 |
| Capítulo V. Pruebas | 38 |
| Capítulo VI. Conclusiones y trabajo futuro | 52 |
| Apéndice I. El puerto paralelo | 54 |
| Referencias | 59 |

Índice de figuras

Capítulo I

| | |
|---|----|
| Figura 1.1: Características de un pixel. | 5 |
| Figura 1.2: Representación binaria de una imagen. | 7 |
| Figura 1.3: Mezcla aditiva de colores. | 8 |
| Figura 1.4: Cubo RGB. | 9 |
| Figura 1.5: Componentes del modelo de color HSI. | 11 |
| Figura 1.6: Cono de colores. | 12 |
| Figura 1.7: Cono de colores. | 13 |
| Figura 1.8: Ejemplo de dilatación. | 14 |
| Figura 1.9: Ejemplo de Erosión. | 15 |

Capítulo II

| | |
|---|----|
| Figura 2.1: Ejemplo de un control remoto universal. | 21 |
|---|----|

Capítulo III

| | |
|--|----|
| Figura 3.1: Interfaz del sistema. | 25 |
| Figura 3.2: Ejecutando vfm. | 26 |
| Figura 3.3: Imagen con pico hacia arriba. | 31 |
| Figura 3.4: Imagen con un pico hacia la derecha. Imagen con un pico hacia la derecha rotada 90°. | 32 |

Capítulo IV

| | |
|--|----|
| Figura 4.1: Ubicación en la placa del control remoto de las funciones del sistema. | 34 |
| Figura 4.2: Interior de los botones del control remoto. | 35 |
| Figura 4.3: Conexión. | 35 |

Capítulo V

| | |
|---|----|
| Figura 5.1: Regiones detectadas en la imagen 9 de la tabla 5.3 | 50 |
| Figura 5.2: Regiones de piel detectada en la imagen 14 de la tabla 5.2. Borde de la imagen. | 50 |

Apéndice

| | |
|---------------------------------------|----|
| Figura a1: Cable paralelo. | 54 |
| Figura a2. Pines del puerto paralelo. | 56 |

Índice de tablas

Capítulo II

| | |
|--|----|
| Tabla 2.1: Señales válidas reconocidas por el sistema. | 19 |
|--|----|

Capítulo V

| | |
|--|----|
| Tabla 5.1 Procedimiento realizado con buena iluminación. | 39 |
|--|----|

| | |
|---|----|
| Tabla 5.2 Procedimiento realizado con exceso iluminación. | 40 |
|---|----|

| | |
|---|----|
| Tabla 5.3 Secuencia de imágenes tomadas con la cámara web, con poca iluminación y fondo controlado. | 40 |
|---|----|

| | |
|---|----|
| Tabla 5.4 Secuencia de imágenes tomadas con una cámara digital, con buena iluminación y fondo controlado. | 43 |
|---|----|

| | |
|--|----|
| Tabla 5.5 Secuencia de imágenes tomadas con una cámara digital, con buena iluminación y fondo no controlado. | 46 |
|--|----|

| | |
|--|----|
| Tabla 5.6 Resultados obtenidos de las diferentes secuencias de imágenes. | 49 |
|--|----|

Apéndice

| | |
|---|----|
| Tabla a1. Descripción de los pines del puerto paralelo. | 56 |
|---|----|

Introducción

Cuando surgieron las primeras computadoras, hace ya más de 50 años, algunos científicos se preguntaron si estas máquinas podrían llegar a realizar tareas que requirieran un comportamiento inteligente.

Entonces se empezaron a desarrollar programas que trataran de emular la inteligencia humana, es así como nace una nueva disciplina conocida como Inteligencia Artificial.

En los primeros años se notaron avances aparentemente muy rápidos, con programas de computadora que juegan ajedrez, programas con la habilidad de responder a ciertos diagnósticos y consultas de los consumidores.

Surge la Visión Artificial también conocida como Visión por Computador (del inglés Computer Vision) como una rama de la Inteligencia Artificial, cuyo objetivo es “entender” o “comprender” una escena o ciertas características de una imagen.

Sin embargo, debido a que las leyes que rigen al mundo visual no son fáciles de describir, la tarea de lograr que la computadora realice el proceso de reconocimiento se vuelve extremadamente difícil. Por supuesto, las imágenes pueden ser pensadas como mapas definidos sobre dominios bidimensionales, y se comprende el proceso de formación de la imagen de la interacción física entre superficies, volúmenes y luz, pero el “vocabulario” usado por el cerebro para analizar esta entrada visual, esto es, las representaciones visuales, permanecen aún extremadamente inaccesibles.

Esto hace la tarea de construcción de sistemas de visión artificial diferente de otras tareas de Inteligencia Artificial, tales como el procesamiento de lenguaje natural. Esto es también un lado fascinante de la visión computacional, debido a que somos libres para proponer la representación visual más apropiada para una tarea dada.

No es posible reconocer completamente a un objeto solamente a través de sus propiedades geométricas, pues al hacerlo se pierde una cantidad muy significativa de información acerca de él, información que puede ser crucial cuando se le quiera reconocer. Aunado a esto, los enfoques geométricos usualmente tratan de representar al objeto de manera tridimensional, lo que requiere una gran capacidad de procesamiento del sistema de reconocimiento.

En la década de los 90's se ha propuesto el uso de modelos basados en la apariencia para reconocer a los objetos, de tal manera que es posible mediante el uso de ellos preservar no solamente las características geométricas de los objetos, sino también otro tipo de información tal como la textura, el color y otras características contenidas dentro del contorno.

No se ha podido desarrollar un sistema que tenga la capacidad de reconocer muchos objetos con propiedades muy diferentes como lo hacemos los humanos. Se han desarrollado muchas aplicaciones, pero la tarea de analizar y procesar es compleja, ya que se tienen que tomar en cuenta muchas características, tales como, el color, tamaño, textura, para poder distinguir los objetos, ya sea porque varían en su apariencia visual, porque se ven desde diferentes puntos de vista, o porque son muy complejos y a su vez están formados por otros objetos (como una cara humana), por ello que los sistemas de visión, se enfocan a reconocer un pequeño grupo de objetos.

La visión artificial ya no ofrece solamente soluciones a campos como la robótica o el control de calidad sino que continuamente surgen nuevas aplicaciones que hasta hace poco se consideraban imposibles, se han creado sistemas que mediante visión artificial reconocen partes humanas para el reconocimiento de las mismas y autenticación de personas, las huellas dactilares, las retinas, el iris, los patrones faciales, las venas de la mano o la geometría de la palma de la mano, representan ejemplos de características físicas (estáticas), mientras que entre los ejemplos de características del comportamiento se incluye la firma, el paso y el tecleo (dinámicas). La voz y las características de las manos se consideran una mezcla de características físicas y del comportamiento, pero todos los rasgos biométricos comparten aspectos físicos y del comportamiento.

La visión humana juega un papel muy importante en la vida del hombre, muchos científicos hoy en día tratan de descubrir la naturaleza de la visión humana investigando cómo se perciben y procesan los objetos. La visión artificial incluye muchos campos de desarrollo, pero en general es un área que usa técnicas para estimar los detalles de imágenes.

Esto originó entonces la motivación para realizar este trabajo, mostrar que por medio de la visión artificial se pueden generar soluciones útiles para la vida diaria, y concretamente, en el control de dispositivos sin intervención directa del humano y esto en un futuro probablemente permita construir un dispositivo comercial que se use en cualquier hogar del mundo.

Esta tesis presenta cómo se ha abordado el problema de controlar un televisor, sin el uso de un control remoto, sino por medio de la visión artificial, que mediante la identificación de señales reconocidas por el sistema en secuencias de imágenes, controle de forma remota dicho televisor, sin tener contacto directo con el usuario, ni con ningún dispositivo de control remoto.

En primer lugar el sistema tiene que identificar el momento en que en la secuencia de imágenes aparezca “una mano” y enseguida interpretar la señal hecha, para finalmente llevar a cabo la tarea ordenada.

El sistema presenta múltiples limitantes pero su desarrollo actual da la pauta para futuros trabajos.

Capítulo I

Conceptos preliminares

Procesamiento de imágenes

El procesamiento de imágenes puede ser visto como una transformación de una imagen en otra imagen, es decir, a partir de una imagen se obtiene otra imagen modificada.

Desde el punto de vista de los sistemas de visión artificial el único propósito del procesamiento de imágenes es hacer el análisis posterior más simple y más fiable. Por tanto el procesamiento de imágenes debe facilitar la extracción de información para un posterior análisis [1].

Pixel

Pixel es la abreviatura de la expresión inglesa Picture Element (Elemento de imagen), y es la unidad más pequeña que se encuentra en las imágenes.

El pixel es la unidad mínima en que se divide la retícula de la pantalla del monitor y cada uno de ellos tiene diferente color.

Su tono de color se consigue combinando los tres colores primarios (rojo, verde y azul) en distintas proporciones. (En realidad, el azul y el verde son pigmentos secundarios, pero son colores primarios de la luz, junto con el rojo) [3].

Un pixel tiene tres características distinguibles:

- Forma cuadrada.
- Posición relativa al resto de pixeles de un mapa de bits.

- Profundidad de color (capacidad para almacenar color), que se expresa en bits.

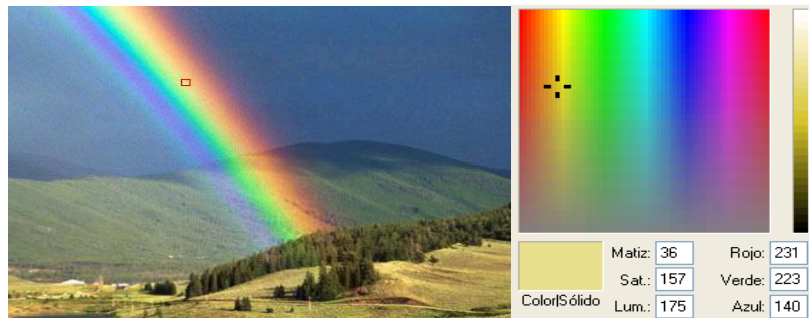


Figura 1.1 Características de un pixel.

Mapa de bits

También llamada imagen rasterizada o bitmap, es una estructura o fichero de datos que representan generalmente una rejilla rectangular de pixeles o puntos de color, denominada raster, en un monitor de computadora, papel u otro dispositivo de representación.

Los gráficos rasterizados se distinguen de los gráficos vectoriales en que estos últimos representan una imagen a través del uso de objetos geométricos, como curvas y polígonos, no del simple almacenamiento del color de cada pixel.

Imagen digital

El concepto de imagen está asociado a una función bidimensional $f(x,y)$, cuya amplitud o valor será el grado de iluminación (intensidad de la luz) en el espacio de coordenadas (x,y) de la imagen para cada punto. El valor de esta función depende de la cantidad de luz que incide sobre la escena vista, así como de la parte que sea reflejada por los objetos que componen dicha escena. Estos componentes son llamados *iluminación* y *reflexión*, siendo descritos por $i(x,y)$ y $r(x,y)$ respectivamente. El producto de ambas funciones proporciona la función $f(x,y)$.

$$f(x,y) = i(x,y) r(x,y) \quad (1)$$

La naturaleza de la iluminación viene determinada por la fuente de luz, mientras que la reflexión depende de las características del objeto en la escena. [1]

Una imagen digital en escala de grises es un matriz de $M \times N$ elementos numéricos cuyos valores posibles van del 0 (negro) al 255 (blanco), siendo este número la intensidad luminosa en el determinado punto o pixel (picture element), por convención el origen de la imagen se encuentra en el extremo izquierdo superior.

