RECONOCIMIENTO E INTERPRETACIÓN DE GESTOS MANUALES POR MEDIO DE VIDEO.

NÉSTOR ORLANDO BALSERO GARCÍA
DIEGO ANDRÉS BOTERO GALEANO
JUAN PABLO ZULUAGA MORALES

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.
2005

RECONOCIMIENTO E INTERPRETACIÓN DE GESTOS MANUALES POR MEDIO DE VIDEO.

NÉSTOR ORLANDO BALSERO GARCÍA
DIEGO ANDRÉS BOTERO GALEANO
JUAN PABLO ZULUAGA MORALES

Informe final del Trabajo de Grado presentado para optar al título de Ingeniero Electrónico

Director
CARLOS ALBERTO PARRA RODRÍGUEZ
Ingeniero Electrónico, Ph.D.

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.
2005

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERIA ELECTRÓNICA

RECTOR MAGNÍFICO: R.P. GERARDO REMOLINA VARGAS S.J.

DECANO ACADÉMICO: Ing. FRANCISCO JAVIER REBOLLEDO

DECANO DEL MEDIO UNIVERSITARIO: R.P. ANTONIO J. SARMIENTO S.J.

DIRECTOR DE CARRERA: Ing. JUAN CARLOS GIRALDO

DIRECTOR DEL PROYECTO: Ing. CARLOS ALBERTO PARRA RODRÍGUEZ, Ph.D.

ARTÍCULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946

“La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.
Solo velará porque no se publique nada contrario al dogma y a la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia.”

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS:

INTRODUCCIÓN

En este reporte se presenta la implementación de un sistema en el cual, mediante el tratamiento digital de imágenes, capturadas de una secuencia de video, se determina un conjunto de órdenes básicas realizadas por un interlocutor con sus manos, estas órdenes son representadas y quedan disponibles para múltiples aplicaciones.

El proyecto se implementó en la tarjeta de evaluación ADSP-BF533 EZKIT lite de Analog Devices, la cual se programa para que tenga la capacidad de interactuar con un usuario en tiempo real[1], mediante 14 gestos generados con las manos,los cuales constituyen un nuevo alfabeto. De la misma manera, se implementa en el entorno de programación Visual C++ un algoritmo de soporte al sistema desarrollado, permitiendo la visualización en tiempo real de imágenes resultantes de diferentes etapas del procesamiento y el resultado del reconocimiento, constituyéndose entre otras en una herramienta para el aprendizaje, por parte del usuario, de los gestos pertenecientes al alfabeto desarrollado.

Este nuevo alfabeto está basado en modelar los dedos de la mano como bits, así con una sola mano se pueden obtener 32 gestos, con la posibilidad de ampliar el alfabeto hasta 64 gestos con una sola mano en el caso ideal y más de 2000 gestos con las dos manos, se descartan algunos gestos por limitaciones fisiológicas de la mano.

El sistema implementado es robusto a rotaciones, translaciones, cambios de escala de las manos del interlocutor en el plano de la cámara.

[1] Para esta aplicación se considera tiempo real aproximadamente 100 ms.

Como aplicación final se visualiza en un display siete segmentos el símbolo asignado al gesto reconocido.
El reconocimiento de gestos a partir de una secuencia de video permite interactuar con máquinas de manera eficiente y a distancia, abriendo una gama variada de aplicaciones, entre las cuales se encuentran: control de brazos robots a distancia, realidad virtual, control de procesos industriales, interacción con aplicaciones de Windows, entre otros.

En este informe se presentarán, en primer lugar, ciertos conceptos teóricos que permiten la comprensión del trabajo, abordando en segundo lugar las especificaciones del sistema (realizando una descripción tanto del software como del hardware implicados), siguiendo con la explicación de los desarrollos implementados y terminando con el análisis de las pruebas de desempeño efectuadas al sistema.

1. DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema implementado parte del procesamiento de una secuencia de video, en la que se encuentra una persona usando camisa oscura de manga larga gesticulando en primer plano, capturada por una cámara bajo condiciones de iluminación y fondo controladas.

Cada imagen de la secuencia de video es segmentada para determinar la región en la que se encuentra la mano del interlocutor gestual (región de la imagen que va a ser procesada). En la región de interés se realiza un proceso de adelgazamiento para limitar la cantidad de información a procesar y para permitir el éxito de la etapa de reconocimiento. Una vez la imagen es adelgazada, se identifican puntos de interés cuya posición respecto al centro de masa de la mano segmentada determinan un vector que se compara con un conjunto de vectores base, previamente establecidos en la etapa de entrenamiento del sistema para un posterior reconocimiento.

La fase de entrenamiento es una etapa del diseño del sistema para establecer los vectores base, mediante una serie de pruebas y análisis estadísticos de los resultados obtenidos al aplicar el algoritmo (que será descrito posteriormente) en diferentes interlocutores. Se establecen los vectores base a partir de imágenes de diferentes personas (hombres y mujeres), para desarrollar un sistema funcional para una alto porcentaje de la población.

El sistema propuesto es implementado inicialmente en un computador personal en el entorno de programación Visual C++ mediante la evaluación de diferentes

algoritmos computacionalmente eficientes, viables para su implementación en tiempo real en un procesador embebido con recursos de hardware más limitados.

En la figura 1 se muestra el diagrama en bloques simplificado del proceso realizado.

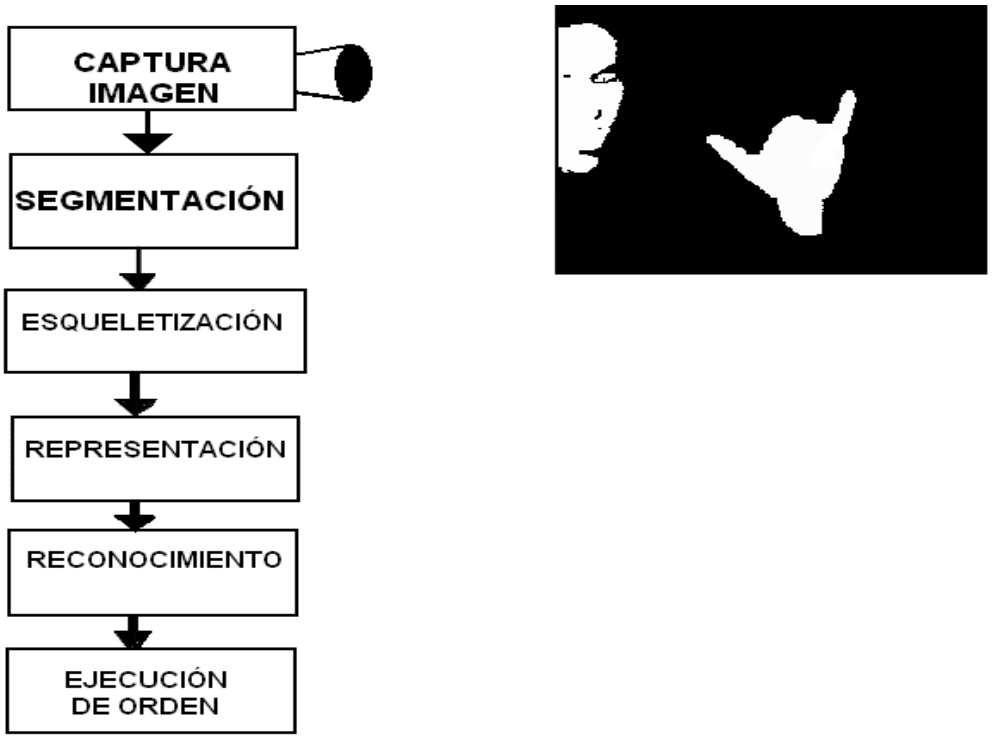


Figura 1. Diagrama en bloques general del proceso

2. MARCO TEÓRICO

2.1. Elementos de un sistema de procesamiento de imágenes

Los elementos usuales de un sistema de procesamiento de imágenes son los siguientes: captura, almacenamiento, procesamiento, reconocimiento y presentación.

En la fase de captura los rayos reflejados por los objetos deben ser capturados y convertidos en una señal eléctrica para poder ser procesados, esta tarea es realizada por la cámara. Si el dispositivo de captura es una cámara analógica, entonces la señal eléctrica que produce generalmente está en uno de los dos estándares de video más comunes: NTSC[2] y PAL[3]. Esta señal análoga se debe digitalizar y decodificar para poder aplicar las distintas técnicas de procesamiento de imágenes y transformaciones morfológicas. Estas permiten su segmentación y la posible extracción de las características que llevan a la etapa de interpretación y reconocimiento de patrones **[1].** Una vez realizado todo el análisis de la imagen es importante representar de manera gráfica los resultados obtenidos, esto mediante el diseño de una interfaz gráfica como salida del sistema.

Durante el procesamiento digital de imágenes se requiere el análisis de un amplio conjunto de elementos que van desde las fuentes de luz, necesarias para iluminar los objetos hasta los algoritmos encargados de clasificarlos, reconocerlos e interpretarlos.

[2] NTSC: estándar americano de video, siglas de National Television System Committee,
[3] PAL: estándar europeo de video, siglas de Phase Alternation Line Ratio

Uno de los factores determinantes para la adecuada captura de las imágenes es la iluminación, con ésta, se busca resaltar los aspectos de la escena que interesen en la respectiva aplicación, garantizando que la cámara esté capturando los objetos siempre con la misma intensidad.

2.1.1. Iluminación

La iluminación existente en el entorno no siempre es la adecuada ya que se obtienen imágenes con bajo contraste, es decir, poco o demasiado iluminadas. Esto conlleva a problemas como sombras no deseadas, que afectan directamente la complejidad del algoritmo**[2].**

Para definir claramente el tipo de iluminación es necesario identificar las propiedades de los objetos que intervienen en la escena, pueden ocurrir tres situaciones cuando un haz luminoso incide sobre un material; que se refleje, se absorba o se transmita a través de él.

2.1.2. Tipos de iluminación

- Iluminación direccional: Esta técnica consiste en aplicar una iluminación orientada al objeto usando un haz altamente direccional. Es óptima en aplicaciones que buscan el reconocimiento de objetos ya que define claramente las distintas regiones de una imagen. Este tipo es de las más utilizadas debido a su fácil uso**[2].**

- Iluminación difusa: Con este tipo de iluminación se intenta que los haces luminosos incidan sobre el objeto desde todas las direcciones. Se utiliza cuando se van ha tomar imágenes desde diversos puntos de vista, por lo que no se pueden favorecer a unas zonas más que otras**[2].**

- Iluminación a contra luz: Consiste en iluminar el objeto por detrás de forma que la fuente luminosa, el objeto y la cámara estén alineados. Esta técnica genera imágenes prácticamente con sólo dos niveles de gris, es decir, imágenes binarias**[2].**

- Iluminación oblicua: puede considerarse un caso particular de la iluminación direccional. El objetivo principal es la creación de sombras encargadas de aumentar el contraste de las partes tridimensionales. Este tipo de iluminación es utilizado para generar sombras sobre objetos cuyo contraste es pequeño respecto al fondo**[2].**

2.1.3. Profundidad de campo

Durante el proceso de captura de la imagen se necesita que el interlocutor esté dentro de una zona en la cual el objeto forme una imagen que está dentro de la profundidad de enfoque[4]**[2].**

2.2. Imágenes digitales

El concepto de imagen esta asociado a una función bidimensional $f(x, y)$, cuya amplitud o valor será el grado de iluminación 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, determinada por la fuente, y reflexión, que depende de las características del objeto en la escena**[4].**

[4] Intervalo de distancia en el cual los objetos se ven nítidos.

2.2.1. Muestreo y cuantificación de la imagen

Durante el proceso de adquisición de imágenes resulta interesante evaluar dos factores que pueden producir pérdida significativa de información y resultan determinantes a la hora de realizar su posterior procesamiento. Por una parte, el muestreo de una imagen tiene el efecto de reducir la resolución espacial de la misma, por lo que con el aumento del paso de muestreo se puede observar notoriamente la pérdida de la información así como de la generación de ruido.

El efecto de la cuantificación viene dado por la imposibilidad de tener un rango infinito de valores de medida para la intensidad de brillo de los píxeles, este tema es primordial en trabajos que deseen realizar o extraer características a partir del color de la imagen. La tecnología actual permite trabajar con ocho bits de información o equivalentemente 256 niveles de gris para codificar este valor lumínico **[4].**

2.2.2. Relaciones entre píxeles

Un píxel p de coordenadas (x, y) presenta un total de cuatro vecinos en el plano vertical y horizontal, siendo sus coordenadas las representadas en la figura 2.

	$x, y-1$	
$x-1, y$	x, y	$x+1, y$
	$x, y+1$	

Figura 2. Píxel de interés y su conectividad tipo 4

Este conjunto de píxeles se denomina vecindad de tipo 4 del píxel p, se denota $N_4(p)$ **[2].**

Además se puede considerar la existencia de otros cuatro vecinos asociados a las diagonales, cuyas coordenadas se visualizan en la figura 3.

$x-1, y-1$		$x+1, y-1$
	x, y	
$x-1, y+1$		$x+1, y+1$

Figura 3. Píxel de interés y sus vecinos diagonales

La suma de los anteriores define los ocho vecinos del píxel p, se denomina vecindad de tipo 8 y se denota $N_8(p)$ **[2].**

La conectividad entre píxeles es un concepto importante usado para establecer las fronteras de objetos y las regiones componentes de una imagen. Para establecer si dos píxeles están conectados hemos de establecer si son adyacentes en algún sentido.

2.2.3 Conectividad

La conectividad entre píxeles es un concepto importante usado para establecer las fronteras de los objetos y las regiones componentes de una imagen. Para establecer si dos píxeles están conectados hemos de establecer si son adyacentes en algún sentido (por ejemplo si son 4-vecinos y si sus niveles de gris cumplen algún criterio de similitud). Así en una imagen binaria con valores 0 y 1, dos píxeles pueden ser 4-vecinos y no estar conectados salvo que tengan el mismo valor **[2]**.

2.2.4. Distancia

Con la distancia se quiere obtener el mínimo número de pasos elementales que se necesitan para ir de un punto a otro. Dados tres píxeles *p*, *q* y *z* con coordenadas $(x, y), (s, t)$ y (u, v) respectivamente, se puede definir una función de distancia D si cumple:

- $D(p, q) \geq 0, (D(p, q) = 0, \Rightarrow p = q)$
- $D(p, q) = D(q, p)$
- $D(p, z) \leq D(p, q) + D(q, z)$

Las funciones de distancia usadas comúnmente son:

- La distancia euclídea entre *p* y *q* se define como:

$$D_E(p, q) = \sqrt{(x - s)^2 + (y - t)^2}$$

Para esta medida de distancia, los píxeles están a una distancia r de un píxel dado, definiendo un disco de radio r centrado en el punto *(p,q)* **[4].**

- Distancia Manhattan: Se toman solamente en cuenta los vecinos de orden 4.

$$D = |x - s| + |y - t|$$

- Distancia tablero de ajedrez: En esta distancia también se tiene en cuenta los vecinos de tipo 8.

$$D(p, q) = \max(x - s, y - t)$$

2.3. Sistemas de representación del color

El ojo humano percibe los colores según la longitud de onda de la luz que a él llega, permitiendo dividir el espacio de color en seis regiones: violeta, azul, verde, amarillo, naranja y rojo.

La caracterización de la luz en una escena de color es fundamental. Si la luz es acromática, su único atributo es su intensidad, donde la luz que contiene la mayor intensidad aparece como luz blanca, mientras que la ausencia de luz es percibida como color negro. Por otra parte la luz cromática expande el espectro electromagnético y permite realizar el análisis de la escena a partir de tres componentes: luminancia, brillo y resplandor. La luminancia es medida en lumens (lm), expresa la cantidad de energía que el observador percibe de la fuente de luz. El brillo consiste en la claridad u oscuridad relativa a cada tono de color. El resplandor, es la cantidad total de energía que fluye de la fuente de luz **[3].**

Parece claro que todo lo relativo al color posee un fuerte componente subjetivo al intervenir tanto la percepción visual de cada persona como la interpretación por parte del cerebro humano de esta información visual. Sin embargo, también parece claro que se necesita un patrón de referencia, una descripción objetiva que sea capaz de describir el espectro de color al margen de las circunstancias personales. Es así como en el año 1931 fue definido por la CIE (*Commission Internationale de l'Eclairage*) el primer estándar de valores de longitud de onda para los colores rojo, verde, azul. Estos colores, conocidos como primarios, permiten expresar cualquier otro color a partir de su combinación. Algunos de los métodos de visualización desarrollados por de la CIE son denominados espacios de color, entre los que se encuentran los siguientes:

2.3.1. RGB

El espacio RGB se basa en la combinación de tres señales de luminancia cromática distinta: El rojo, el verde y el azul (Red, Green, Blue). La manera más sencilla e intuitiva de conseguir un color concreto es determinar la cantidad de color rojo, verde y azul que se necesita combinar, para ello se realiza la suma aritmética de las tres componentes **[3]**.

2.3.2. YUV

Es un espacio de color en el que Y representa la luminancia (brillo), U y V son componentes de crominancia (color). YUV se crea a partir del R, G, B. Los valores ponderados de R, G, B son sumados para producir Y, U resulta de la resta entre Y y la componente B del espacio R, G, B, con una escalización posterior y V de la resta entre Y y la componente roja de R, G, B, con una posterior escalización diferente a la correspondiente a U **[3]**. Se define así:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

2.3.3. YCbCr

YCbCr es un espacio de color desarrollado con base en la recomendación de la ITU-R BT 601 del estándar mundial de video digital. Es una versión escalizada del espacio YUV. El espacio RGB se separa en una parte de luminancia (Y) y dos partes de crominancia (Cb y Cr) **[3]**. Se define así:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$
$$Cr = (R - Y) * 0.713 + 128$$
$$Cb = (B - Y) * 0.564 + 128$$

2.3.4. Lab

Los tres parámetros en el modelo representan la luminancia del color (L, cuyo valor más pequeño corresponde al negro), su posición entre el rojo y verde (a, cuyo valor más pequeño corresponde al verde) y su posición entre el amarillo y el azul (b, cuyo valor más pequeño corresponde al azul) **[3]**.

$$L = 116 * Y^{1/3} \quad \text{para} \quad Y > 0.008856$$
$$L = 903.3 * Y \quad \text{para} \quad Y \leq 0.008856$$
$$a = 500 * (f(x) - f(y))$$
$$b = 200 * (f(y) - f(z))$$

donde,

$$f(t) = t^{1/3} \quad \text{para} \quad t > 0.008856$$
$$f(t) = 7.787 * t + \frac{16}{116} \quad \text{para} \quad t \leq 0.008856$$

Los desarrollos de software gráfico manejan imágenes utilizando uno o varios canales los cuales representan información acerca de uno de los elementos del color en la imagen. El uso de canales hace posible manipular imágenes en sofisticadas formas. El usuario puede ajustar un sólo elemento de color en una imagen, comparar el color en las imágenes separadas analizando y ajustando sus respectivos canales y usar canales para adicionar detalle a una imagen que fácilmente puede modificar y remover después.

Para el manejo de imagen se hace importante conocer los diferentes tipos en los que se almacena y procesan las imágenes.

2.4. Procesamiento digital de imágenes

2.4.1. Pre-procesamiento

Después de adquirir la imagen el pre-procesamiento es el segundo paso del procesamiento digital de imágenes. Es muy útil porque ayuda a suprimir información que no es relevante para los objetivos particulares de análisis en un caso dado. Típicamente, una imagen tiene la propiedad de que un píxel tiene un valor cercano al de sus vecinos. Toda imagen se ve inmersa en ruido que se puede caracterizar como aditivo blanco gaussiano (AWGN) que puede afectar los píxeles de manera que la propiedad anteriormente mencionada no se cumpla. Para limitar el efecto del ruido se puede realizar un proceso de suavizado o realce de la imagen. Y como este, se puede aplicar muchos otros filtros según las aplicaciones, para limitar el efecto del ruido.

Por lo tanto, el objetivo del pre-procesamiento es una mejora de los datos de la imagen que suprima las distorsiones indeseadas e incremente las características relevantes para su posterior procesamiento **[4]**.

2.4.2. Segmentación de imágenes

La segmentación de imágenes tiene su origen en numerosos estudios psicológicos que indican la preferencia de los humanos por agrupar regiones visuales en términos de proximidad, similitud y continuidad, para constituir un conjunto de unidades significativas. Estas propiedades no son fáciles de cumplir y sin embargo son fundamentales para una buena segmentación **[4]**.

En general la segmentación es una de las tareas más significativas en el procesamiento de imágenes. Este paso del proceso determina el eventual éxito o fracaso del análisis de la imagen.

Los algoritmos de segmentación de imágenes tienen tres formas comunes: métodos basados en bordes, técnicas basadas en regiones y técnicas de umbral.

2.4.2.1. Segmentación por bordes

Se centran en la detección de contornos. Delimitan el borde de un objeto y segmentan los píxeles dentro del contorno como pertenecientes a ese objeto. Su desventaja consiste en conectar contornos separados o incompletos, lo que los hace susceptibles a fallas **[3]**.

2.4.2.2. Segmentación por regiones

En esta aproximación todos los píxeles que correspondan a un objeto se agrupan juntos y son marcados para indicar que pertenecen a una región. Los píxeles son asignados a regiones según algún criterio que los distingue del resto de la imagen. Un criterio muy estricto puede provocar fragmentación mientras uno poco estricto ocasiona uniones indeseadas **[4]**.

2.4.2.3. Segmentación por umbral

Esta técnica segmenta la imagen píxel por píxel, es decir, no toman en consideración el valor de los píxeles vecinos para el proceso. Si el valor de un píxel está dentro de un rango de valores especificado para un objeto el píxel es segmentado. Son efectivas cuando los objetos y el fondo de la imagen tienen rangos de valores diferentes y existe un contraste marcado entre ellos. Como la información de los píxeles vecinos es ignorada, las fronteras de regiones borrosas pueden ocasionar problemas **[4]**.

2.4.3. Representación de imágenes

Después que una imagen ha sido segmentada, el conjunto de píxeles segmentados son usualmente representados y descritos en una estructura adecuada para su posterior procesamiento. La representación es conveniente porque en lugar de procesar una gran cantidad de datos se procesa una estructura sencilla que contiene la información útil para un posterior reconocimiento.

Algunos esquemas de representación son: esqueleto de una región, códigos de cadena, aproximaciones poligonales, entre otros **[2,3,4]**.

2.4.3.1. Esqueletización

Este método remueve información redundante produciendo una imagen más simple, reduciendo el espacio y el tiempo de accesos a memoria facilitando la extracción de características topológicas de la región de interés.

Su propósito es representar la forma de un objeto con un número relativamente pequeño de píxeles, cuyo grosor es de 1 píxel únicamente. De esta forma, todos los píxeles del esqueleto son estructuralmente necesarios para su posterior análisis.

El principio básico de los métodos de adelgazamiento es eliminar repetidamente aquellos puntos del contorno de un objeto de tal manera que la eliminación de dicho punto no afecte la conectividad del objeto y respete la condición de punto final local. El decidir cuando un píxel altera o no la conectividad global del objeto al ser eliminado, se puede determinar haciendo uso de criterios de conectividad locales, considerando la vecindad de dicho píxel. En general, el adelgazamiento no permite la reconstrucción del objeto original.

Cuando se adelgaza una imagen se deben tener en cuenta una serie de limitaciones y consideraciones, entre ellos:

1) Tiempo de procesamiento: Existe un compromiso entre el tiempo de ejecución de los algoritmos y el método de adelgazamiento utilizado.

2) Forma resultante del objeto: Dentro de los problemas mas graves, independiente del método utilizado, es la generación de ramas parásitas que no estén presentes en el objeto original.

El esqueleto de una imagen se puede extraer fácilmente utilizando alguna de las distintas técnicas de adelgazamiento, entre las que se encuentran: transformación de eje medio, y el algoritmo Shan Zuen **[4]**.

2.4.3.1.1. Transformación de eje medio

Esta técnica de esqueletizado se basa en la determinación de aquellos píxeles de la imagen que son equidistantes a la frontera de la misma, formando el llamado eje medio (medial axis). El mayor problema de esta transformación es que no asegura ni la conectividad del esqueleto ni el que tenga un ancho de un píxel, sin embargo su fácil implementación lo hace atractivo cuando se tienen recursos limitados.

No obstante, es posible solucionar el problema de la desconexión entre ramas si posterior a esta transformación se realiza un proceso de enlace.

2.4.3.1.2. Métodos morfológicos iterativos

Esta técnica enfoca el problema desde una perspectiva absolutamente diferente, desgastando los bordes de una imagen binaria hasta llegar al esqueleto. Este desgaste consiste en un proceso iterativo, donde en cada barrido de la imagen se eliminan aquellos puntos cuya vecindad cumple un determinado conjunto de condiciones y se mantienen los demás, terminando el mismo al llegar a un barrido en el cual ningún punto es eliminado.

Unos de los algoritmos de esta técnica de adelgazamiento de gran aceptación por su relativa simplicidad de implementación y su efectividad para conservar las características topológicas, aún bajo traslaciones o rotaciones (de ángulos agudos), de la región de interés, es el algoritmo Zhang Suen [11].

La idea básica del algoritmo radica en determinar los píxeles que pueden ser eliminados analizando sus respectivos ocho vecinos. Se erigen cuatro reglas fundamentales de obligatorio cumplimiento para proceder con el borrado de un píxel de la imagen.

La primera regla establece que un píxel puede ser descartado si y sólo si tiene más de uno y menos de siete vecinos con su mismo valor. Con el cumplimiento de esta regla se asegura que los puntos finales[5] ("end points") de la imagen esqueletizada no son removidos y que los píxeles son descartados partiendo de los contornos y no desde el interior de la región de interés.

La segunda ley reside sobre el concepto de conservación de regiones. En efecto, se instaura la noción del índice de cruce que aplica para regiones de vecindad tipo

[5] end point : píxel de una imagen adelgazada que cumple con la propiedad de tener sólo un vecino tipo 8 con mismo valor.

8. El índice de cruce hace referencia al número de regiones conectadas por un píxel y no al número de vecinos de mismo valor que pueda tener. Para un píxel dado, su índice de cruce es el número de regiones resultantes en caso de ser removido. Así pues, sólo puede ser borrado un píxel cuyo índice de cruce sea la unidad, esto para garantizar que el número de regiones conectadas tipo 8 se conserve luego del proceso de adelgazamiento.

Los resultados del proceso de adelgazamiento son dependientes de la manera en que se realice el barrido por la imagen. Lo ideal sería realizar un barrido de arriba hacia abajo, de izquierda a derecha y promediar con los resultados obtenidos con un barrido de abajo hacia arriba, de derecha a izquierda; pero se incrementaría la complejidad computacional del algoritmo. Como resultado de la observación de los esqueletos resultantes siguiendo diferentes direcciones en el barrido se determinó una muy aceptable conservación de la forma y disposición topológicas realizando dos recorridos sobre la imagen.

Para el análisis de dichos píxeles vecinos se definen las direcciones con respecto al píxel de interés como se evidencia en la figura 4.