Una imagen digital a colores esta formada por 3 matrices de $M \times N$ elementos numéricos cuyos valores posibles van del 0 (negro) al 255 (blanco), siendo este número la intensidad luminosa en cada una de las bandas espectrales del RGB (Rojo, Verde, Azul), de cada punto o pixel (picture element), a diferencia de las imágenes en escala de grises, las imágenes a color requieren de la combinación de las 3 bandas de color, para representar el color de un pixel.

Por ejemplo, un determinado punto blanco de una imagen en escala de grises se describiría: $P(x, y) = 255$, sin embargo en una imagen a colores para describir el color del mismo punto se realizaría así : $P(x, y) = (255,255,255)$, esto debido a que el $(0,0,0)$ corresponde al negro absoluto y el $(255,255,255)$ al blanco absoluto.

Imagen Binaria

Una imagen binaria es una función de dos variables discretas $a[m,n]$ que puede tomar dos valores, '0' o '1', dependiendo del nivel de gris de la imagen (una imagen binaria tiene dos niveles: blanco y negro) [4].

Por ejemplo, el conjunto de todos los píxeles negros en una imagen binaria constituye una descripción completa de la misma.

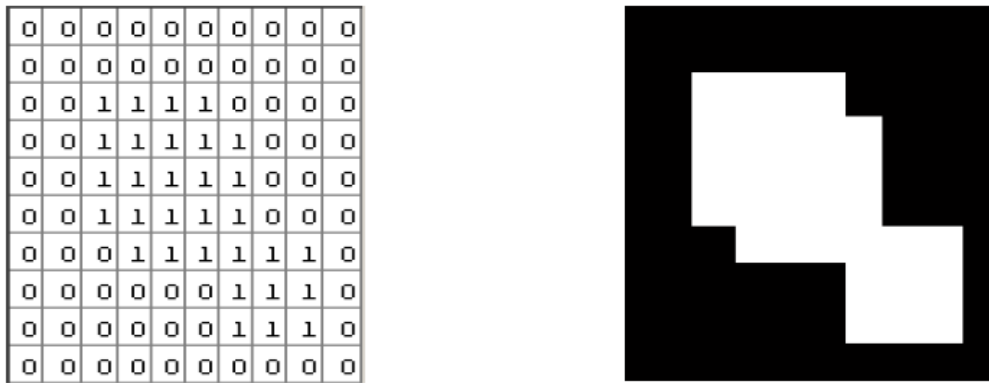


Figura 1.2. Representación binaria de una imagen.

Los conjuntos en cuestión pertenecen al espacio Z , donde cada elemento del conjunto es una tupla (vector 2D) cuyas coordenadas son las coordenadas $[m,n]$ de un píxel blanco (o negro, según la convención) de la imagen.

Segmentación

La segmentación es el proceso por el cual se extrae de la imagen cierta información subyacente para su posterior uso. La segmentación está basada en dos principios fundamentales: discontinuidad y similitud. Cabe pues enfocar la segmentación orientada a bordes (discontinuidad) y orientada a regiones (similitud), así una región es en líneas generales un área de la imagen en la que sus píxeles poseen propiedades similares (de intensidad, color, etc.), mientras que un borde es una línea que separa dos regiones, por tanto de diferentes propiedades (discontinuidad) [1].

Modelo de color RGB

Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores luz primarios [5] [11].

RGB (del inglés Red, Green, Blue; "rojo, verde, azul") de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul.

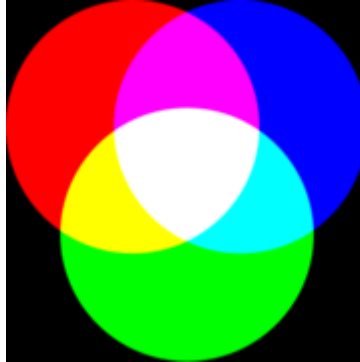


Figura 1.3. Mezcla aditiva de colores.

Para indicar con qué proporción mezclamos cada color, se asigna un valor a cada uno de los colores primarios, de manera, por ejemplo, que el valor 0 significa que no interviene en la mezcla, y a medida que ese valor aumenta, se entiende que aporta más intensidad a la mezcla. Aunque el intervalo de valores podría ser cualquiera (valores reales entre 0 y 1, valores enteros entre 0 y 37, etc.), es frecuente que cada color primario se codifique con un byte (8 bits). Así, de una manera estándar, la intensidad de cada una de las componentes se mide según una escala que va del 0 al 255.

Por lo tanto, el rojo se obtiene con $(255,0,0)$, el verde con $(0,255,0)$ y el azul con $(0,0,255)$, obteniendo un color resultante monocromático. La ausencia de color -lo que nosotros conocemos como color negro- se obtiene cuando las tres componentes son 0, $(0,0,0)$. La combinación de dos colores a nivel 255 con un tercero en nivel 0 da lugar a tres colores intermedios: el amarillo $(255,255,0)$, el cian $(0,255,255)$ y el magenta $(255,0,255)$. Y el blanco se forma con los tres colores primarios a su máximo nivel $(255,255,255)$.

El conjunto de todos los colores se puede representar en forma de cubo; cada color es un punto de la superficie o del interior de éste. La escala de grises es la diagonal que une al blanco y al negro.

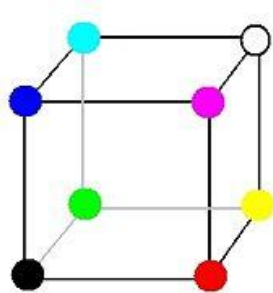
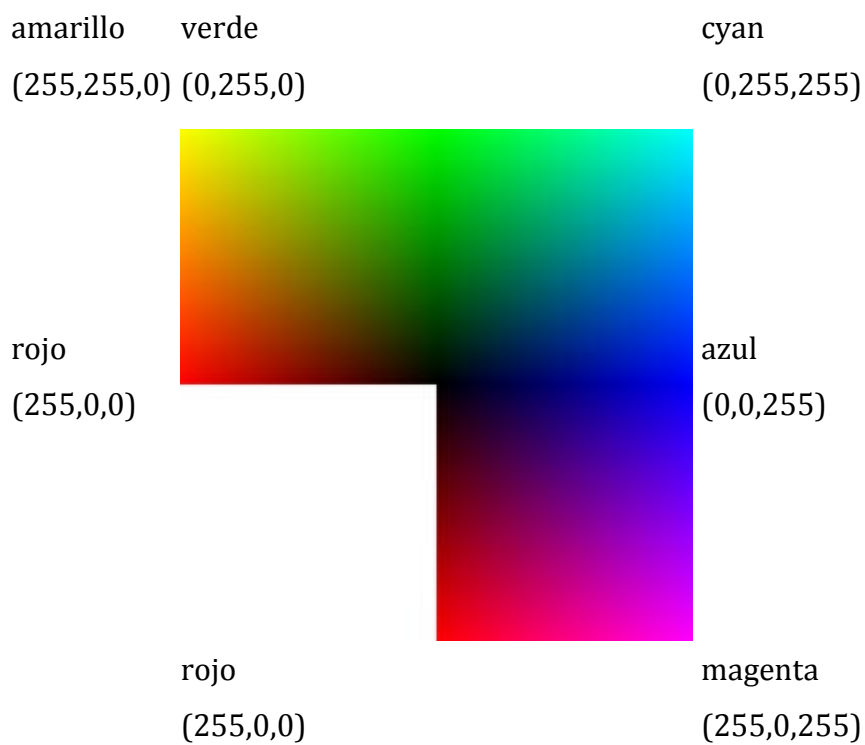


Figura 1.4. Cubo RGB.

Los colores que presentan la máxima saturación y la máxima luminosidad a la vez, son los que reúnen dos requisitos: al menos uno de los coeficientes es 255 y al menos uno de los coeficientes es 0. De esto se deduce que los colores más saturados y más luminosos siguen la siguiente secuencia:



Modelo de color HSI

HSI (del inglés Hue, Saturation e Intensity) con los que los seres humanos especificamos el color percibido, es decir, el matiz o color principal (H), la saturación o pureza de color (S), la intensidad o luminancia (I).

El matiz es el atributo del color que describe su pureza (amarillo puro, rojo, naranja), mientras que la saturación proporciona una medida del grado en que el color puro es diluido por la luz blanca.

El modelo de color HSI debe su utilidad a dos hechos fundamentales. Primero la componente de intensidad (I), se puede separar de la información del color de la imagen, Segundo las componentes de matiz (H) y saturación (S) están íntimamente ligadas con el modo en que los humanos percibimos el color. Estas características hacen del modelo HSI una herramienta ideal para desarrollar algoritmos de procesamiento de imágenes basados en alguna de las sensaciones de color del sistema visual humano. [2]

El color se describe con ayuda de tres magnitudes: longitud de onda predominante, factor de pureza y luminancia [6].

Longitud de onda dominante (λ_d): Es la longitud de onda de la radiación monocromática o color puro más parecida al color considerado (nm). Está asociado con el tono (hue) del color.

Factor de pureza (p): Coeficiente numérico sin unidades y entre 0 y 1 (1 = color puro), que expresa el grado de disolución del color puro (λ_d) en blanco. Está asociado con el concepto de saturación (saturation) del color.

Luminancia (L): Es la medida fotométrica de la intensidad de la radiación luminosa. Está asociada con el brillo (brightness), intensidad (intensity) o nivel de gris (value) del color:

$$I = \frac{R + G + B}{3}$$

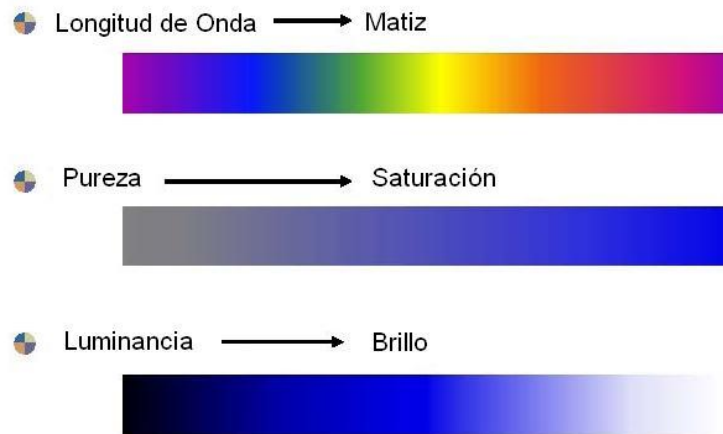


Figura 1.5. Componentes del modelo de color HSI.

Debido a que en este modelo se encuentra mayor facilidad para detectar cambios en sus componentes, su uso está muy ligado a la creación de algoritmos de procesamiento de imágenes que están basadas en propiedades del sistema de visión humano.

En ocasiones este modelo también es conocido como HSV (matiz, saturación, valor), HSB (matiz, saturación, brillo) o HSL (matiz, saturación, luminancia).

Los parámetros de luminancia (L), longitud de onda dominante (λ_d) y factor de pureza (p) pueden representarse mediante un cono.

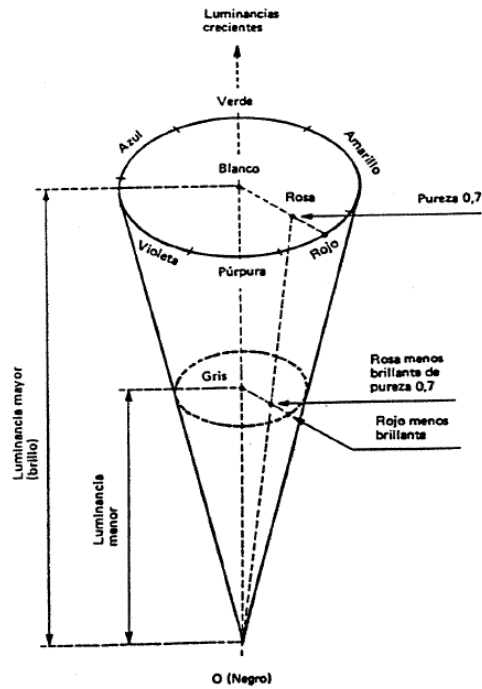


Figura 1.6. Cono de colores.