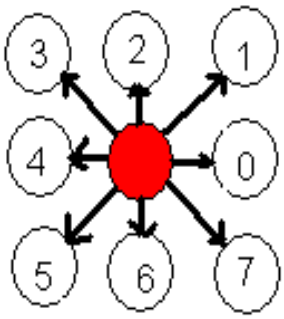


Figura 4. Asignación de direcciones alrededor de un píxel de interés.

En el primer barrido de la imagen, se verifican si los vecinos de un píxel en las direcciones 1, 3, 5 y 3, 5, 7 pertenecen al objeto de interés y en el segundo recorrido se analizan las direcciones 7,1,3 y 1,5,7 del píxel para permitir su remoción. Estas reglas adicionales permiten la obtención de imágenes

adelgazadas que conservan de mejor manera la distribución topológica de la imagen procesada evitando cambios sustanciales como se evidencia en la figura 5 y 6.

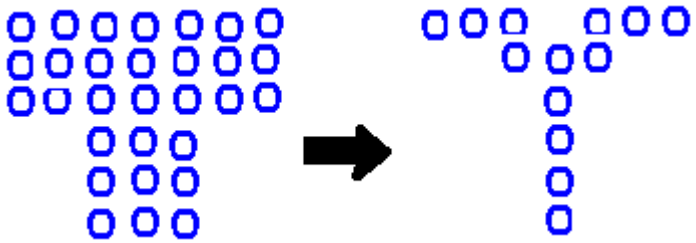


Figura 5. Pérdida de distribución topológica

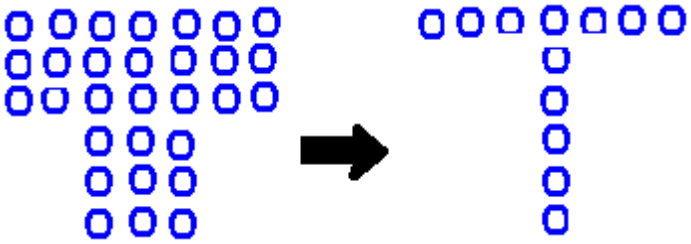


Figura 6. Conservación de distribución topológica.

Aunque de buen desempeño, persiste la sensibilidad del algoritmo a puntos espurios imperfectamente segmentados o contornos poco definidos de la región de interés como se evidencia en la figura 7.

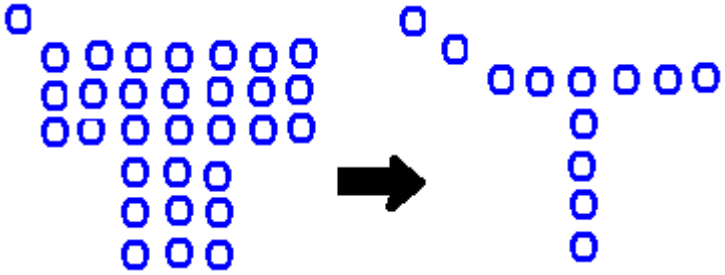


Figura 7. Sensibilidad del algoritmo a puntos espurios

Cuando una iteración completa del algoritmo deja la imagen sin cambio alguno, se termina la operación morfológica deseada y se obtiene el esqueleto de la imagen original.

2.4.4. Reconocimiento e interpretación

La representación paramétrica de curvas es una herramienta computacional muy utilizada para la extracción de características de las mismas o como base para el diseño y modelado de piezas.

Una curva paramétrica es definida por un conjunto de puntos discretos conocidos como puntos de control y un conjunto de funciones base combinados linealmente.

Usando la secuencia de puntos de control se trata de construir de manera precisa una curva con las características deseadas.

En particular, la representación de curvas utilizando funciones base B-Splines es de alta popularidad pues exhiben suficiente flexibilidad morfológica (se puede obtener cualquier forma de curva) y su generación, en términos computacionales, es sencilla para órdenes menores que cuatro.

Una curva se define en términos de su representación con funciones base B-Splines de la siguiente manera:

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathbf{P}_i N_{i,p}(t)$$

donde n es el número total de puntos de control menos 1, i es el subíndice que hace referencia a un punto de control en particular, P_i es el i-ésimo punto de

control y $N_{i,m}$ (t) es la función B-Spline de orden (p-1) correspondiente al i-ésimo punto de control que satisface la siguiente ecuación recursiva:

$$N_{i,0}(t) = \begin{cases} 1 \text{ if } t_i \leq t < t_{i+1} \text{ and } t_i < t_{i+1} \\ 0 \text{ otherwise} \end{cases}$$

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t).$$

Una función B-Splines es una serie de (n-2) segmentos definidos por (n+1) puntos de control. Cada segmento de curva es formado por (p+1) puntos de control y cada punto de control influencia (p+1) segmentos de curva.

Las funciones B-Splines satisfacen ciertas propiedades que tienen que ser tenidas en cuenta en el momento de hacer uso de la representación **[5]**.

- El orden del polinomio es siempre uno menos que el número de puntos de control.
- La curva generada sigue la forma del polígono obtenido uniendo los puntos de control.

- Los puntos escogidos tienen control local sobre la forma de la curva generada.
- La curva generada no pasa forzosamente por todos los puntos de control pero se tiene la certeza de que el primer y último puntos de control son los extremos (“end points”) de la curva representada.

El resultado de cualquier sucesión de transformaciones lineales que sufra una determinada curva se ve manifestado por la misma sucesión de transformaciones lineales realizado al conjunto de puntos de control **[5]**.

Una conveniente escogencia de los puntos de control es fundamental para una apropiada representación de la curva. Se debe realizar un análisis sobre las distintas características que pueden ser extraídas de la curva realizando sólo un estudio sobre sus puntos de control **[4]**.

3. ESPECIFICACIONES

El sistema implementado tiene la capacidad de interactuar con un interlocutor a partir del procesamiento digital de imágenes en tiempo real. Se propone un nuevo alfabeto bastante amplio, del cual se seleccionan catorce gestos (ver anexo C), finalmente se visualiza la letra asociada al gesto reconocido.

El sistema es robusto a rotaciones, translaciones, cambios de escala. El interlocutor debe gesticular en un plano paralelo a la cámara.

El proyecto consta de dos vertientes: una parte de software y otra parte de hardware. La aplicación de software se desarrolla con el objetivo de evaluar los algoritmos que se van a implementar en la tarjeta de evaluación.

La aplicación de software se desarrollo en un computador personal Compaq Pentium 3 con sistema operativo Windows 2000, los procesos de desarrollo del proyecto se implementan dentro del ambiente de programación Visual C++ de Microsoft, como complemento al ambiente de desarrollo se utilizó la biblioteca OpenCv, esta implementa una gran cantidad de algoritmos para el análisis e interpretación de imágenes(ver anexo A). En el computador personal se procesan 20 frames por segundo y como aplicación final se visualizan ventanas que presentan el procesamiento de la imagen como parte demostrativa, así como una ventana con la letra asociada al gesto.

Para la parte de hardware, se usa la tarjeta de evaluación ADSP-BF533 EZKIT lite, esta es una tarjeta que permite evaluar prototipos usando el procesador Blackfin 533 de Analog devices, la tarjeta de evaluación cuenta con múltiples

periféricos que permiten implementar diferentes aplicaciones (ver anexo B). En la DSP se procesan 25 frames por segundo.

La tarjeta tiene un cristal de 27 MHz, el cual es multiplicado por 10 para obtener el reloj del core, y un reloj del sistema de 54 Mhz, (ver figura 9) el procesador y las memorias de primer nivel manejan el reloj del core, mientras los demás dispositivos trabajan con el reloj del sistema. Particularmente la interfaz paralela de periféricos trabaja a la mitad del reloj del sistema.

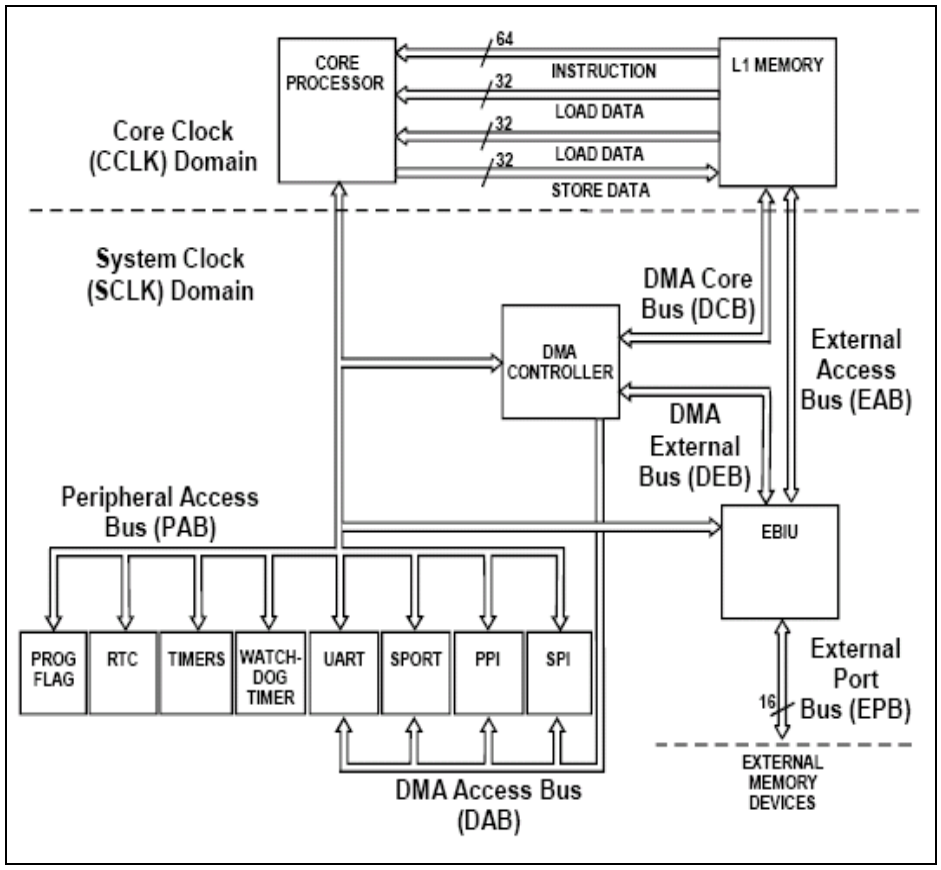


Figura 8. Reloj y sistema de buses[6]

[6] extraída del HARDWARE REFERENCE MANUAL ADSP BF 533 Blackfin Processor (Pág. 6-6).

Como salida del sistema se maneja un display en el cual se presentara la letra asociada al gesto realizado.

Para la aplicación de hardware se necesita una cámara análoga NTSC o PAL, y para la aplicación en el computador personal se usa una cámara digital tipo CMOS.

Las condiciones de iluminación deben ser de tipo direccional, y el fondo debe ser lo más opaco posible.

La mano no debe sobrelaparse con otra región de piel, y se debe usar camisa de manga larga oscura.

4. DESARROLLOS

Es importante resaltar que gran parte del desarrollo de este proyecto es producto de la investigación y por lo tanto algunos de los resultados obtenidos van a ser fruto del criterio de diseño del grupo.

Con el propósito de cumplir los objetivos planteados se realiza una primera parte en Microsoft Visual Studio 6.0, plataforma netamente de software, en ella se busca desarrollar y evaluar diferentes algoritmos, que posteriormente se implementarán en el hardware de propósito específico escogido.

Es conveniente aclarar que se busca desarrollar un sistema a nivel de cómputo eficiente, debido a las limitaciones de hardware de un sistema embebido, y más aún resaltando su orientación a tiempo real.

4.1 Desarrollo Teórico.

En este trabajo se desarrolla un nuevo alfabeto (ver anexo C), para interactuar con una máquina. Este nuevo alfabeto está basado en modelar los dedos de la mano como bits (ver anexo D) de esta forma lo que la máquina debe hacer para interpretar un gesto, es buscar si están o no los dedos y su ubicación relativa con respecto al antebrazo, dando al problema una naturaleza de tipo binario, lo cual es óptimo para cualquier maquina electrónica.

Se desarrolla una representación de la mano que resulta apropiada para un posterior reconocimiento, esto mediante un vector en el que se almacena

información de la morfología que presenta la mano. El vector que describe la mano es generado a partir de encontrar la ubicación de detalles en la imagen, similar a los algoritmos de reconocimiento de huellas.

Se determinan puntos estratégicos de la mano que permiten describir la mano de diferentes interlocutores. Entre estos se encuentra el centro de masa el cual se logra volver más estático a medida que se gesticula ampliando la región descubierta del antebrazo (lo cual es obvio porque hay mayor área).

4.1.1. Algoritmos de segmentación

Para el desarrollo de este módulo es de vital importancia trabajar con una adecuada iluminación. La técnica de iluminación implementada fue totalmente direccional a la mano del interlocutor.

Teniendo la iluminación controlada se trata de rescatar la información propia de las zonas más brillantes que en este caso son las manos. Para hacer más evidente el contraste entre las manos y las zonas correspondientes al fondo, utilizar un fondo negro opaco (de manera que no refleje la luz con la que se ilumina la escena) se muestra como una buena alternativa.

Como la iluminación y el fondo son controlados las regiones de interés resultan ser las zonas brillantes de la imagen, por lo cual resulta conveniente proyectar la imagen capturada de la secuencia de video a un espacio en el que la información lumínica se trate independientemente de la información cromática. Así espacios de color adecuados podrían ser los espacios YCbCr, YUV o el Lab mencionados anteriormente **[10]**.

Por la premisa de manejar y procesar las imágenes en tiempo real se debe limitar al máximo procesos dispendiosos y se debe tener en cuenta las posibilidades más

sencillas con las que se puede abordar el problema con eficiencia. La tarjeta de evaluación hacia la cual está orientada la implementación de los algoritmos que permiten el cumplimiento de los objetivos planteados realiza una conversión de la imagen capturada de la secuencia de video al espacio YCbCr utilizando recursos netamente de hardware. Así pues, realizar un proceso de proyección a un espacio de color diferente para realizar la segmentación se torna poco práctico por cuanto se añaden algoritmos extendiendo el tiempo de procesamiento, y se obtendrán los mismos resultados bajo condiciones controladas.

Una vez determinado el espacio de color a utilizar se trabaja solamente con el canal correspondiente a la información de luminancia. Sabiendo que las zonas más brillantes para cada imagen corresponden a las regiones de color piel y que el fondo es negro opaco, se determina un umbral acorde con las observaciones hechas. En efecto, se podría utilizar un umbral lo suficientemente bajo pues el contraste entre la zona brillante y la zona opaca es evidente.

4.1.2. Algoritmo de limpieza de imagen esqueletizada

Como consecuencia de una segmentación defectuosa dadas las variaciones de iluminación, aparición de sombras y desempeño del algoritmo de esqueletización se obtienen puntos terminales espurios que limitan el proceso de reconocimiento por tratarse de puntos que no representan los dedos o el antebrazo, así mismo los nudillos resultan ser ruido para la representación. Antes de la obtención de los puntos finales se hace necesario realizar un proceso de limpieza de la imagen adelgazada resultante. Este proceso debe filtrar las ramas espurias que por lo general se encuentran cercanas al centro de masa. No obstante como el sistema debe ser robusto a cambios de escala se filtran las ramas que estén a una distancia menor que una proporción establecida a partir de la imagen de la mano. Se determinan las distancias de los puntos finales obtenidos de la imagen esquelética al centro de masa de la imagen segmentada y se establece un umbral.

En efecto, los puntos finales correspondientes a las distancias que se muestren menores que dicho umbral son descartados.

Para mantener la robustez del sistema a la distancia cámara–usuario, el proceso de limpieza de la imagen adelgazada establece un umbral que sea una proporción de la distancia más grande entre el punto del antebrazo, los puntos correspondientes a los dedos y el centro de masa. Así el umbral es adaptativo.

En este proyecto se usan únicamente los puntos finales del esqueleto de la mano, dado que las ramas en un esqueleto varían de acuerdo al algoritmo implementado y a la ubicación de la mano. Se usa el centro de masa, como punto de control para una representación robusta de la mano.

4.1.3. Algoritmos de selección de la región de interés

Una vez segmentada la imagen es conveniente diseñar una unidad con el objetivo de discriminar regiones que no son de interés para la aplicación en cuestión. Durante el desarrollo de este proyecto el único sector de interés de la imagen es donde se encuentra la mano del interlocutor, otros sectores de la escena como la cara del interlocutor resultan ser ruido en la aplicación. Para ello, fue necesario implementar una segmentación por regiones para diferenciar la mano de otras regiones (ver figura 9).

Figura 9. Marcada de regiones.

Una vez diferenciadas cada una de las regiones se escoge únicamente la región que contiene la mano, mediante el establecimiento de un área de interés **[11],** para esto se busca el mínimo rectángulo que contiene la mano. Todo el siguiente proceso se realiza exclusivamente a esta área, permitiendo obtener ventaja en cuanto al tiempo de procesamiento, evitando análisis innecesarios.

Realizar una esqueletización de la parte correspondiente a la mano del interlocutor gestual se hace conveniente para la representación de la imagen como un vector, cuyos elementos son los puntos de interés que se determinen para la etapa de reconocimiento.

4.1.4. Algoritmos de representación

Basados en un soporte teórico formal, se trata de justificar la escogencia de los puntos de control que va a ser preponderante en la intención de extraer las características topológicas de la mano a partir de una imagen adelgazada. Se quiere, a partir de una imagen esquelética, hallar los puntos de control más adecuados para la representación de la curva resultante del proceso de adelgazamiento, teniendo en cuenta que limitar su número es importante para el cumplimento de la orientación a tiempo real del procesamiento que se quiere realizar. En efecto, por cuanto el interés es el de extraer información útil de una curva en particular para el adecuado reconocimiento en tiempo real, se debe escoger puntos clave que representen de manera conveniente los cambios que la curva puede sufrir constantemente. Apoyándose en la propiedad de curvas representadas por B-Splines que determina que el resultado de cualquier sucesión de transformaciones lineales que sufra una determinada curva se ve manifestado por la misma sucesión de transformaciones lineales realizados al conjunto de puntos de control, se resalta la relativa robustez del sistema a rotaciones y/ó traslaciones que pueda sufrir la mano del interlocutor gestual. Se justifica entonces

el análisis sólo de los puntos de control para extraer características importantes de la imagen y no analizar la imagen compleja.

Haciendo alusión a las propiedades de las funciones base B-Splines, se puede destacar el hecho que el cambio de un punto de control da la idea intuitiva de un cambio en la forma de la curva representada. Así pues, es razonable escoger un punto de control que esté muy ligado a la geometría de la mano del interlocutor gestual. En efecto dicho punto puede revelar un cambio en la organización topológica de la región de interés. Un punto que se ve muy relacionado con cualquier forma generada en una región de interés es su centro de masa y se muestra de conveniente escogencia.

Sabiendo que una representación con funciones base B-Splines no pasa forzosamente por todos los puntos de control pero que se tiene la certeza de que el primer y último puntos de control son los extremos (“end points”) de la curva representada, se muestra razonable escoger precisamente los puntos definidos como puntos terminales (“end points”) en la imagen esqueletizada.

Sabiendo que el centro de masa de la mano segmentada y los puntos terminales están estrechamente ligados a la distribución topológica de las manos, se justifica la escogencia de dichos puntos de control para su posterior análisis en aras de una conveniente extracción de características de la imagen adelgazada.

El proceso de representación se basa en encontrar los dedos y el antebrazo del interlocutor y de esta forma generar un vector con los ángulos entre los dedos y el antebrazo.

Cuando la mano se encuentra completamente vertical, el ángulo del antebrazo respecto al origen (ver figura 10) es idealmente 270 grados y el ángulo entre el dedo pulgar y el antebrazo es un poco mayor a 90 grados. Cuando se rota la mano en el plano, el ángulo del antebrazo con respecto al origen cambia, tal y

como era de esperarse, sin embargo la diferencia de ángulos entre el dedo pulgar y el antebrazo sigue siendo un poco mayor a 90 grados, de la misma forma la diferencia de ángulos entre el antebrazo y cada uno de los dedos, permanece constante.

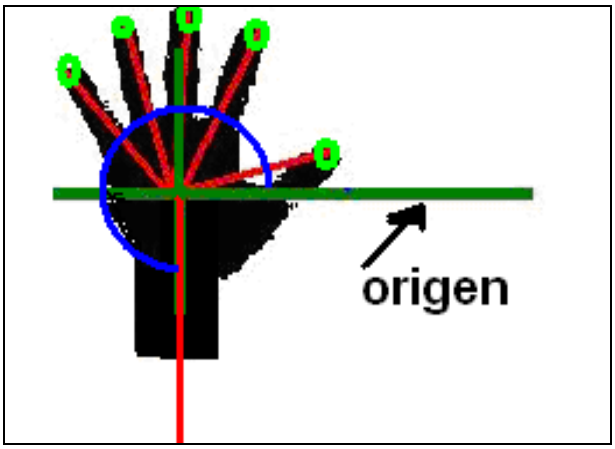


Figura 10. Determinación de puntos finales.

Se trabaja con los ángulos para que el sistema sea robusto a translaciones, acercamientos y alejamientos. Dado que los ángulos dependen de las relaciones entre longitudes, las cuales son constantes siempre y cuando el objetivo se encuentre en el plano de la cámara, es decir en planos paralelos imaginarios como se muestra en la figura 11.

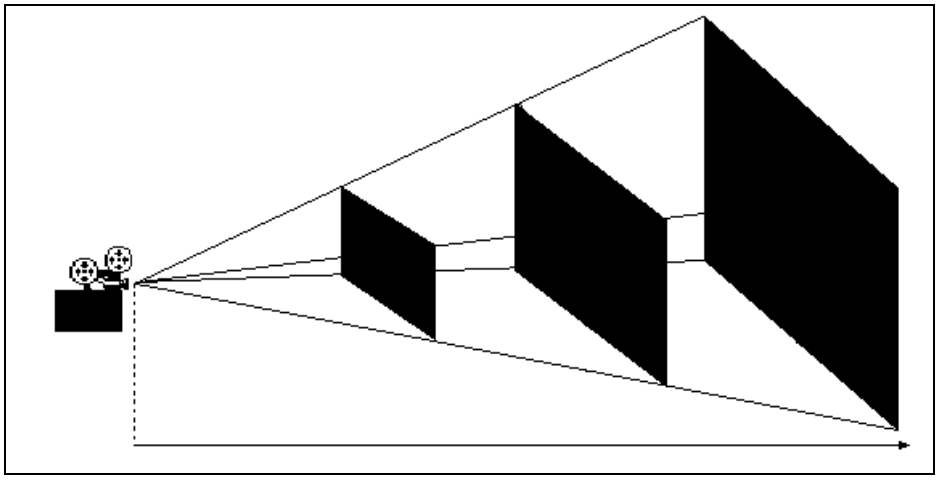


Figura 11. Planos de cámara donde debe estar la mano del interlocutor.

Mediante un ejemplo particular se muestra el proceso de obtención del vector que representa la imagen con los ángulos de los puntos finales con respecto al antebrazo(ver figura 12 y Tabla 1).

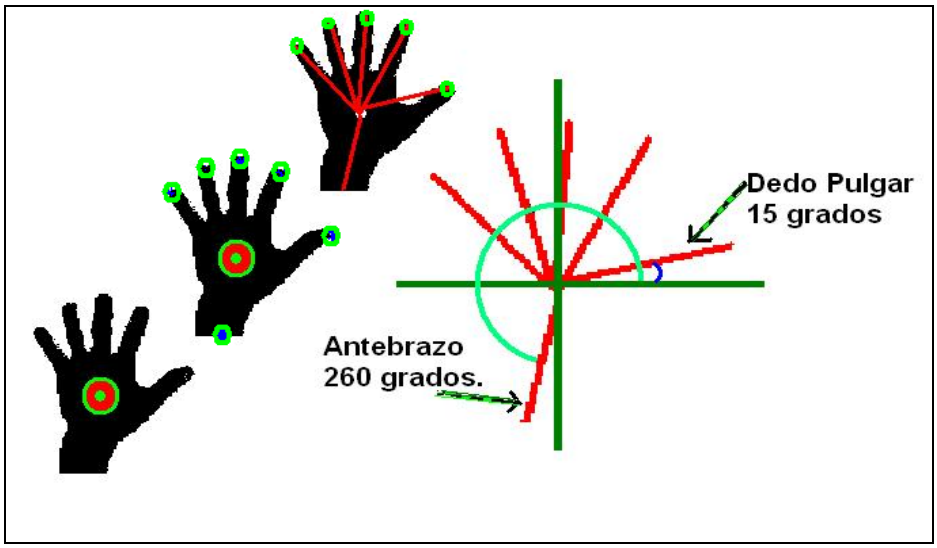


Figura 12. Determinación de puntos finales y ángulos relativos

En la tabla 1 se muestran los datos correspondientes a los ángulos de las ramas del esqueleto mostrado en la figura 15 así como los ángulos relativos al antebrazo y se evidencia el vector de características resultante que va a ser comparado con los vectores base, establecidos en la etapa de reconocimiento.

Ángulos dedos	**ángulo antebrazo**	**diferencia ángulos**
15 grados	260 grados	245 grados
60 grados	260 grados	200 grados
87 grados	260 grados	173 grados
122 grados	260 grados	138 grados
168 grados	260 grados	92 grados
Vector resultante: (92, 138,173,200,245)		

Tabla 1

De este modo, los vectores base son arreglos con dimensión igual al número de dedos en cada gesto, así la dimensión de cada uno de estos vectores es máximo cinco, en el caso de los gestos que presentan cinco dedos, sus componentes corresponden los ángulos entre el antebrazo y cada uno de los cinco dedos.

Para encontrar cada uno de los ángulos se usa como punto de referencia el centro de masa, el cual varía a medida que el interlocutor gesticula. De este modo para lograr un centro de masa más estático cabe la posibilidad de descubrir más el antebrazo del interlocutor, lo cual trae como consecuencia que los ángulos entre los dedos sean más parecidos entre si, lo que implicaría más probabilidad de error para el posterior reconocimiento. Después de una serie de pruebas se estableció que el punto óptimo era aproximadamente 4 dedos debajo de la muñeca del interlocutor.

4.1.5. Algoritmos de entrenamiento

Para cada uno de los gestos para los cuales el sistema debe responder, debe existir al menos un vector base. Los vectores base son el resultado de una recopilación y análisis estadístico de los datos obtenidos por el algoritmo que encuentra los ángulos de los puntos finales de la imagen esquelética relativos al antebrazo. En efecto, para cada gesto se toma una muestra significativa de datos de los ángulos relativos obtenidos estableciendo como ángulos base el promedio de los dichos ángulos en el caso en que la varianza de dichas medidas sea lo suficientemente pequeña.

Como primera aproximación se trabaja con imágenes obteniendo los resultados presentados en el anexo D. Para que la fase de entrenamiento sea más eficiente se opta por diseñar un algoritmo en el que se almacena en un vector de 100 posiciones los datos de los ángulos relativos correspondientes a cada cuadro capturado. Así por cada interlocutor gestual se toman 100 medidas que son

promediadas para determinar vectores base, producto de una muestra más significativa. Para cada interlocutor se toman datos cuando el eje del antebrazo forma ángulos aproximados de 45°,90° y 135° con respecto a la horizontal (100 cuadros para cada dirección) y luego se le toman medidas con movimiento (análisis en 100 cuadros).