La longitud de onda corresponde con un punto en la circunferencia de la base.

El radio de la circunferencia indica el factor de pureza:

Centro circunferencia ($p = 0$): color blanco

Extremo circunferencia ($p = 1$): color puro

La altura del cono representa la luminancia (L):

Negro: vértice del cono

Blanco: centro de la base del cono

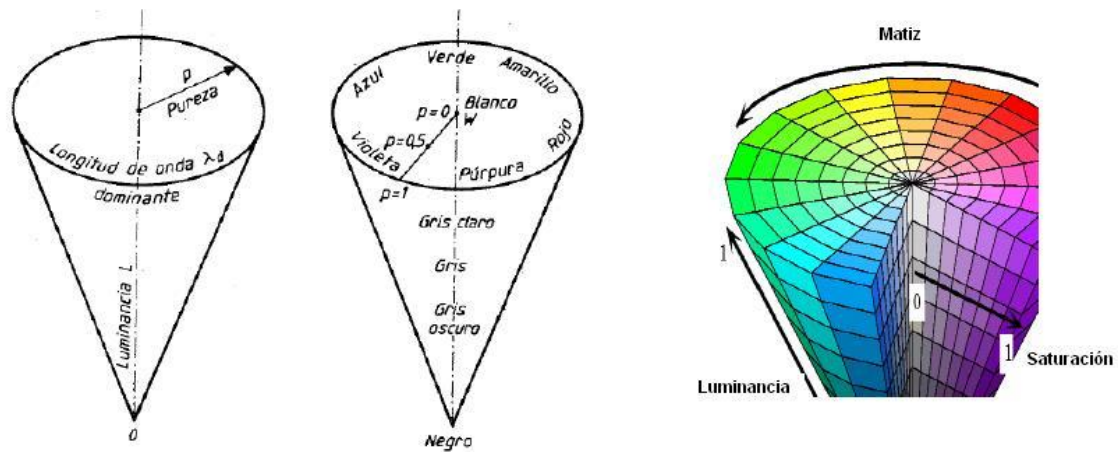


Figura 1.7. Cono de colores.

Algunos de los ejemplos de la utilidad de este modelo varían desde el diseño de sistemas para verificar el grado de madurez de las frutas o la inspección del acabado de color de determinados productos. La idea consiste en utilizar las propiedades del color del mismo modo que lo haría una persona encargada de esta misión.

Morfología

Se entiende por morfología el estudio de la forma y la estructura. Aplicando esto al tratamiento digital, sería el estudio de las formas y texturas de las imágenes, es decir, la modificación de la forma o estructura de los puntos que forman la imagen.

La morfología matemática es una herramienta muy utilizada en el procesamiento de imágenes.

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

Operadores morfológicos básicos:

También llamados operadores de rango (Rank operators), son dilatación, erosión y sus combinaciones: apertura y cierre. Existen más, basados en estos últimos, pero se salen de esta mención básica.

Dilatación y Erosión

Estas operaciones son fundamentales en el procesamiento morfológico. De hecho, la mayoría de los algoritmos morfológicos están basados en estas dos operaciones.

En general, la dilatación aumenta el tamaño de un objeto. La cantidad y la forma en que aumenta el tamaño dependen de la elección del elemento estructural.

Una de las aplicaciones más simples de la dilatación es la unión de píxeles relacionados.

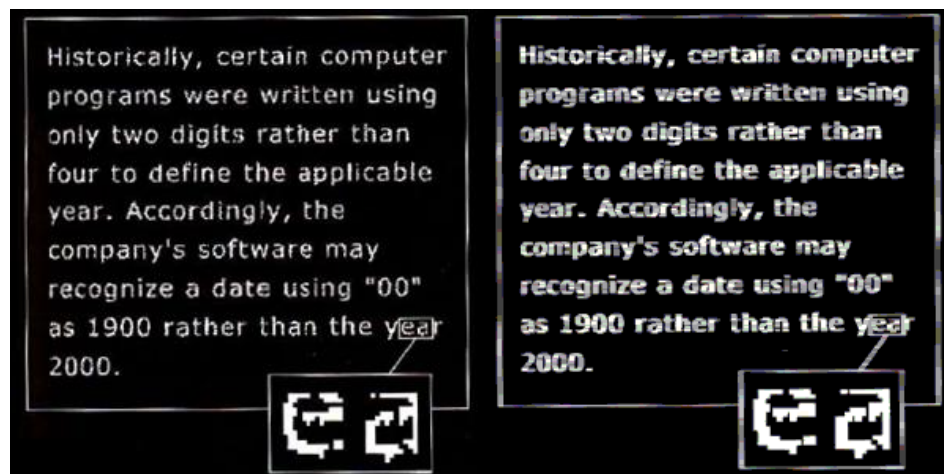


Figura 1.8. Ejemplo de dilatación.

La erosión disminuye el tamaño de los objetos. Como pasa en la dilatación, la cantidad y la forma en que se produce esta disminución depende del elemento estructural elegido.

Uno de los usos más simples de la erosión es para la eliminación de detalles irrelevantes.



Figura 1.9. Ejemplo de Erosión.

Apertura y cierre

Son las combinaciones de erosión/dilatación y dilatación/erosión respectivamente. El objetivo de la apertura y del cierre es simplificar la imagen, es decir, suavizan la señal.

La apertura (opening) se definirá como una combinación de erosiones y dilataciones en las que lo primero es una erosión y la última es una dilatación siempre con el mismo elemento estructural.

Se denomina apertura ya que al empezar por una erosión se tiende a romper las piezas en sus partes constitutivas.

La apertura sirve para eliminar pequeños detalles claros, pues lo que hace es erosionar para eliminar pequeños detalles oscuros, y más tarde dilatar para recuperar, ya sin esos detalles, el nivel de gris del principio.

Después de una erosión no se puede recuperar la imagen original con una dilatación ya que se habrán perdido pequeños detalles. Sin embargo esto tiene una gran utilidad ya que la información que se mantiene corresponderá a las partes más importantes de los objetos y además se habrá eliminado pequeños elementos debidos al ruido.

El cierre (closing) es la operación dual de la apertura. La apertura suaviza los contornos del objeto, rompe enlaces delgados y elimina pequeñas protusiones. El cierre también suaviza los contornos pero rellena fisuras y pequeños agujeros

La aplicación de la dilatación al principio, elimina los detalles de poca importancia de un nivel más claro, y la aplicación posterior de la erosión restablece el nivel del principio de gris general [1].

MATLAB

MATLAB es la abreviatura de MATrix LABoratory (laboratorio de matrices). Se trata de un software matemático muy versátil que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Es un entorno de computación y desarrollo de aplicaciones totalmente integrado, orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos.

MATLAB dispone de un amplio abanico de programas de apoyo especializados denominados Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa principal.

Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'toolbox' de procesamiento de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neurales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc. es un entorno de cálculo técnico, que se ha convertido en estándar de la industria, con capacidades no superadas en computación y visualización numérica.

En el presente trabajo se hace uso del toolbox de procesamiento de imágenes, el cual se describe brevemente a continuación.

Image Processing Toolbox

El Image Processing Toolbox ofrece un amplio conjunto de herramientas que permiten fácilmente manipular, procesar y analizar imágenes. Permite restaurar imágenes deterioradas o con ruido, mejorar imágenes para hacerlas más inteligibles, extraer características, analizar formas y texturas y registrar pares de imágenes. Debido a la orientación basada en matrices, permite efectuar de forma compacta operaciones matemáticas esto facilita efectuar el procesamiento de imágenes y operaciones de análisis, tales como FFTs, filtrado 2D, morfología binaria, manipulación geométrica, conversión de espacios de colores, compresión, análisis de componentes conectados y más. ¹

¹ Para más información puede acceder al siguiente link: <http://www.mathworks.com/products/image/?BB=1>

Capítulo II

Planteamiento del Problema

Este proyecto de tesis plantea el diseño y construcción de un prototipo de control remoto, por medio de la visión artificial, específicamente en este trabajo se pretende controlar un televisor.

Objetivo general:

- Diseñar el sistema según los requerimientos y restricciones apropiados.

Objetivos específicos:

1. Lograr un tratamiento eficiente de las imágenes tomadas por la cámara, para que el análisis posterior sea más rápido y eficaz
2. Realizar la mejor interpretación de la señalización contenida en las imágenes.
3. Controlar un televisor mediante un dispositivo electrónico de bajo costo que opera sin intervención directa del usuario.

Descripción del problema

Una cámara web conectada a una computadora, estará tomando secuencias de dos imágenes cada 2 segundos, una vez tomadas estas imágenes, se procede a hacer un tratamiento de ellas, se pretende eliminar la mayor cantidad de ruido posible y tratar de detectar si entre las dos imágenes capturadas se detecta movimiento.

Luego de que el ruido en las imágenes es eliminado y verificando que no hay movimiento entre ellas, se toma una de estas dos imágenes y se realiza la conversión del espacio de color RGB al espacio de color HSI, de lo que resulta otra imagen y procede a identificar si en esta última existen regiones de piel, enseguida se crea una nueva imagen con la mayor región de piel detectada y finalmente se

procede a verificar si existe alguna señal válida para el sistema, si es así, la computadora envía por medio del puerto paralelo la señal reconocida al dispositivo de hardware y finalmente el circuito envía la señal al televisor, haciendo que éste ejecute la acción deseada.

La cámara web estará tomando secuencias de dos imágenes, hasta que el usuario lo desee.

Las señales reconocidas por el sistema son las siguientes:

| | |
|---|---|
|  <p>Decremento de Volumen</p> |  <p>Aumento de Volumen</p> |
|  <p>Decremento de Canal</p> |  <p>Aumento de Canal</p> |

Tabla 2.1. Señales válidas reconocidas por el sistema.

El Control Remoto

Un control remoto es un dispositivo electrónico usado para realizar una operación remota sobre una máquina.

El término se emplea generalmente para referirse al mando a distancia para el televisor u otro tipo de aparato electrónico casero como los estéreos, DVD's, computadoras y para encender y apagar un interruptor, la alarma o abrir la puerta del garaje.

Estos aparatos son normalmente pequeños objetos, fácilmente manipulables con una mano, poseen una matriz de botones los cuáles se ajustan a las diferentes funciones del aparato controlado, como por ejemplo el canal del televisor, el número de canción o el volumen.

La mayoría de estos controles remotos se comunican con sus respectivos aparatos mediante señales de infrarrojo (IR) y sólo unos pocos utilizan señales de radio.

Su fuente de energía suele ser pequeñas pilas de tipo AA, AAA o de botón [7].

También existen los controles universales también llamados mandos universales los cuales son dispositivos electrónicos remotos que muchas veces son la única solución para controlar modelos de televisores muy antiguos o de poca comercialización.

Normalmente son utilizados cuando el control original sufre algún daño o se pierde, algunas ocasiones pierden algunas funciones respecto al control original, pero son útiles para funciones básicas.

Hay distintos formatos, desde los simplificados a los más sofisticados para varios aparatos.

Todos tienen en común la necesidad de introducir un código para su funcionamiento, es importante guardar las instrucciones y los códigos ya que en caso de desprogramación quedan inservibles sin estos datos.

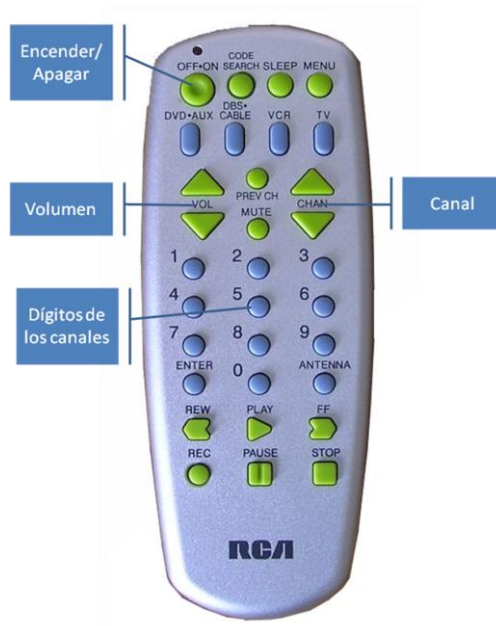


Figura 2.1. Ejemplo de un control remoto universal.

Delimitación del problema

El sistema sólo ejecuta cuatro funciones:

1. Decremento de volumen
2. Aumento de volumen
3. Decremento de canal
4. Aumento de canal

Las señales propuestas para el sistema tienen características que permiten una buena y no tan compleja detección, quizá la más significativa consiste en que una vez identificada una mano en la imagen, se procede a verificar si hay algún pico en ella (en donde el pico se puede ver como el dedo pulgar de la mano) y la orientación de tal pico.

Para la detección de señales se emplearán métodos para el tratamiento de las imágenes, tales como eliminación de ruido, segmentación, detección de regiones y extracción de características.

Es claro que las señales que el sistema podrá reconocer son pocas, comparadas con las funciones de un control remoto común, es posible agregar más señales, pero se necesitaría tomar en cuenta más características y emplear más métodos, pero esto queda fuera del alcance de este trabajo ya que eso requiere más tiempo para su implementación.

El sistema también se verá limitado en cuanto a condiciones de iluminación, para el buen funcionamiento de éste es necesario tener una buena iluminación, para poder obtener imágenes de mayor calidad que proporcionen los resultados deseados.

Capítulo III

Solución en Software

El sistema prototipo está desarrollado en MATLAB ya que, como se mencionó anteriormente MATLAB permite fácilmente manipular, procesar y analizar imágenes, y ya que posee el Image Processing Toolbox esto facilita aún más las operaciones como filtrado, morfología, conversión de espacios de colores, análisis de componentes conectados y más, MATLAB también permite adquirir imágenes desde dispositivos de captura, fácil y eficazmente, es por ello que se eligió para el desarrollo del sistema.

En cuanto al hardware con el que se desarrolló y probó el sistema es el siguiente:

- Computadora personal con procesador Celeron , 128M en RAM, 10G de DD, ejecutando Windows XP y MATLAB 7
- Internet USB Video Camera, Genius, Interface USB 1.1/1.0

El algoritmo propuesto para la solución del problema se describe a continuación:

1. Se obtiene una secuencia de dos imágenes, cada dos segundos, tomadas por una cámara web montada en la parte superior de un televisor.
2. Una vez obtenidas las dos imágenes se procede a eliminar el mayor ruido posible y enseguida se verifica que no se perciba movimiento.
3. Si no se percibe movimiento en las imágenes, se toma una sola imagen de dicha secuencia, la cual será la imagen a analizar y se continúa con el paso número 5.
4. Si se percibe movimiento en la secuencia de imágenes tomadas se regresa al paso número 1.
5. Una vez obtenida la imagen se hace la conversión del espacio de color RGB al espacio de color HSI, esto se hace para identificar si en la imagen es reconocida una mano.

6. Enseguida se aplica una umbralización para obtener una imagen binaria.
7. Se buscan los bordes.
8. Se buscan las componentes conectadas.
9. Se crea una nueva imagen con la región más grande.
10. Se procede a hacer la búsqueda de picos.
11. Si se encuentra una señal reconocida por el sistema, se manda a ejecutar la acción deseada, mediante el puerto paralelo de la computadora.
12. Si no se encuentra ninguna señal reconocida por el sistema, se regresa al paso número 1.

Para que el sistema inicie su ejecución el usuario debe presionar el botón **Iniciar**, y para finalizar la ejecución debe presionar el botón **Detener**, ambos botones se encuentran en la interfaz del sistema, mostrada en la figura 3.1.

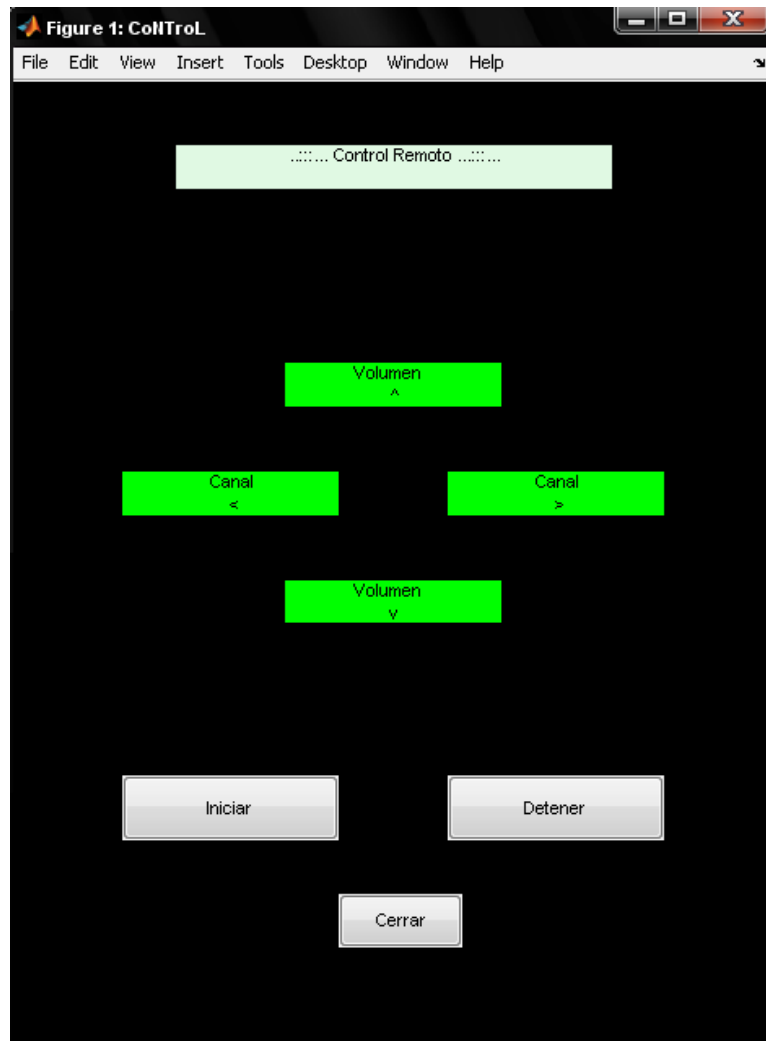


Figura 3.1. Interfaz del sistema.

A continuación se describe con mayor detalle el algoritmo a seguir y se presentan las funciones utilizadas:

Para obtener imágenes desde la cámara web se utilizó la herramienta vfm, vfm² es un conjunto de librerías dinámicas que permiten acceder directamente a los controladores del dispositivo registrados por Windows. De esta manera Web-cam's de bajo costo pueden ser utilizadas como dispositivos para captar una imagen y poderla utilizar en MATLAB y sus toolbox.

² La librería esta disponible para descarga en: <http://webscripts.softpedia.com/script/Scientific-Engineering-Ruby/Test-and-Measurement/matlab-vfm-36212.html>

La herramienta una vez instalada puede ser utilizada de manera muy sencilla, solo hay que poner en la línea de comandos:

```
>>vfm
```

La herramienta cuenta con tres menús que controlan la captura. En el menú **Driver** se permite seleccionar una de varias fuentes de captación de video instaladas, en el menú **Configure** se puede configurar el formato y características propias del driver previamente seleccionado, en el menú **Window** se permite controlar el flujo de la presentación de la imagen, en este caso se utiliza el modo **Preview** para que la imagen se esté mostrando en la ventana en todo momento:

```
vfm('preview');
```

Aparece la siguiente ventana:



Figura 3.2. Ejecutando vfm.

Las imágenes captadas utilizando la herramienta vfm son entregadas al espacio de trabajo en formato RGB.

La función que captura imágenes tiene el siguiente formato:

```
>>imagen_capturada=vfm('grab', n);
```

Donde `imagen_capturada` es la variable que contiene la imagen RGB y `n` es el número de 'frames' a capturar, de tal forma que si se quisiera capturar el 'frame' actual visto por la cámara se procedería a escribir:

```
>>imagen_RGB=vfm('grab',1);  
  
>>imshow(imagen_RGB)
```

Una vez obtenida la secuencia de imágenes (2 imágenes c/seg) se procede a eliminar el ruido que contengan, el objetivo de la eliminación de ruido es suavizar las variaciones bruscas de niveles de gris sin que ello difumine los bordes, base para la extracción de contornos y extracción de características de la imagen.

La función **imfilter** puede manejar imágenes multidimensionales y filtros multidimensionales, esto es que filtra cada plano de la imagen tridimensional individualmente con el mismo filtro de dos dimensiones.

La función aplicada en el algoritmo para la eliminación de ruido es la siguiente:

```
h = ones(5,5)/25;  
x = imfilter(rgb1,h);  
y = imfilter(rgb2,h);
```

Luego de aplicar el filtro se procede a verificar si en las imágenes se detecta movimiento, esto se consigue haciendo la resta de las dos imágenes y verificando que la diferencia entre ellas sea un valor muy pequeño o cero.

Ya que se obtiene una imagen libre de ruido y de movimiento, ésta se convierte del espacio de color RGB al espacio de color HSI para hacer la segmentación, para pasar del espacio de color RGB a HSI se procede de la siguiente forma:

```
r=F(:, :, 1);  
g=F(:, :, 2);  
b=F(:, :, 3);
```

```

th=acos((0.5*((r-g)+(r-b)))/(sqrt((r-g).^2+(r-b).*(g-
b))+eps));
H=th;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
S=1-3.*(min(min(r,g),b))./(r+g+b+eps));
I=(r+g+b)/3;

hsi=cat(3,H,S,I);

```

Donde tenemos como resultado la imagen hsi, y se continua con la segmentación.

La facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral (Santa Fe, Argentina) realizó una investigación titulada “Comparación de métodos de detección de piel en modelos de color YCbCr y HSI para reconocimiento de caras.” De la cual obtuvieron las siguientes reglas para la detección de piel [8]:

- 1) Si $(h < 0.1)$ pixel de piel
- 2) Si $(s > 0.3)$ pixel de piel

En este trabajo de tesis la regla del inciso 2) fue modificada a $s > 0.1$ ya que en las pruebas realizadas arrojó mejores resultados. Quedando como reglas para la detección de piel:

- 1) Si $(h < 0.1)$ pixel de piel
- 2) Si $(s > 0.1)$ pixel de piel

Si el pixel analizado no cumple con las reglas es descartado como perteneciente a una región de piel. De aquí, que se puede pasar de una imagen de color a una imagen binaria que represente las regiones de piel y no piel.

Una vez obtenida la imagen binaria se obtienen los bordes, esto se hace con la función **canny** la cual esta basada en el algoritmo de detección de bordes

propuesto por J. Canny en 1993, MATLAB tiene implementada esta función y posee el siguiente formato:

```
B_canny = edge(I, 'canny');
```

Se obtiene una nueva imagen formada por los bordes encontrados, ahora se procede a dilatar los bordes, esto se hace para obtener mejores resultados en las regiones, ya que la función **imdilate** engrosa la imagen.

La función tiene el siguiente formato:

```
se = strel('line',10,10);  
D = imdilate(cannyH,se);
```

Donde:

cannyH es la imagen que va a ser dilatada.