Para esta fase de entrenamiento se toman muestras con gente heterogénea, de diferente contextura física y de intensidades de color de piel distintas. La cantidad de muestras varía de acuerdo a la varianza de los datos obtenidos. Las tablas con los resultados se muestran en el anexo E.

Una vez encontrado el vector del objeto a analizar, se calcula el error cuadrático medio con cada uno de los vectores base con la misma dimesión, posteriormente si el error cuadrático medio no excede un umbral preestablecido mediante una serie de pruebas, se decide a favor o en contra y de ser lo suficientemente parecido, se puede asignar una salida valida del sistema.

4.1.6. Algoritmos de reconocimiento

Con el propósito de reconocer en una imagen un gesto, se hace un análisis morfológico para encontrar un vector que represente el objeto, para luego compararlo mediante la distancia euclidiana con vectores que representan gestos validos, establecidos en una etapa previa de entrenamiento.

El reconocimiento se hace de acuerdo a la dimensión del vector obtenido al representar la imagen, con el algoritmo de representación desarrollado. Se definen cinco espacios vectoriales. Un espacio vectorial corresponde a los gestos con un solo dedo, otro espacio a los gestos con dos dedos, y los otros tres corresponden a los gestos de tres, cuatro y cinco dedos respectivamente.

En la figura 13 y 14 se presenta gráficamente los espacios vectoriales para los gestos representados en una dimensión y tres dimensiones. El vector verde corresponde al vector resultante de la representación de la imagen capturada, los azules son los vectores base y en rojo se representan las regiones de decisión.

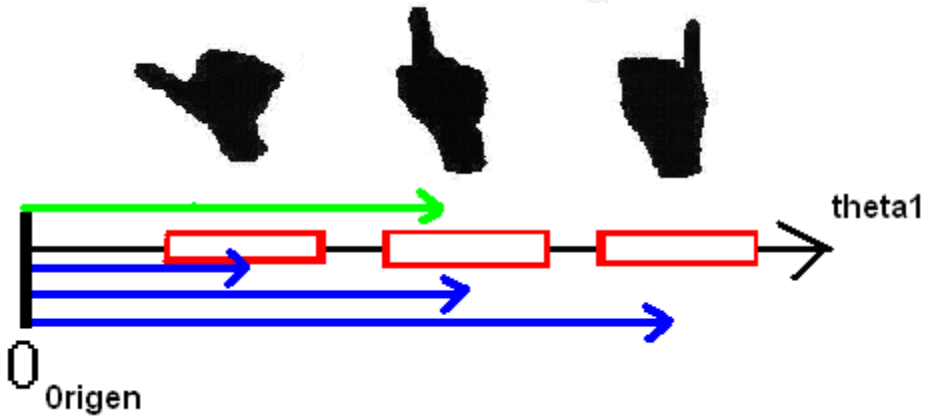


Figura 13. Regiones de decisión para vectores de 1 dimensión, 1 dedo.

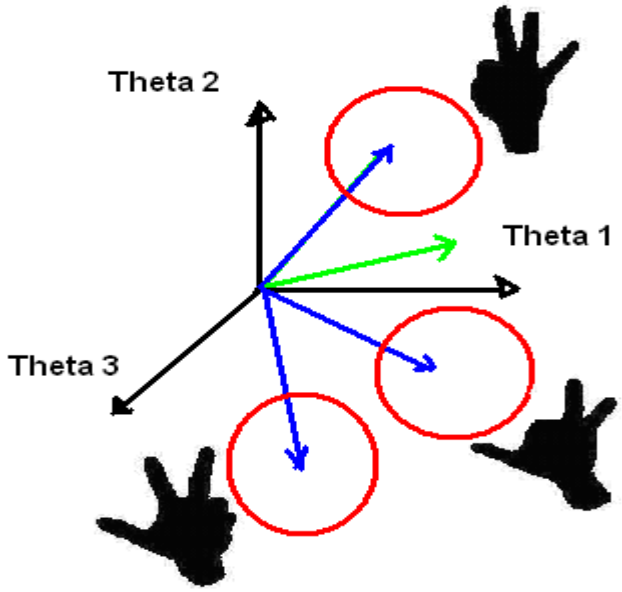


Figura 14. Regiones de decisión para vectores de 3 dimensiones, 3 dedos.

4.2. Desarrollo del software

Para la captura de la secuencia de video se deben ajustar ciertos parámetros de captura de la cámara**.** Se determina el tamaño de cada cuadro capturado, además del formato de compresión utilizado, en este cado es MPEG-2[7] y la cantidad de cuadros a procesar por segundo **[9]**. Para determinar estas propiedades de captura se hace uso de la librería gráfica Highgui que está incluida en la librería de OpenCV y reúne diversas funciones miembro apropiadas para el tratamiento de imágenes y video.

El proceso de diferenciación de objetos se realiza de forma iterativa a cada una de las imágenes analizadas, para posteriormente seleccionar el objeto de interés. Se implementaron los dos algoritmos de esqueletización presentados anteriormente, no obstante la diferencia de tiempo entre los dos procesos fue de unos pocos milisegundos, sin embargo el algoritmo Shang Zuen conserva mejor la estructura morfológica de la mano, indispensable para un posterior reconocimiento.

[7] mpeg-2: estándar de compresión de video, siglas de Motion Photography Environment Group.

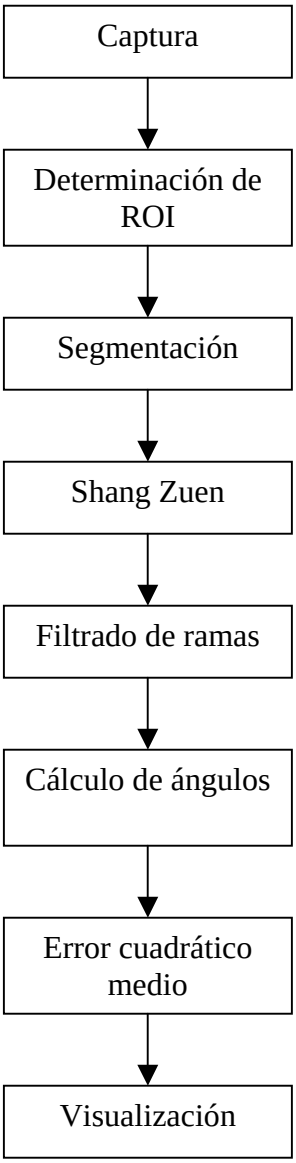


Figura 15. Diagrama en bloques del proceso en la computadora personal

Se desarrolla una interfaz gráfica haciendo uso de las bibliotecas de OpenCV, buscando una amigable presentación del sistema, que permita visualizar en tiempo real la evolución del proceso, así como la representación del gesto realizado. Para el desarrollo de la interfaz grafica y el correcto funcionamiento de los recursos a utilizar (ventanas, marco para la visualización de imágenes, cuadros de texto, etc), se requiere una adecuada conexión e hilación de los controles y

ventanas con código personalizado, este código corresponde a funciones que genéricamente reciben el nombre de manejadores o manipuladores.

4.3. Desarrollo del hardware

El algoritmo de adelgazamiento implementado en el procesador es el MAT (Medial Axis Transform[8]) debido a que el tiempo de proceso, aproximadamente 35 ms, resultó ser 5 a 6 veces menor que el tiempo logrado con el algoritmo de adelgazamiento Zhang Suen, de 150 ms, el cual varía de acuerdo a la geometría y tamaño de los objetos de la imagen. El hecho de trabajar con los ángulos relativos al antebrazo, le da al sistema la capacidad de ser robusto a rotaciones, sin embargo, dado que el algoritmo de adelgazamiento de eje medio no conserva exactamente la estructura topológica de la mano, las rotaciones están limitadas a aproximadamente 45 grados con respecto al origen.

Para diferenciar los objetos se uso un algoritmo de marcado no iterativo, el cual es menos eficiente que su versión recursiva debido a la cantidad de recorridas adicionales de la imagen, para realizar el proceso de marcado iterativo es indispensable hacer uso de buffer circulares para evitar que se sature la pila del stack pointer del procesador.

El desarrollo en el hardware se muestra sintetizado en el diagrama en bloques representado en la figura 16:

[8] Transformada de eje medio

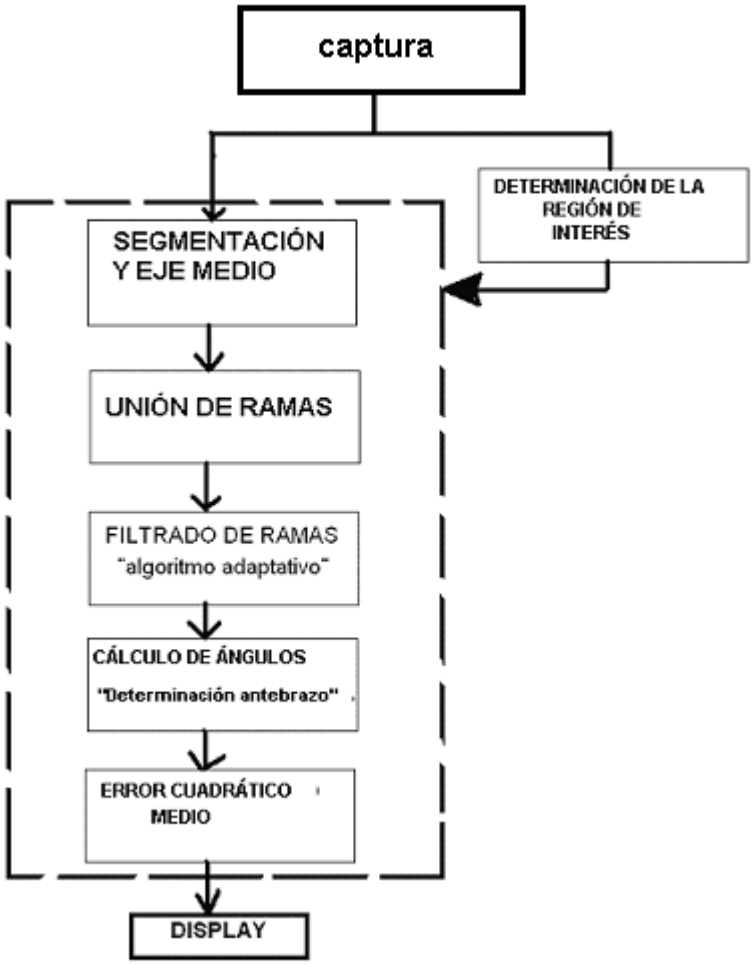
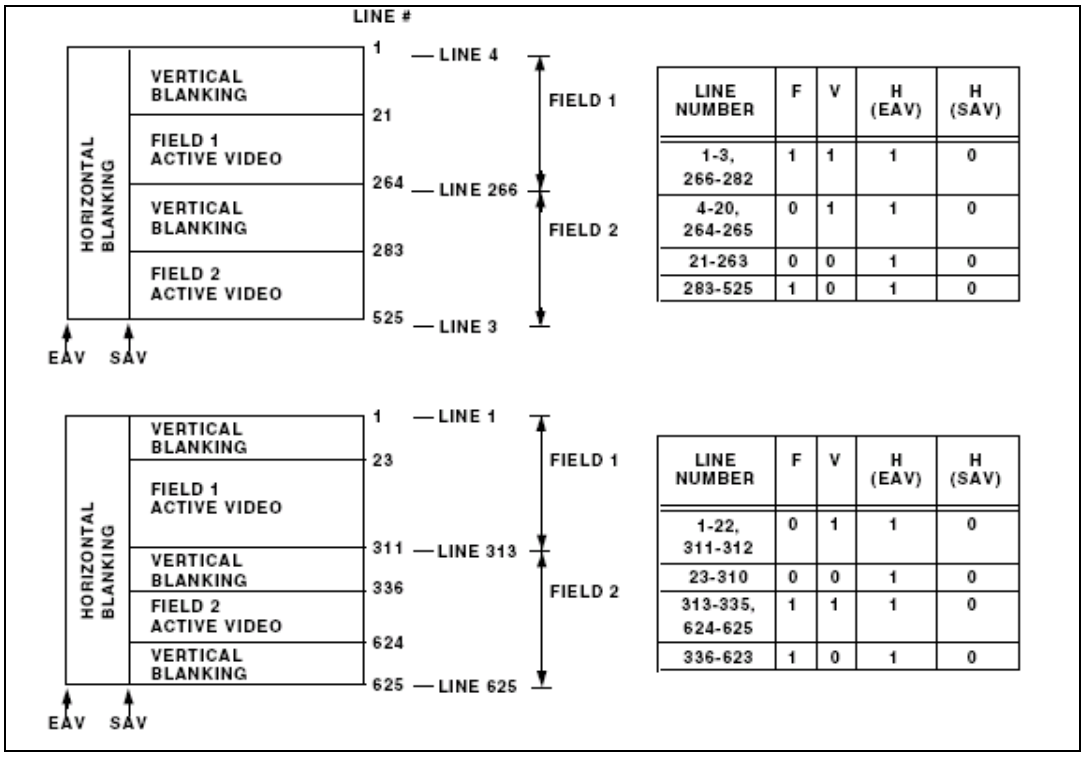


Figura 16. Diagrama en bloques del proceso en la DSP

En el desarrollo del sistema se usan principalmente el codificador de vídeo ADV7183, la interfase paralela de periféricos (PPI), el controlador de DMA y la memoria asíncrona SDRAM. Se descarta el uso de la memoria L1 para almacenar la imagen, porque el tiempo de proceso usando la memoria L2 resulta ser suficiente para la orientación del proyecto al tiempo real, dando la posibilidad de configurar la memoria L1 como memoria cache.