Teniendo la imagen dilatada se buscan las componentes conectadas, esto es para quedarnos con la región de piel más grande detectada e iniciar la búsqueda de picos.

```
CC=bwlabel(D,4); % Componentes conectadas  
RGB = label2rgb(CC, @jet, 'k'); % 'k' especifica el color de fondo  
(negro)
```

Las áreas de las regiones encontradas se almacenan en un vector:

```
stats = regionprops(CC, 'Area');  
allArea = [stats.Area]; % Crea un vector con las areas de las regiones  
NumReg=size(allArea) % Numero de regiones de la imagen
```

Se busca la región más grande:

```
areaM=allArea(1);  
  
for t=2:NumReg(2)  
    if areaM<allArea(t);  
        areaM=allArea(t);  
    end  
end
```

Y se crea una imagen con dicha región:

```
idx=find([stats.Area]==areaM); % Crea una imagen con la región mas
grande
F = ismember(CC,idx);
```

Ahora el último paso es realizar la búsqueda del pico, esto se hace utilizando la siguiente función:

```
while cp<30 && i<tam(1)
    j=1;
    while (band1 ==0 || band2==0) && j<tam(2)
        if F(i,j)==1 && band1==0
            c1=j;
            band1=1;
        end
        if F(i,k)==1 && band2==0
            c2=k;
            band2=1;
        end
        j=j+1;
        k=k-1;
    end

    k=tam(2);
    d=c2-c1;

    if bd==0 && (d>5 && d<60)
        d2=d;
        bd=1;
    end

    if band1==1 && band2==1 && (d>5 && d<60)
        if d>=d2-2 && d<=d2+5
            d2=d;
            cp=cp+1;
        end
    end
end
```

```
band1=0;  
band2=0;  
i=i+1;  
end
```

El orden de búsqueda del pico en la imagen es el siguiente: pico hacia arriba (canal en aumento), pico hacia la derecha (aumentar volumen), pico hacia abajo (canal en decremento), pico hacia la izquierda (disminuir volumen).

La función presentada para la detección del pico reconoce un pico hacia arriba, como se muestra en la imagen 3.3.

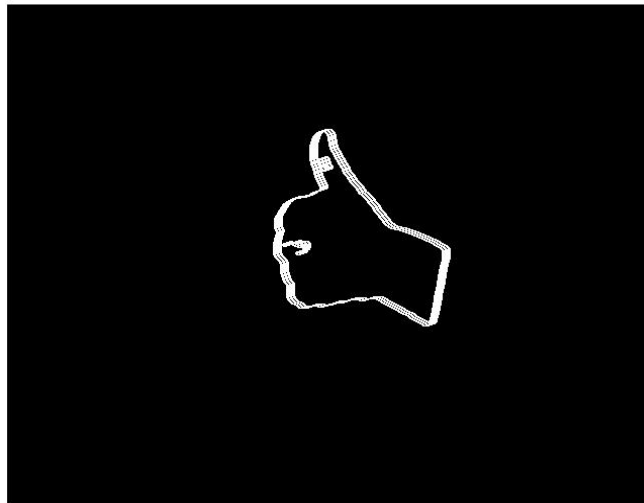


Figura 3.3 Imagen con un pico hacia arriba.

Esta función misma se utiliza para la detección del picos hacia la derecha, pico hacia abajo, y pico a la izquierda de la siguiente forma:

Si no se encuentra un pico hacia arriba, se procede a rotar la imagen 90° para la búsqueda de un pico hacia la derecha, 180° para la búsqueda de un pico hacia abajo y de 270° para la búsqueda de un pico hacia la izquierda.

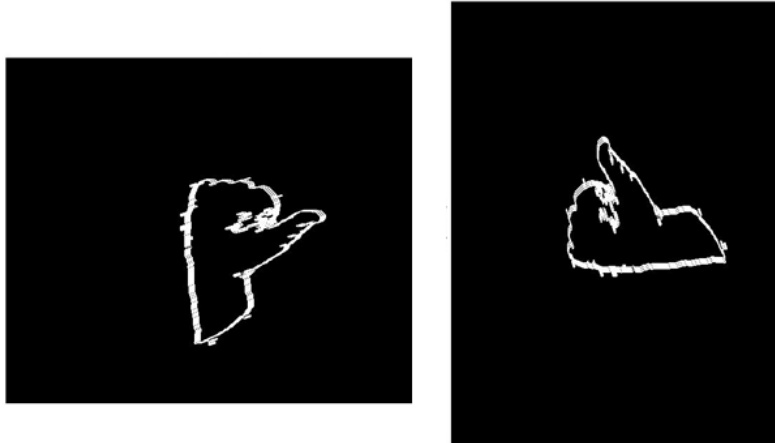


Figura 3.4. (De Izquierda a derecha) Imagen con un pico hacia la derecha. Imagen con un pico hacia la derecha rotada 90°.

Finalmente, si se encuentra una señal reconocida por el sistema, lo que resta es mandar la señal a través del puerto paralelo, esto se hace de la siguiente forma:

Para utilizar el puerto paralelo desde MATLAB fue necesario utilizar Ntport Library.³

El formato de la función para el envío de datos es el siguiente:

```
function [result] = matport('function', parameters)
```

Matport permite enviar o recibir datos desde algún puerto que se especifica con los parámetros.

Por ejemplo para recibir datos tenemos:

```
myvalue = matport('Inport', 97);
```

Lee del puerto 97 y el resultado lo almacena en la variable myvalue.

³ Más información acerca de Ntport en el apéndice I.

Para el envío de datos tenemos:

```
matport('outp', 888, 12);
```

En este caso se envía el valor 12 por el puerto 888 del puerto paralelo.

Para el envío de las señales se procedió de la siguiente manera:

1. Pico hacia arriba

```
matport('outp', 888, 1);
```

2. Pico hacia abajo

```
matport('outp', 888, 8);
```

3. Pico hacia la izquierda

```
matport('outp', 888, 4);
```

4. Pico hacia la derecha

```
matport('outp', 888, 2);
```

Capítulo IV

Solución en Hardware

El objetivo de esta sección es presentar una sencilla implementación en hardware, para la activación de los botones de un control remoto por medio de las líneas de salida que proporciona el cable paralelo.

Esta parte del trabajo es muy importante ya que una vez identificada alguna señal válida, la señal tiene que ser enviada desde la computadora al televisor mediante el puerto paralelo de alguna manera.

Como ya se ha mencionado en capítulos anteriores, para este trabajo solo se tomaron en cuenta cuatro funcionalidades, aumentar canal, disminuir canal, aumentar volumen y disminuir volumen, que son los cuatro botones del control remoto que se activaran. La ubicación de éstas funciones en la placa del control remoto, se muestra en la Ilustración 14.



Figura 4.1. Ubicación en la placa del control remoto de las funciones del sistema.

Para que alguna de estas funcionalidades se active, se necesita presionar el botón del control remoto correspondiente, los botones del control remoto contienen en el interior material conductor (Ilustración 15) que permiten cerrar el circuito, y mandar la señal al televisor.



Figura 4.2. Interior de los botones del control remoto.

El objetivo era lograr este mismo comportamiento pero sin presionar los botones del control remoto. Sino que una vez detectada por el sistema una señal valida, el control remoto enviara la señal al televisor.

Para lograr esto se realizó la siguiente conexión:

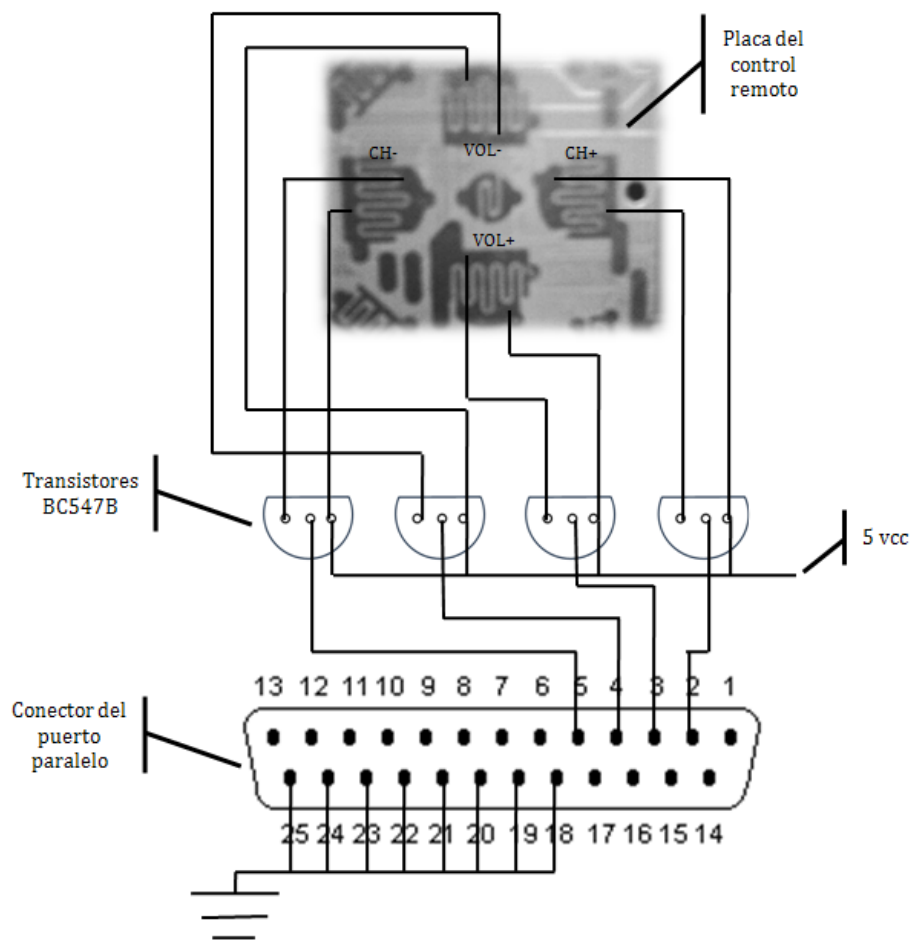


Figura 4.3. Conexión

Esta conexión activa los botones del control remoto automáticamente, una vez que alguna señal válida es detectada, se envía por el puerto paralelo un 1 (5vcc) al pin correspondiente a la señal detectada, esta señal llega a la base del transistor, el transistor cierra el circuito del control remoto y finalmente se envía la señal correspondiente al televisor.

A continuación se describe la conexión con más detalle.

Material utilizado:

- 1 Cable paralelo
- 1 Control remoto universal (con baterías)
- 4 Transistores BC547B
- Fuente externa de 5vcc
- 1 Protoboard
- Cable
- Soldadura

Descripción de la conexión:

De las ocho líneas de datos disponibles del cable paralelo, se utilizaron sólo cuatro, el pin 2, pin 3, pin 4 y el pin 5, una para cada funcionalidad.

Pin 2: Aumentar canal

Pin 3: Aumentar volumen

Pin 4: Disminuir volumen

Pin 5: Disminuir canal

Debido a que se necesitaba cerrar los circuitos de las funciones del control remoto, se utilizaron transistores, ya que éstos permiten controlar el paso de voltaje a través de sus terminales. El transistor que se utilizó es BC547B tipo NPN.

Cada terminal colector de los transistores se conecta a 5 vcc, y cada una de las líneas de datos se conecta a la terminal base de cada transistor.

Cuando el puerto paralelo envía 5vcc hacia la terminal base de un transistor, el voltaje fluye por el emisor y colector, cerrando el circuito del control remoto, y enviando la señal al televisor.

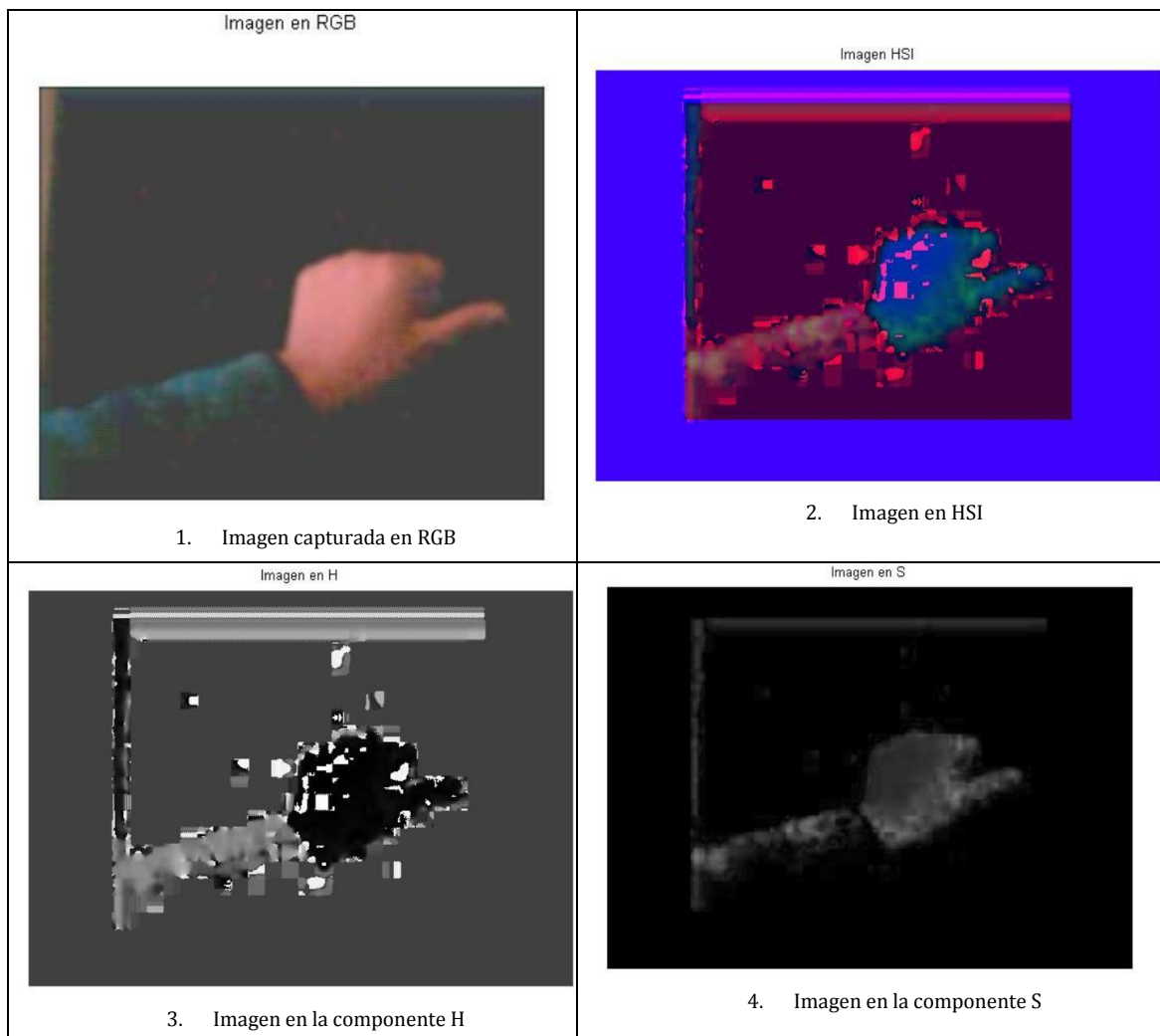
Como se puede observar la conexión presentada es muy sencilla, si utilizáramos las cuatro líneas de datos restantes del cable paralelo, podríamos hacer uso de otras cuatro funcionalidades del control remoto estableciendo otras cuatro señales válidas adicionales.

Capítulo V

Pruebas

El sistema se probó con diferentes fondos y con diferentes condiciones de iluminación en la habitación.

A continuación se muestra el procedimiento llevado a cabo, en uno de los mejores casos para el sistema, una imagen tomada con la cámara web, con un fondo controlado (negro), la distancia entre la señal de la mano y la cámara de 50cm y con la adecuada intensidad de luz.



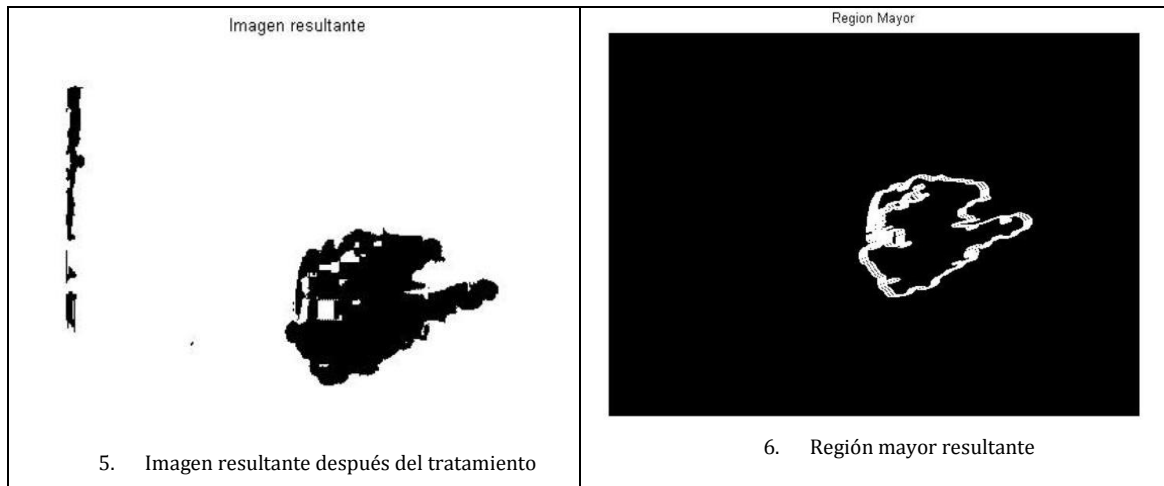
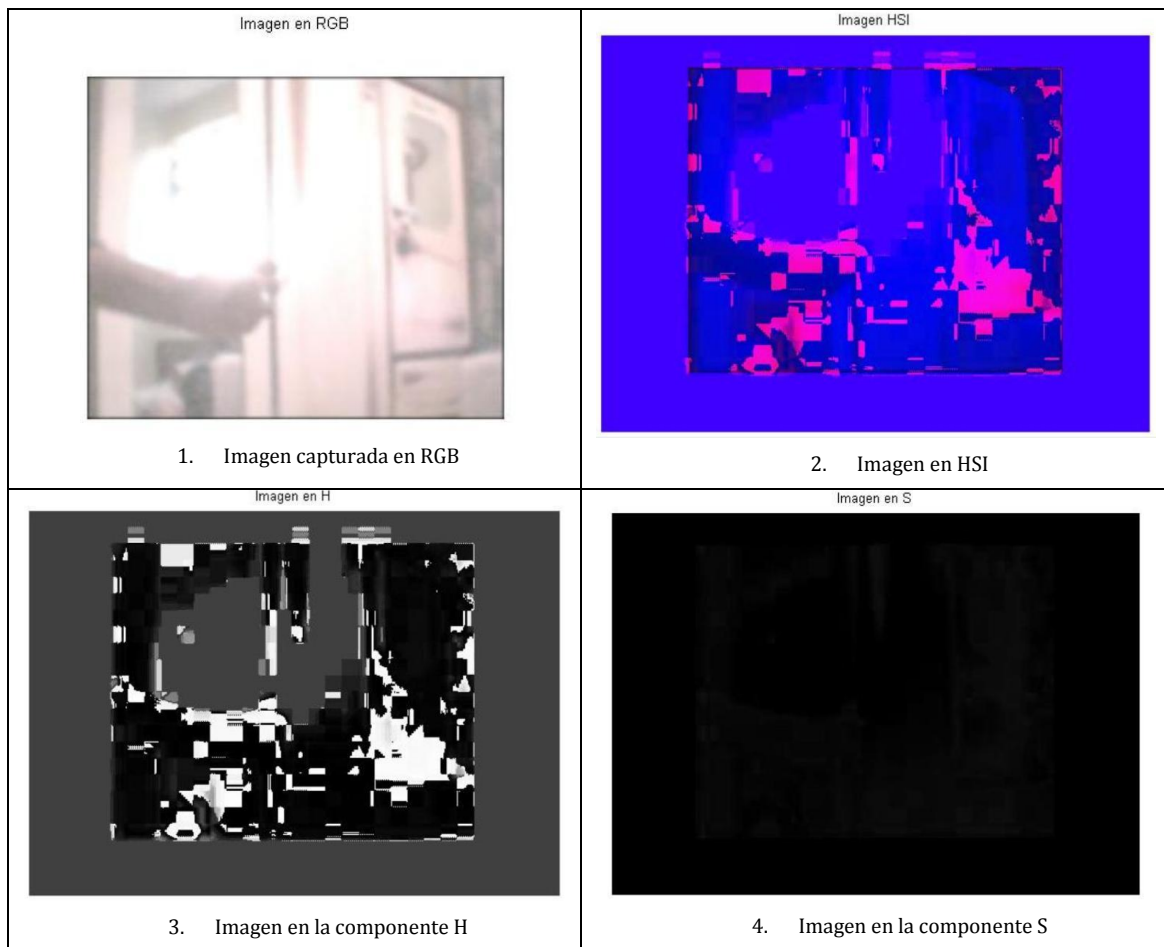


Tabla 5.1 Procedimiento realizado con buena iluminación.

Como se ha mencionado, el buen funcionamiento del sistema, depende mucho del ambiente, a continuación se presenta lo que el sistema hace, cuando hay mucha luz entre la cámara web y la señal hecha con la mano.



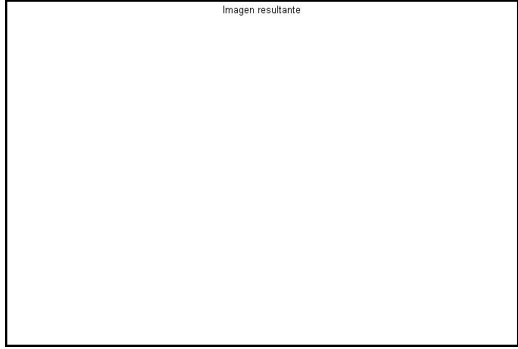
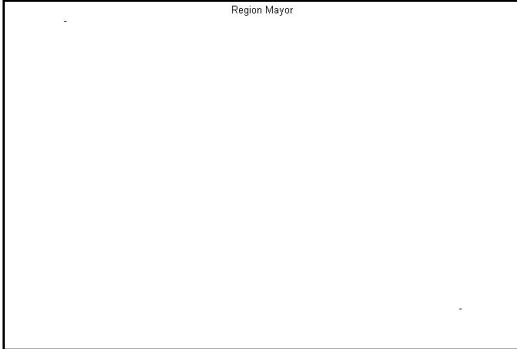

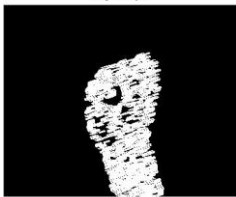

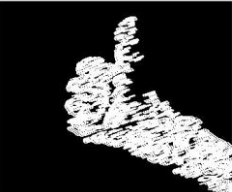


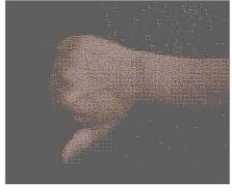
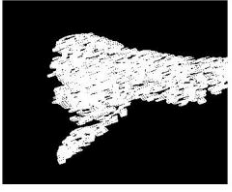

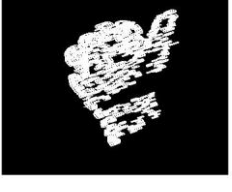



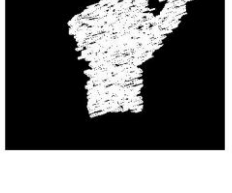
| | |
|---|--|
|  <p>5. Imagen resultante después del tratamiento</p> |  <p>6. Región mayor resultante</p> |
|---|--|












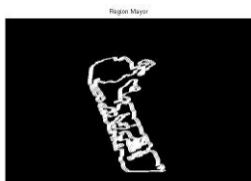
Tabla 5.2. Procedimiento realizado con exceso de iluminación.