Para el manejo de las imágenes el sistema de desarrollo cuenta con un ADC el cual recibe una señal análoga de vídeo NSTC o PAL (cuyas componentes se

muestran en la figura 17) a través de un conector RCA macho, este, devuelve una señal digital con información de imagen y blanqueo(sincronismo).



LINE NUMBER	F	V	H (EAV)	H (SAV)
1-3, 266-282	1	1	1	0
4-20, 264-265	0	1	1	0
21-263	0	0	1	0
283-525	1	0	1	0

LINE NUMBER	F	V	H (EAV)	H (SAV)
1-22, 311-312	0	1	1	0
23-310	0	0	1	0
313-335, 624-625	1	1	1	0
336-623	1	0	1	0

Figura 17. Estándar de video NTSC[9]

El codificador de video ADV7183 entrega una trama YCbCr 4:2:2, con la siguiente secuencia: blanqueo Cb Y Cr Y Cb Cr…. Cb Y Cr Y blanqueo (como se puede observar en la figura 16), un total de 1440 píxeles por fila, correspondientes a 720 de luminancia, 360 de crominancia azul (Cb = Y-B) y 360 de crominancia roja (Cr= Y- R).

[9] Extraída del HARDWARE REFERENCE MANUAL ADSP BF 533 Blackfin Processor (Pág. 11-15).

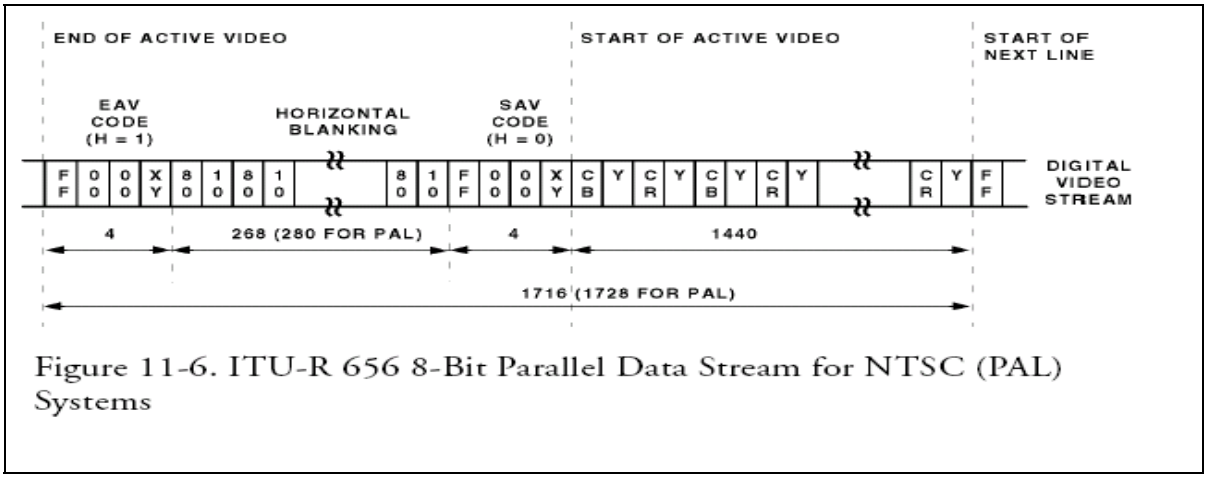


Figura 18. Trama de video[10]

El PPI es un puerto bidireccional half-dúplex[11] que se puede configurar para transferir datos a una tasa de 16 bits cada ciclo de reloj del sistema, de estos 16 bits tan sólo 4 son dedicados y 12 son banderas programables configuradas como datos del PPI.

A través del PPI se filtra la información de blanqueo (sincronismo), la cual no es necesaria para este caso, resulta ser únicamente desperdicio de memoria y tiempo. De esta forma se transmite al DMA únicamente el video activo.

También a través del PPI se filtra la información de crominancia (Cb & Cr), pues en el proyecto se usa únicamente la luminancia, dado que el fondo y la iluminación se encuentran controlados.

El PPI junto al DMA permiten implementar un sub-muestreo de la imagen exclusivamente con hardware, con lo que se optimiza la velocidad de proceso al doble, ya que se elimina la mitad de los accesos a memoria. Como consecuencia cuando se sub-muestrea una imagen, aunque se pierde resolución, no se afecta de manera significativa el rendimiento de la aplicación, dado que las imágenes a

[10] Extraída del HARDWARE REFERENCE MANUAL ADSP BF 533 Blackfin Processor (Pág. 11-14).

[11] Permite entrar o salir con video pero no simultáneamente.

analizar son partes del cuerpo que no tienen muchos detalles que resulten indispensables en el momento de su análisis.

Una vez filtrada (blanqueo, crominancia) y sub-muestreada la imagen (todo esto mediante hardware), se tiene en memoria un total de 360 por 250 píxeles, correspondientes a las 360 filas por 250 columnas de la imagen almacenados de forma consecutiva en la posición de memoria para la cual se configura previamente el DMA (ver figura 19). La información de luminancia tiene valores entre 0 y 255, por tanto se almacena en memoria en formato unsigned char (8 bits por píxel), es decir se almacenan 4 píxeles en una posición de memoria (la memoria SDRAM es de 32 bits).

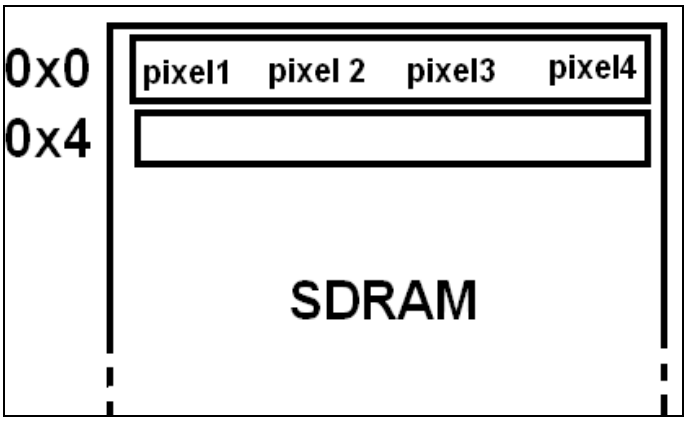


Figura 19. Cuatro píxeles en cada posición de memoria.

Dado que el DMA usa un contador de 16 bits (2 ^16=64 K) para direccionar la memoria, lo cual es insuficiente para el tamaño de la imagen (360*250 = 90K), es necesario usar DMA bidimensional (DMA2D), el cual da una capacidad de direccionamiento de 64K * 64 K.

Se configura el DMA para que genere una interrupción una vez la imagen completa haya sido almacenada en memoria, y deje de transferir datos hasta que

sea habilitada nuevamente. Mientras el DMA realiza la transferencia de la imagen, el procesador espera, y una vez la imagen esta completa en memoria el procesador analiza la imagen y el DMA se deshabilita.

Dentro de la rutina de la interrupción del DMA se deshabilita el PPI y posteriormente se procesa la imagen. Una vez analizada la imagen se procede a habilitar el DMA y el PPI, para que el DMA transfiera a memoria otra imagen proveniente del PPI y se repita el proceso. Se debe tener especial cuidado de habilitar primero el canal del DMA antes de habilitar el periférico y de deshabilitar el periférico antes de deshabilitar el canal del DMA, para evitar un conflicto de buses, dada la arquitectura de la tarjeta. Para reiniciar el DMA se debe modificar el registro de configuración del DMA (DMA_X_CONFIG), con lo que se reinicia la transferencia desde el ADC a memoria, y una vez almacenada una imagen ejecutar la rutina de interrupción, que corresponde al proceso de la imagen **[5]**.

La aplicación consta de dos partes una parte de reconocimiento y otra parte de entrenamiento. La parte de entrenamiento se hace para encontrar la mano en la imagen, para un posterior reconocimiento. Cuando un interlocutor gesticula no cambia constantemente de posición, razón por la cual no es necesario buscar constantemente la mano sino que una vez encontrada se puede fijar una región de interés, y usar esta durante un determinado tiempo. En el hardware dedicado se hace un entrenamiento cada doscientos reconocimientos. La razón por la cual no resulta viable estar fijando la roi es que este es demasiado costoso en cuanto a tiempo de proceso, mientras el reconocimiento se dura en el peor de los caso 150 ms, la región de interés se halla en aproximadamente un segundo (período durante el cual no se reconocen gestos válidos).

La presentación escogida para demostrar el correcto reconocimiento fue realizada a través de un display siete segmentos. Este se controla usando las banderas programables del procesador. Así, cada vez que se reconoce un gesto se habilitan y decodifican de forma apropiada para su representación.

5. ANÁLISIS DE RESULTADOS

En este capítulo se presentan las diferentes pruebas realizadas al sistema al igual que los resultados obtenidos. Las pruebas están orientadas a medir la confiabilidad del reconocimiento y su ejecución en tiempo real.

Se realizan las siguientes pruebas:

- Velocidad de procesamiento
- Pruebas de segmentación
- Pruebas de reconocimiento del sistema

5.1. Velocidad de procesamiento

Este factor es fundamental para los propósitos del proyecto, su ejecución en tiempo real permite procesar continuamente el video sin discriminar gran cantidad de tramas, las cuales pueden tener información significativa de la escena. En esta prueba se calcula el tiempo que tarda el sistema en capturar, procesar y representar el gesto realizado por el interlocutor.

Para determinar el tiempo de procesamiento en Visual C++, se hace uso de la herramienta de perfiles (Profile) integrada en el entorno de programación en cuestión. Se realiza el estudio del tiempo tomado por cada función del proceso a lo largo del análisis de 1000 cuadros de la trama de video. Se obtiene información de las funciones más costosas computacionalmente y el tiempo total empleado para el procesamiento de los 1000 cuadros capturados (ver anexo F). Se obtiene un tiempo total de procesamiento de las 1000 imágenes capturadas de 85,28 segundos, es decir un tiempo promedio por cuadro de 85.28 ms.

En la tarjeta de desarrollo Ez-Kit Lite BlackFin 533 se tienen a disposición dieciséis banderas programables que pueden ser configuradas como salidas o entradas. Dado que el PPI usa 8 de las banderas programables para transferir datos, únicamente se dispone de 8 banderas para ser configuradas como salidas. Siete de estas banderas se utilizan para manejar directamente el display, la bandera que sobra se asigna para medir el tiempo del proceso en un osciloscopio, de tal forma que cada vez que se realiza el proceso, se invierte esta bandera, así la señal obtenida es una señal cuadrada con periodo igual a la mitad del tiempo de procesamiento. Las mediciones se realizan por medio del osciloscopio Tektronix TDS-3012 (ver figura 20).

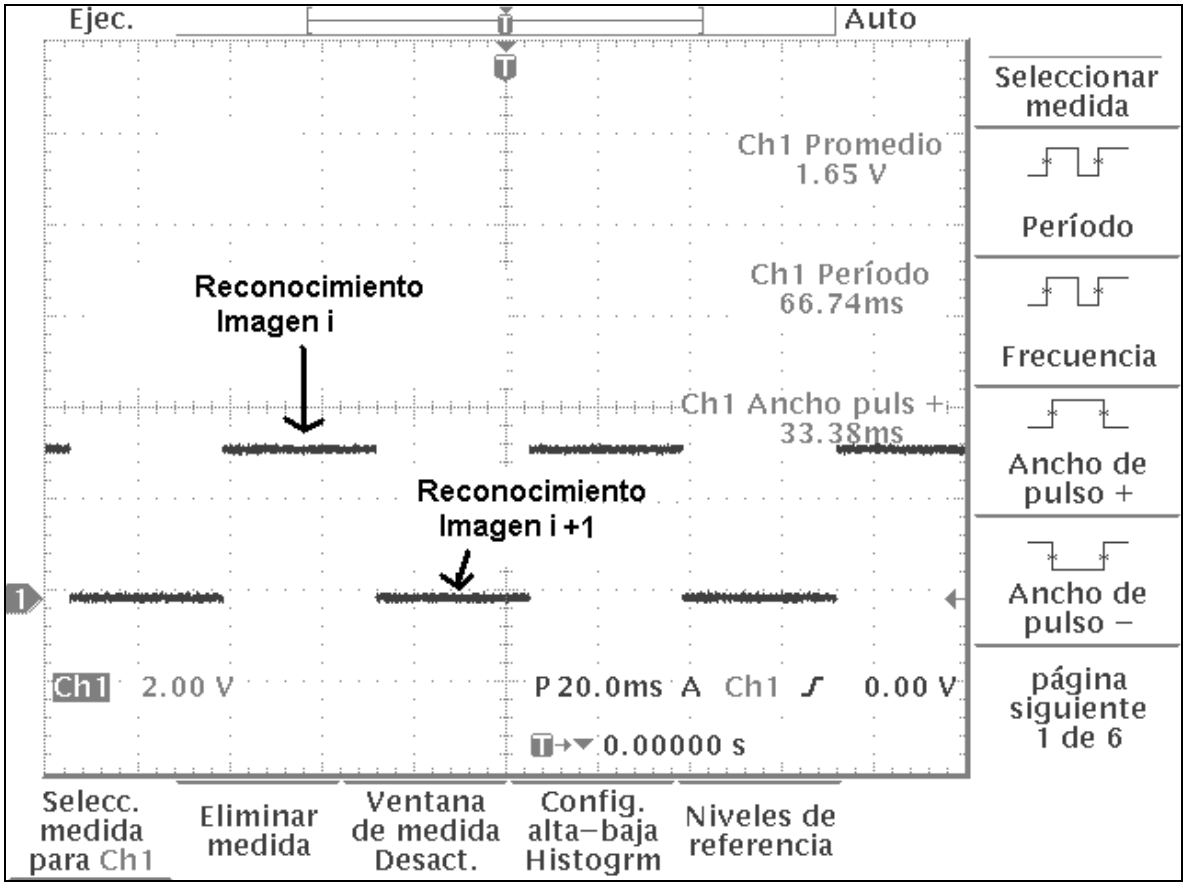


Figura 20. Tiempo de proceso en cada imagen[12]

[12] Extraída del osciloscopio Tektronix TDS-3012

Cada periodo corresponde al procesamiento de dos imágenes, siendo aproximadamente 33 ms el tiempo para procesar cada imagen.