En la imagen 1 se presenta la imagen capturada en RGB, la imagen 2 muestra la imagen en HSI, las imágenes 2 y 3 muestran la imagen en las componentes H y S respectivamente, como se puede observar en éstas últimas dos imágenes es muy difícil identificar la zona de la imagen donde se encuentra piel, ya que debido a la luz los colores reales se distorsionan y los colores de fondo se asemejan la mano que se encuentra en la imagen, en consecuencia la imagen 5 muestra una imagen totalmente blanca, ya que en ningún punto de las imágenes se cumple la regla de detección de piel, finalmente la imagen 6 también es totalmente blanca ya que no existen regiones en la imagen 5.

A continuación presenta una secuencia de imágenes tomadas con la cámara web y con poca iluminación en la habitación.

| | Imagen captada | Imagen final | Respuesta del sistema | Respuesta real |
|----------|---|---|-----------------------|---------------------|
| 1 |  |  | No se detecto señal | No se detecto señal |

| | | | | |
|---|--|---|---------------------|---------------------|
| 2 | <p>Imagen en RGB</p>  | <p>Region Mayor</p>  | Canal en aumento | Canal en aumento |
| 3 | <p>Imagen en RGB</p>  | <p>Region Mayor</p>  | No se detecto señal | No se detecto señal |
| 4 | <p>Imagen en RGB</p>  | <p>Region Mayor</p>  | Canal en decremento | Canal en decremento |
| 5 | <p>Imagen en RGB</p>  | <p>Region Mayor</p>  | No se detecto señal | Aumentar volumen |
| 6 | <p>Imagen en RGB</p>  | <p>Region Mayor</p>  | No se detecto señal | No se detecto señal |
| 7 | <p>Imagen en RGB</p>  | <p>Region Mayor</p>  | No se detecto señal | Aumentar volumen |

| | | | | |
|----|--|---|---------------------|---------------------|
| 8 | <p>Imagen en RGB</p>  | <p>Región Mayor</p>  | Canal en aumento | Canal en aumento |
| 9 | <p>Imagen en RGB</p>  | <p>Región Mayor</p>  | Aumentar volumen | Canal en decremento |
| 10 | <p>Imagen en RGB</p>  | <p>Región Mayor</p>  | Disminuir volumen | Disminuir volumen |
| 11 | <p>Imagen en RGB</p>  | <p>Región Mayor</p>  | No se detecto señal | No se detecto señal |
| 12 | <p>Imagen en RGB</p>  | <p>Región Mayor</p>  | Aumentar volumen | Disminuir volumen |
| 13 | <p>Imagen en RGB</p>  | <p>Región Mayor</p>  | No se detecto señal | Aumentar volumen |






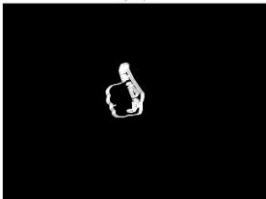





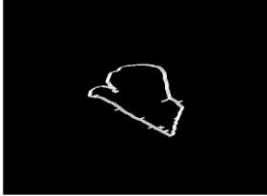

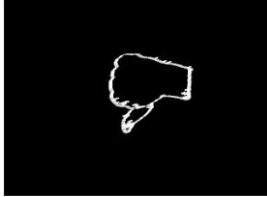

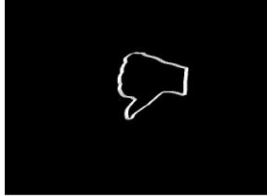

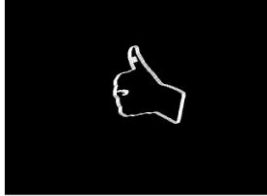



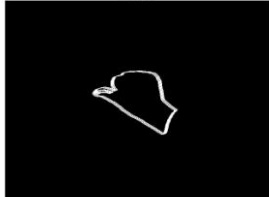









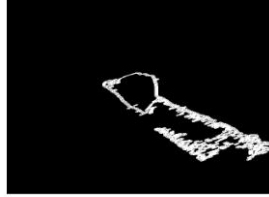
| | | | | |
|----|---|---|------------------|---------------------|
| 14 |  |  | Aumentar volumen | Canal en aumento |
| 15 |  |  | Canal en aumento | No hay señal valida |

Tabla 5.3 Secuencia de imágenes tomadas con la cámara web, con poca iluminación y fondo controlado.

Ahora presenta una secuencia de imágenes tomadas con una cámara web y buena iluminación en la habitación y fondo controlado.

| | Imagen captada | Imagen final | Respuesta del sistema | Respuesta real |
|---|---|---|-----------------------|------------------|
| 1 |  |  | Canal en aumento | Canal en aumento |
| 2 |  |  | Canal en aumento | Canal en aumento |

| | | | | |
|---|---|---|------------------------|------------------------|
| 3 |  |  | Aumentar volumen | Aumentar volumen |
| 4 |  |  | Disminuir volumen | Disminuir volumen |
| 5 |  |  | Canal en decremento | Canal en decremento |
| 6 |  |  | Canal en decremento | Canal en decremento |
| 7 |  |  | Canal en aumento | Canal en aumento |
| 8 |  |  | Aumentar volumen | Aumentar volumen |

| | | | | |
|----|---|---|---------------------------|------------------------|
| 9 |  |  | Disminuir volumen | Disminuir volumen |
| 10 |  |  | No se detecto señal | No hay señal válida |
| 11 |  |  | Aumentar volumen | No hay señal válida |
| 12 |  |  | Aumentar volumen | Aumentar volumen |
| 13 |  |  | Canal en decremento | Canal en decremento |
| 14 |  |  | Aumentar volumen | Disminuir volumen |


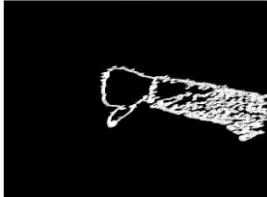


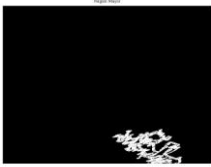








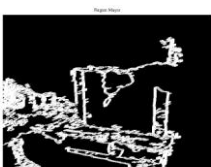





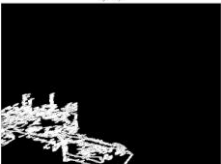















| | | | | |
|----|---|---|---------------------|---------------------|
| 15 |  |  | No se detecto señal | Canal en decremento |
|----|---|---|---------------------|---------------------|

Tabla 5.4 Secuencia de imágenes tomadas con una cámara digital, con buena iluminación y fondo controlado.

Finalmente se muestra una secuencia de imágenes tomadas con una cámara web, buena iluminación y fondo no controlado.

| | Imagen captada | Detección de piel | Imagen final | Respuesta del sistema | Respuesta real |
|---|---|---|--|-----------------------|---------------------|
| 1 |  |  |  | No se detecto señal | Canal en decremento |
| 2 |  |  |  | Aumentar volumen | Aumentar volumen |
| 3 |  |  |  | No se detecto señal | Disminuir volumen |
| 4 |  |  |  | No se detecto señal | Canal en decremento |

| | | | | | |
|----|---|---|--|------------------------|------------------------|
| 5 |  |  |  | No se detecto señal | Disminuir volumen |
| 6 |  |  |  | No se detecto señal | No hay señal válida |
| 7 |  |  |  | Aumentar volumen | No hay señal válida |
| 8 |  |  |  | Aumentar volumen | Canal en aumento |
| 9 |  |  |  | Canal en aumento | Canal en aumento |
| 10 |  |  |  | No se detecto señal | Canal en aumento |
| 11 |  |  |  | Aumentar volumen | Aumentar volumen |













| | | | | | |
|----|--|--|---|------------------------|------------------------|
| 12 |  |  |  | Canal en aumento | Canal en aumento |
| 13 |  |  |  | Canal en decremento | Canal en decremento |
| 14 |  |  |  | Canal en aumento | Canal en aumento |
| 15 |  |  |  | No se detecto señal | Canal en aumento |

Tabla 5.5 Secuencia de imágenes tomadas con una cámara digital con buena iluminación y fondo no controlado.

La presente tabla muestra los resultados obtenidos:

| | No. de imágenes | Aciertos | Errores |
|---|-----------------|----------|---------|
| a) Imágenes tomadas con la cámara web, fondo controlado y poca iluminación. | 15 | 9 | 6 |
| b) Imágenes tomadas con un cámara digital, fondo controlado y buena iluminación | 15 | 13 | 2 |
| c) Imágenes tomadas con una cámara digital, fondo no controlado y buena iluminación | 15 | 7 | 8 |

Tabla 5.6 Resultados obtenidos de las diferentes secuencias de imágenes.

En el caso a) de la tabla 5.3 las imágenes obtenidas por la cámara web, carecen de buena resolución, aunque el fondo de la imagen a nuestra percepción es negro, ciertas áreas de éstas imágenes muestran otras tonalidades, lo que ocasiona una mala segmentación. Por ejemplo para la imagen 9 mostrada en la tabla 5.4 se obtuvieron los resultados mostrados en la figura 5.1, donde el pulgar de la mano es detectado una región separada del puño, lo que ocasiona que cuando se obtiene la región mayor, el pulgar no sea tomado en cuenta, ocasionando un resultado erróneo.

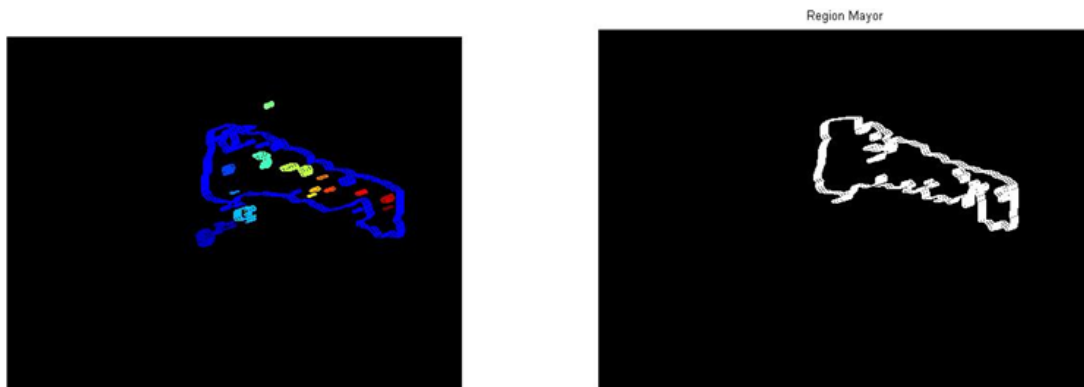


Figura 5.1. Regiones detectadas en la imagen 9 de la tabla 5.3

En el caso b) la mayoría de las imágenes no muestran errores en la segmentación, aunque en las imágenes 14 y 15, el suéter que aparece, muestra en algunas regiones, similitud con el color de la piel y en la detección, se obtiene la imagen mostrada en la figura 5.2. Por otra parte aún cuando la mano es detectada, la función implementada para la detección del pico, detecta el primer pixel blanco que encuentra en la imagen y de ahí continúa en la búsqueda del pico, como el orden de búsqueda del pico es: pico hacia arriba, pico hacia la derecha, pico hacia abajo y pico hacia la izquierda, entonces en la figura 5.2 se detecta el pico hacia la derecha, y ya no se analizan los demás casos por lo que este es el resultado arrojado.