Cada cien veces se realiza un entrenamiento de la región de interés (ROI[13]), se puede apreciar claramente en la figura 21 que el proceso de entrenamiento resulta ser computacionalmente más costoso que el proceso de reconocimiento.

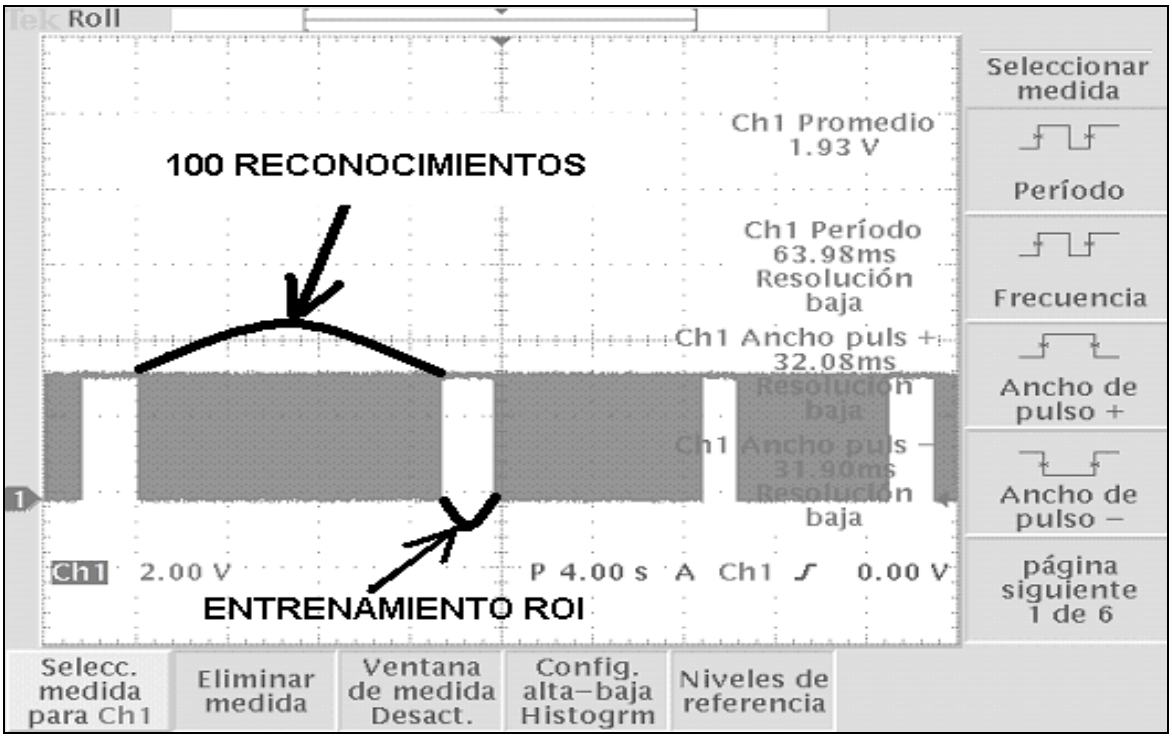


Figura 21. tiempo reconocimiento y de entrenamiento de la región de interés[14]

El tiempo de reconocimiento varía de acuerdo con la imagen que se procese y la región de interés pre-establecida. El algoritmo empleado consta de una etapa de entrenamiento para la fijación de la región de interés, entrenamiento de duración aproximada 1.5 segundos, esta se actualiza cada 100 procesos como se evidencia

[13] ROI: sigla proveniente del idioma inglés ” Region Of Interest”
[14] Extraída del osciloscopio Tektronix TDS-3012.

en la figura 22. Así se puede establecer que para el procesamiento de 100 imágenes se emplea un tiempo de 100*33ms+1.5s = 4.8 segundos; lo que corresponde a un tiempo promedio de proceso de 48ms por imagen.

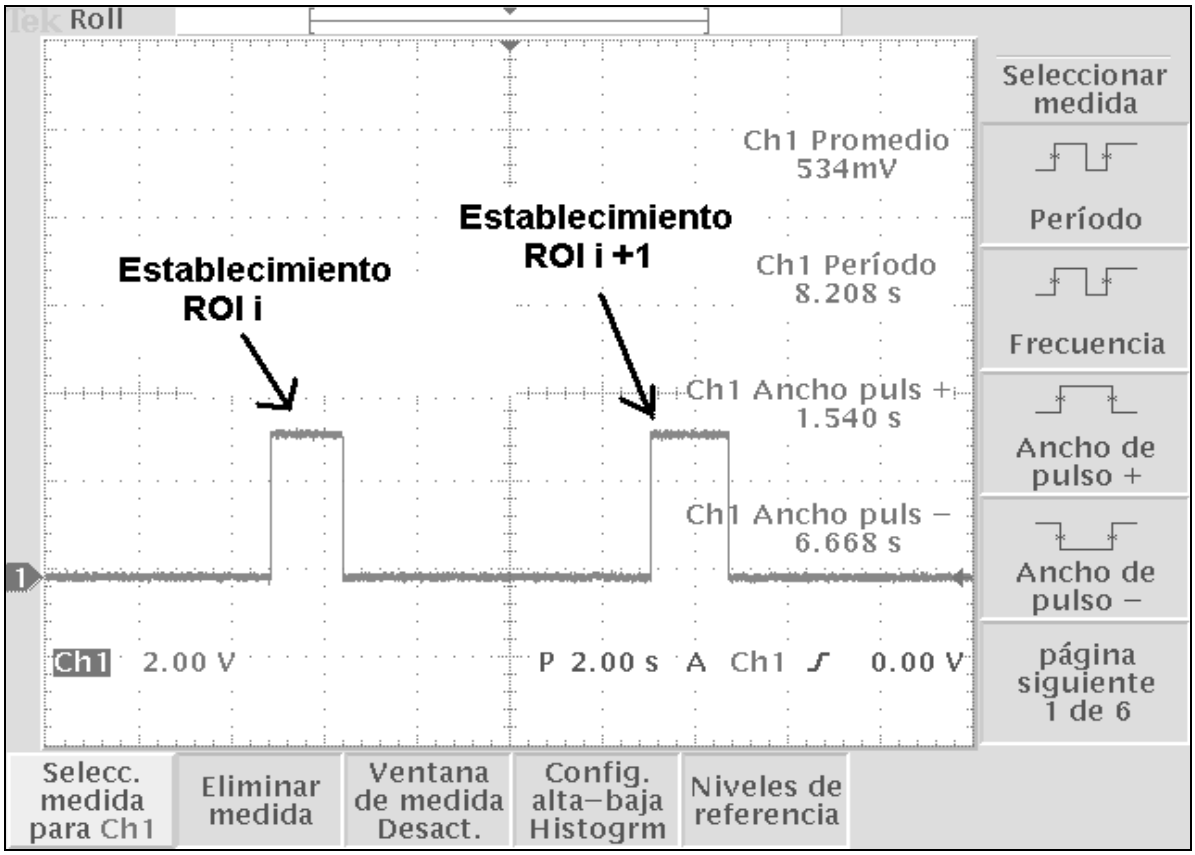


Figura 22. Duración de la etapa de establecimiento de la región de interés[15]

Una propuesta para mejorar el desempeño en el tiempo del sistema es ubicar la mano, y luego hacer uso de un algoritmo de seguimiento, para evitar buscar constantemente en toda la imagen. En la naturaleza, los seres vivos lo hacen de forma análoga: primero se identifica el objetivo para un posterior reconocimiento.

El tiempo de proceso es un poco mayor en el computador personal pues en el entorno de programación Visual C++ se implementan algoritmos más costosos

[15] Extraída del osciloscopio Tektronix TDS-3012

computacionalmente. En efecto, se trabaja con el proceso iterativo Zhang Suen para adelgazar la imagen y no tomando como imagen esquelética el resultado de tomar sólo el eje medio de la región de interés. Además, en el procesador dedicado, se marca una vez cada cien procesos (por el extensivo costo que implica la función recursiva utilizada en el proceso de marcado) mientras que en el entorno de programación se marca siempre que un cuadro de la secuencia de video va a ser analizado.

5.2. Pruebas de segmentación

Esta etapa es realmente crítica en el proceso de la imagen, una mala segmentación puede hacer equivocar fácilmente al sistema.

Para una buena segmentación es conveniente tener las mejores condiciones de iluminación. Se realizan pruebas con diferentes tipos de iluminación y los resultados de la etapa de segmentación son los mostrados en las figuras 23,25 y 27[16] y los de la etapa de esqueletización en las figuras 24,26 y 28.

- **Iluminación deficiente**

Se refiere con iluminación deficiente a la que no está totalmente direccionada a la mano de la persona que gesticula, generando sombras indeseadas y heterogeneizando la región de la mano lo que resulta en segmentaciones (figura23) y adelgazamientos (figura 24) defectuosos.

[16] Las imágenes mostradas en estas figuras son extraídas de la herramienta del entorno de programación de la tarjeta de desarrollo DSP++ que permite ver imágenes resultantes de procesamientos intermedios efectuados por el sistema implementado.

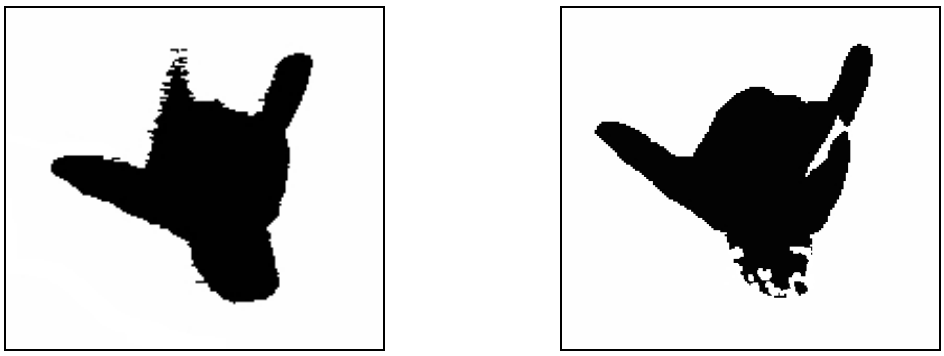


Figura 23. Segmentación resultante con iluminación deficiente.

Como se puede observar, con este tipo de iluminación la segmentación obtenida no es homogénea, existen sectores de la mano donde la región de interés se divide y por ende el siguiente proceso se realiza erróneamente. Estos sectores se generan por la aparición de sombras no deseadas afectando también el proceso de esqueletización de la región de interés (figura 24).

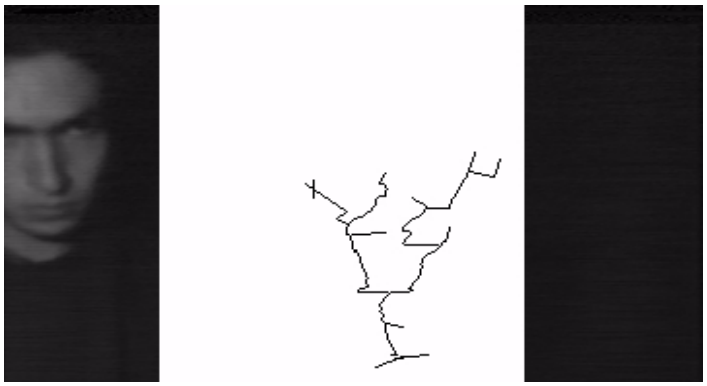


Figura 24. Esqueletización resultante con iluminación deficiente.

- **Iluminación aceptable**

Como iluminación aceptable se entiende la iluminación que se direcciona a la mano aunque por su baja intensidad no permite la discriminación clara del límite donde acaba el antebrazo **[3]**.

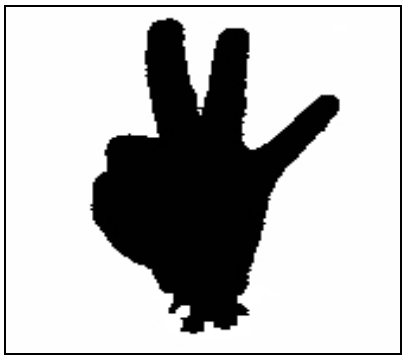


Figura 25. Segmentación resultante con iluminación aceptable.

Con esta iluminación se logra obtener un mejor ambiente de trabajo, la región de interés ya no aparece dividida. Sin embargo existen problemas en el punto límite del antebrazo, el corte no aparece bien definido, lo que causa errores al tomar este punto. Como el módulo de reconocimiento toma como referencia el punto final para obtener los resultados, este error es realmente grave para el efectivo reconocimiento del gesto (ver figura 26).