Figura 5.2. (De izquierda a derecha) Región de piel detectada en la imagen 14 de la tabla 5.2. Borde de la imagen.

Finalmente para el caso c) se tienen imágenes tomadas con una cámara web con resolución de 7.1MP, con fondo no controlado. Como se puede notar en el

proceso de detección de piel, la mano es reconocida, sin embargo este proceso detecta, colores que se asemejan a la piel, y en el caso de que la persona aparezca en la imagen tomada, como es el caso de la imagen 15, se detectan más regiones de piel, (en este caso el rostro), aún cuando la mano es detectada en la imagen, se toma la región con mayor área detectada se construye una nueva imagen con dicha región y se inicia la búsqueda del pico.

En la mayoría de las imágenes mostradas en la tabla 5.3 existen regiones detectadas con área superior a la región de la mano y esto no permite determinar si existe alguna señal válida para el sistema en la imagen.

Capítulo VI

Conclusiones y Trabajo futuro

Conclusiones:

Se creo un sistema prototipo que es capaz de controlar un televisor mediante la identificación de señales hechas con la mano de una persona, estas señales son captadas por una cámara web y son tratadas, en caso de dichas imágenes posean alguna señal válida, la señal correspondiente es enviada mediante el puerto paralelo hacia el control remoto universal y éste finalmente envía la señal indicada al televisor.

La solución en software se desarrolló en MATLAB, ya que posee un toolbox de procesamiento de imágenes que facilita la manipulación, procesamiento y análisis de imágenes. Si se pretendiera extenderlo o hacerlo más eficiente en un futuro, la modificación de código en MATLAB se puede realizar fácilmente.

El sistema aún tiene limitaciones, pero aún con esto, en el trabajo se muestra que la metodología propuesta es correcta y hasta este punto de desarrollo es capaz de obtener buenos resultados.

Se espera que este sistema prototipo pueda ser mejorado en un futuro, tanto en el desarrollo en software, como en el desarrollo en hardware.

Trabajo futuro:

Inicialmente este trabajo se pensó como una base para posibles trabajos posteriores. Como se ha mencionado anteriormente el sistema aún cuenta con varias limitaciones, entre ellas se pueden proponer algunas mejoras tales como:

- Mejorar el tratamiento de las imágenes, eliminando ruido y utilizando técnicas que permitan detectar de una manera más eficiente las regiones de piel y la identificación de una mano en una imagen.
- Mejorar el proceso de búsqueda de señales en las imágenes, tomando en cuenta más características, tales como forma y proporción.
- Extender el número de señales identificadas, el sistema sólo identifica cuatro señales y utiliza sólo cuatro de las ocho líneas de datos disponibles en el puerto paralelo, así que en principio se pueden añadir cuatro señales más para utilizar las ocho líneas de datos del puerto paralelo.
- Buscar alguna otra manera de enviar señales a un dispositivo electrónico, que no sea el puerto paralelo, ya que muchas de las computadoras actuales, ya no poseen esta entrada.
- Controlar más dispositivos utilizando esta metodología, si es esto fue posible con un televisor, en principio debería ser posible con algún otro aparato electrónico que sea controlado mediante un control remoto.
- Programar un dispositivo (FPGA) que permita minimizar espacio y mejorar el tiempo de ejecución y que pueda ser colocado en un dispositivo electrónico fácilmente.

Apéndice

El Puerto paralelo

Puerto Paralelo

Un puerto paralelo es una interfaz entre la computadora y un periférico cuya principal característica es que los bits de datos viajan juntos enviando un byte completo o más a la vez. Es decir, se implementa un cable o una vía física para cada bit de datos formando un bus [9].

El cable paralelo es el conector físico entre el puerto paralelo y el periférico.



Figura a1. Cable paralelo.

Originalmente, el puerto paralelo de la PC, fue pensado para conectar impresoras, pero puede ser usado para manejar dispositivos en general.

Transmisión de datos

Los puertos de comunicación de la PC son de particular interés para el estudio de la electrónica, ya que permiten utilizar una computadora personal para controlar circuitos electrónicos, utilizados principalmente en actividades de automatización de procesos, adquisición de datos, tareas repetitivas y otras actividades que demandan precisión.

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.

La mayoría de los sistemas paralelos utilizan ocho líneas de datos para transmitir un byte a la vez, existen excepciones, pero en el presente trabajo nos concentraremos en transferencias de 8 bits que es la configuración del puerto paralelo de la PC.

Un típico sistema de comunicación en paralelo puede ser de una dirección (unidireccional) o de dos direcciones (bidireccional).⁴

El sistema unidireccional es el más simple, consiste de dos elementos: la parte transmisora y la parte receptora. La parte transmisora coloca la información en las líneas de datos e informa a la parte receptora que la información (los datos) está disponible, entonces la parte receptora lee la información en las líneas de datos e informa a la parte transmisora que ha tomado la información (los datos).

Ambas partes sincronizan su respectivo acceso a las líneas de datos, la parte receptora no leerá las líneas de datos hasta que la parte transmisora se lo indique, en tanto que la parte transmisora no colocará nueva información en las líneas de datos hasta que la parte receptora remueva la información y le indique a la parte transmisora que ya ha tomado los datos, a esta coordinación de operaciones se le llama acuerdo o entendimiento (handshaking).

El hardware del Puerto Paralelo

El puerto paralelo de una típica PC utiliza un conector hembra de tipo D de 25 patitas (DB-25 S), es el caso más común, y es el que se utilizó para el desarrollo del sistema [10].

⁴ Para el desarrollo de este trabajo, se utilizó el sistema unidireccional que se describe con mayor detalle y sólo se hace mención del sistema bidireccional.

El orden de los pines del conector es el siguiente:

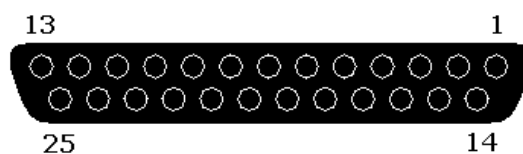


Figura a2. Pines del puerto paralelo.

La siguiente tabla describe la función de cada pin del conector 1284 tipo A:

| Pin | E/S | Polaridad activa | Descripción |
|-------|---------|------------------|--|
| 1 | Salida | 0 | Strobe |
| 2 - 9 | Salida | - | Líneas de datos (bit 0/pin 2, bit 7/pin 9) |
| 10 | Entrada | 0 | Línea acknowledge (activa cuando el sistema remoto toma datos) |
| 11 | Entrada | 0 | Línea busy (si está activa, el sistema remoto no acepta datos) |
| 12 | Entrada | 1 | Línea Falta de papel (si está activa, falta papel en la impresora) |
| 13 | Entrada | 1 | Línea Select (si está activa, la impresora se ha seleccionado) |
| 14 | Salida | 0 | Línea Autofeed (si está activa, la impresora inserta una nueva línea por cada retorno de carro) |
| 15 | Entrada | 0 | Línea Error (si está activa, hay un error en la impresora) |
| 16 | Salida | 0 | Línea Init (Si se mantiene activa por al menos 50 micro-segundos, ésta señal autoinicializa la impresora) |

| | | | |
|---------|--------|---|--|
| 17 | Salida | 0 | Línea Select input (Cuando está inactiva, obliga a la impresora a salir de línea) |
| 18 – 25 | - | - | Tierra eléctrica |

Tabla a2. Descripción de los pines del puerto paralelo.

El puerto paralelo tiene 12 líneas de salida (8 líneas de datos, strobe, autofeed, init, y select input) y 5 de entrada (acknowledge, busy, falta de papel, select y error).

Interfaz del Puerto Paralelo

Eléctricamente el puerto paralelo entrega señales TTL y como tal, teóricamente se le puede conectar cualquier dispositivo que cumpla con los niveles de voltaje específicos de la lógica TTL, sin embargo el hardware del puerto paralelo está muy limitado en cuanto a su capacidad de manejo de corriente, por ésta razón se debe ser muy cuidadoso con el manejo de las señales del puerto, un corto circuito puede dañar permanentemente la tarjeta madre del PC.

Manejo del Puerto Paralelo con MATLAB

MATLAB sólo cuenta con la funcionalidad de manejar el puerto serial y no proporciona el soporte para manejar el puerto paralelo, para hacerlo se necesita una herramienta externa que lo habilite.

Para utilizar el puerto paralelo desde MATLAB es necesario utilizar NTPort Library, es una utilidad que permite a las aplicaciones Win32 tener acceso directo

en tiempo real a los puertos de entrada y salida (I/O). NTPort Library puede ser cargada y descargada dinámicamente, sin necesidad de configuración alguna.⁵

NTPort Library es provista como un DLL estándar así que puede ser utilizada por la mayoría de los lenguajes de programación.

Una vez cargada, con la función **matport** es posible enviar datos o recibirlos desde algún puerto que se especifica en los parámetros.

El prototipo de la función es el siguiente:

```
function [result]= matport('function', parameters)
```

Por ejemplo:

```
myvalue = matport('inport',97);
```

Lee del puerto 97 y el resultado se almacena en la variable myvalue.

Si se requiere enviar algún dato por algún puerto, se realiza de la siguiente manera:

```
matport('outp', 888, 7);
```

En este caso se envía el valor de 7 por el puerto 888 del puerto paralelo.

⁵ En el siguiente link puede descargar NTPort Library 2.6

http://www.freedownloadcenter.com/es/Programacion/Componentes_y_Bibliotecas/NTPort_Library.html

Los lenguajes soportados son: Visual Basic(VB), Visual Basic.NET, Visual C#, Visual C+, Delphi, Delphi for .NET, Borland C++ Builder, C#Builder, JBuilder, y PowerBASIC, Matlab, Borland C++, Visual Fortran, Visual J++, FreeBASIC, ActiveTCL(TCL/TK).

Referencias

- [1] Pajares G., De la Cruz J., (2002). *Visión por computador. Imágenes digitales y aplicaciones*. Alfa Omega.
- [2] De la Escalera A., (2001). *Visión por computador*. Prentice Hall.
- [3] Curso Computación Gráfica II.
<http://www.ciudadanokane.cl/cursos/apunte1.pdf>
- [4] Cáceres J., *La visión artificial y las operaciones morfológicas en imágenes binarias*,
<http://www2.uah.es/jcaceres/uploaded/articulos/FormatoVisonArtificial.pdf>
- [5] Modelo de color RGB., http://es.wikipedia.org/wiki/Modelo_de_color_RGB
- [6] Nociones de Colorimetría. <http://www.depeca.uah.es/docencia/IT-INF/sva/t1-colorimetria.pdf>
- [7] Mando a distancia. http://es.wikipedia.org/wiki/Control_remoto
- [8] Ramello P., (2005). *Comparación de métodos de detección de piel en modelos de color YCbCr y HSI para reconocimiento de caras*. http://cpdsi-fich.wdfiles.com/local--files/tpsaplicacion/2005_Ramello-Piel.pdf
- [9] Puerto paralelo. http://es.wikipedia.org/wiki/Puerto_paralelo
- [10] El puerto paralelo de la PC.
<http://www.modelo.edu.mx/univ/virtech/circuito/paralelo.htm>
- [11] Color luz y color pigmento.
<http://www.fotonostra.com/grafico/colorluzpigmento.htm>