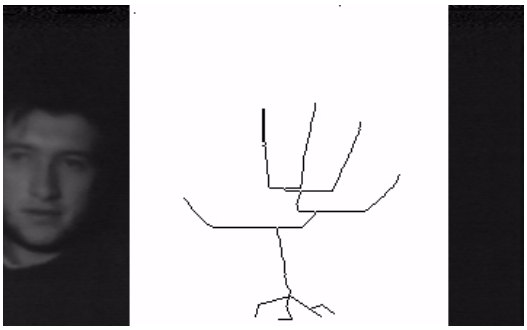


Figura 26. Esqueletización resultante con iluminación aceptable.

- **Iluminación ideal**

Como iluminación ideal se entiende la iluminación cuya intensidad permite resaltar la mano de manera homogénea. La luz se encuentra totalmente direccionada a la mano que es alumbrada de manera uniforme obteniendo como resultado de la segmentación la imagen mostrada la figura 27.

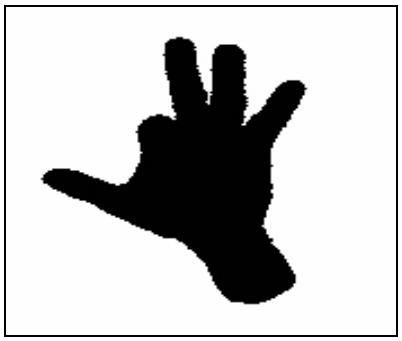


Figura 27. Segmentación resultante con iluminación ideal.

Con este tipo de iluminación, totalmente direccional, la segmentación obtenida es la adecuada para el posterior procesamiento (ver figura 28). Las regiones aparecen homogéneas y bien delimitadas **[3]**.

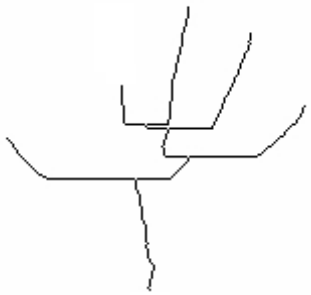


Figura 28. Esqueletización resultante con iluminación ideal.

Otro problema para el proceso de segmentación es originado por la persistencia de la cámara, al realizar los gestos se genera una estela que puede ocasionar errores. La persistencia es inherente a la cámara, en pruebas realizadas con cámaras de alta tecnología se minimiza este problema, sin embargo con cámaras antiguas el problema es más notorio. En la figura 29[17] se puede observar este problema.

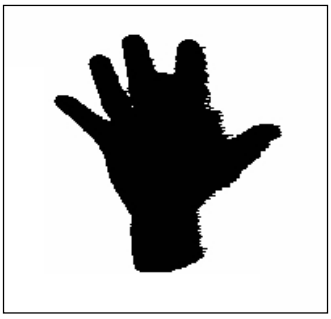


Figura 29. Imagen resultante de su proceso de segmentación

5.3. Pruebas de reconocimiento

Una vez terminado el proyecto se establecen una serie de medidas que informan acerca del desempeño del sistema implementado tanto en el entorno de programación Visual C++ como en el Ez Kit Lite Blackfin 533 de Analog Devices.

Se tratan de medidas en las que se busca vislumbrar la capacidad del sistema de reconocer los gestos del alfabeto para el cual ha sido entrenado.

Para obtener resultados significativos se automatiza el proceso de medición implementando algoritmos en los que se hacen las medidas analizando 100 cuadros por usuario correspondientes a 100 imágenes diferentes. Para establecer las medidas se realizan análisis estadísticos entre el número de cuadros

[17] La imagen mostrada en esta figura es extraída de la herramienta del entorno de programación de la tarjeta de desarrollo DSP++ que permite ver imágenes resultantes de procesamientos intermedios efectuados por el sistema implementado.

capturados y los resultados obtenidos, este proceso se realiza para un gesto y un usuario particular. Luego se promedian los resultados obtenidos con varias personas.

Las pruebas realizadas se dividen en las siguientes categorías:

1) Verdaderos aciertos
2) Falsos aciertos
3) Verdaderos Rechazos
4) Falsos Rechazos

Para establecer el índice de verdaderos aciertos se cuenta el número de cuadros en que se reconoce dicho gesto y se establece su proporción con el número de cuadros analizados.

Para establecer el índice de falsos aciertos, el algoritmo cuenta para cada gesto y para cada usuario el número de veces en que se reconoce como gesto válido uno diferente al que se encuentra gesticulando y determina su proporción con el número de cuadros analizados.
Las pruebas de falsos rechazos consisten en realizar gestos aleatorios del alfabeto, excluyendo el gesto en estudio y medir cuántas veces se reconoce el gesto.

Finalmente se promedian los resultados arrojados para los diferentes usuarios y gestos. Los resultados obtenidos se muestran en la tabla 2. En ella se presentan los datos correspondientes a 100 muestras para cada individuo y la letra asociada, si se suman horizontalmente los resultados de verdaderos aciertos, falsos aciertos, falsos rechazos se obtiene las 100 muestras.

	VERDADEROS ACIERTOS	FALSOS ACIERTOS	FALSOS RECHAZOS
Letra E	97	0	3
	95	0	5
	92	0	8
	81	0	19
Letra G	90	0	10
	70	0	30
	82	0	18
	79	0	21
Letra B	75	0	25
	72	0	28
	70	0	30
	67	0	33
Letra P	89	0	11
	75	0	25
	70	0	30
	68	0	32
letra F	61	0	39
	66	0	34
	68	0	32
	70	0	30
Letra S	60	0	40
	70	0	30
	50	0	50
	23	0	77
Letra U	66	0	34
	60	0	40
	97	0	3
	63	0	37
Letra H	67	0	33
	81	0	19
	58	1	41
	48	2	50
Letra C	45	1	54
	52	0	48
	58	0	42
	69	0	31
Letra L	47	0	53
	55	0	45
	60	0	40
	62	2	36
Letra i	71	0	29
	90	0	10
	80	0	20
	62	0	38
Letra O	72	0	28
	68	0	32
	64	0	36
	60	0	40
Letra N	68	0	32
	66	2	0
	77	0	23
	54	3	43
Letra A	63	0	37
	68	0	32
	50	2	48
	70	0	30

Tabla 2.

A partir de la tabla 2 se puede observar que los gestos con menor número de dedos tienen mayor probabilidad de error porque son caracterizados con un vector

de menor dimensión que el correspondiente a gestos con mayor número de dedos. Debido a esto, en aplicaciones prácticas de interacción hombre-máquina, las órdenes de mayor importancia deben corresponder en lo posible a gestos con el mayor número de dedos.

Los falsos aciertos evidenciados para las letras A, N, C, L y H son provocados por el límite fijado en el proceso a la distancia entre el vector de ángulos relativos al antebrazo de la prueba y los vectores base establecidos en la fase de entrenamiento. En efecto, para que un gesto sea válido la distancia entre el vector prueba y el vector base correspondiente a dicho gesto no debe sobrepasar un umbral. Al aumentar dicho umbral se aumenta el número de falsos aciertos y también el de verdaderos aciertos. Se opta entonces por determinar el umbral de manera que se tenga un compromiso entre aceptable proporción de verdaderos aciertos y bajo número de falsos aciertos.

Los verdaderos rechazos buscan medir la selectividad de los algoritmos, se trata de comprobar que no reconozca un gesto cuando en realidad se está gesticulando otro. Los resultados obtenidos se muestran la tabla 3.

	VERDADEROS RECHAZOS
Letra E	100
	100
	100
	100
Letra G	100
	100
	100
	100
Letra B	100
	100
	100
	100
Letra P	100
	100
	100
	100
letra F	100
	100
	100
	100
Letra S	100
	100
	100
	100
Letra U	98
	100
	100
	99
Letra H	98
	100
	100
	100
Letra C	100
	99
	100
	99
Letra L	98
	99
	100
	94
Letra i	100
	100
	100
	100
Letra O	100
	100
	100
	100
Letra N	100
	100
	100
	100
Letra A	96
	99
	100
	98

Tabla 3

De la tabla 3 se observa que el desempeño del sistema es bastante bueno por cuanto muy pocas veces muestra un gesto diferente al que se está realizando. Las letras L y A presentan mayor margen de error dado que los nudillos de la mano son tomados erróneamente como un dedo elongado y por lo tanto la A queda idéntica a la L. Las letras U y H se confunden cuando el interlocutor abre mucho los dedos realizando el gesto correspondiente a la U obteniendo los mismos ángulos que los que se encuentran gesticulando la letra H. Lo mismo ocurre cuando se abren mucho los dedos índice y meñique propios del gesto correspondiente a la letra H haciendo que el sistema se confunda con el gesto de la letra C.

Para un correcto desempeño del módulo de reconocimiento de patrones es indispensable minimizar ruido presente en el esqueleto de la imagen, este aparece con la generación de ramas espurias las cuales pueden cambiar completamente la forma del objeto en estudio. Debido a que la mayoría de las ramas aparecen alrededor del centro de masa del objeto en estudio, en este caso la mano del interlocutor, estas son filtradas usando proporciones de longitud de la mano ya que esta técnica lo hace robusto a cambios de escala.

Abordando un campo experimental con la meta de cumplir con los objetivos planteados se llega a unos resultados motivantes para proseguir con la investigación. Siempre teniendo en cuenta el compromiso entre robustez de los algoritmos y costo computacional que implican, se consigue la implementación en tiempo real de un proceso abierto a futuras mejoras y orientado a diversas aplicaciones dentro de las cuales cabe resaltar el aspecto social. En efecto, se trata de una interfaz hombre máquina que abre la posibilidad a la comunidad sordo muda de interactuar con comunidades ajenas a su lenguaje de señas.

6. CONCLUSIONES

- Resulta inapropiado hacer una comparación directa de funcionamiento entre el sistema implementado en la computadora personal y el procesador embebido por las diferencias de procesamiento que presentan sus respectivos algoritmos. Se puede recalcar el hecho que por realizar la diferenciación de todos los objetos presentes en la escena en cada cuadro de la secuencia de video, el algoritmo en el computador personal está más orientado a descontrolar el fondo.

- Bajo condiciones de fondo e iluminación controlada, el espacio más apropiado para segmentar la mano es el canal de luminancia, no se justifica implementar una transformación de espacio, porque implica mas computo y se obtendrán los mismos resultados.

- Para obtener una adecuada segmentación de las imágenes, es fundamental generar condiciones de iluminación óptimas, permitiendo simplificar el posterior procesamiento y la robustez de los algoritmos.

- Trabajar con el espacio de color YCbCr resulta una opción óptima pues el convertidor análogo digital de la tarjeta de desarrollo BF 533 de Analog Devices maneja el estándar YCbCr 4:2:2, por lo que ninguna transformación de espacio es requerida.

- Para la orientación en tiempo real del proyecto es más conveniente trabajar únicamente con la luminancia debido a la menor cantidad de información a procesar.

- La técnica de adelgazamiento de transformación del eje medio es computacionalmente más eficiente con respecto a las técnicas iterativas, viéndose atractiva la posibilidad de implementar esta técnica en dispositivos con menor capacidad de cómputo como lo es un procesador digital de señales.

- El establecimiento de áreas de interés en la imagen en estudio, elimina la posibilidad de realizar análisis innecesarios y mejorar sustancialmente los tiempos de procesamiento.

- El sistema desarrollado está basado en un modulo de entrenamiento flexible, el cual da la oportunidad de ampliar fácilmente el lenguaje propuesto. El número de gestos está determinado por la necesidad final de la aplicación. Agregar más gestos al alfabeto puede incrementar el número de falsos aciertos del sistema.

- El alfabeto implementado resulta ser bastante viable para implementar en prototipos de hardware embebidos, dando la posibilidad de desarrollar un dispositivo portátil.

- En casos particulares por fisiología de la mano de diferentes personas, algunos gestos resultan tener una alta varianza con respecto a los vectores base de entrenamiento establecidos, sin embargo surge la posibilidad de re-entrenar el sistema para establecer nuevos vectores base y regiones de decisión.

- El algoritmo de representación de los gestos es flexible a futuras ampliaciones del alfabeto.

- Los gestos que presentan mayor número de dedos resultan ser mejor reconocidos porque presentan mayor dimensión al ser representados con el algoritmo propuesto.

- Como futura mejora se podría hacer un pre-procesamiento adicional de la imagen mediante transformaciones estructurales de erosión y dilataciones, con lo que se lograría reducir el efecto del ruido en la aplicación,

7. BIBLIOGRAFÍA

[1] V.S. Nalga. A Guided Tour to Computer Vision. Addison Wesley, 1993.

[2] De La Escalera Hueso. Visión por Computador, fundamentos y métodos. Madrid, España: Prentice Hall, 2001.

[3] FORSYTH, David. Computer vision: A modern approach. New Jersey, Estados Unidos: Prentice Hall, 2003.

[4] GONZALEZ Rafael. Digital Image Processing. Addison Wesley Publishing Company, 1993

[5] D. Heckenberg y B. C. Lovell, "MIMIE: A Gesture-Driven Computation Interface", Proceedings of Visual Communications and Image Processing, SPIE, V 4067, PP 261-268- Perth 20-23 Junio, 2000.

[6] HARDWARE REFERENCE MANUAL ADSP BF 533 Blackfin Processor.

[7] Cevallos, Francisco Javier. Microsoft Visual C++. Aplicaciones para Win32. Editorial ra-ma. Madrid, 1998.

[8] Intel® Software Products Open Source " Intel Open Source Computer Vision Library Reference Manual" Diciembre 2004, Disponible en la direccion electronica http://www.intel.com/research/mrl/research/opencv.

[9] "Diferential video coding of FACE and gesture events in presentation videos" Robin Tan, James W. Davis Computer Vision Laboratory, Department of Computer Science and Engineering, Ohio State University, USA Received 14 March 2002; accepted 2 February 2004 Available online 7 August 2004J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[10] Haralick, Robert M. Computer and Robot Vision V.2. Addison – Wesley,1993.

[11]Parker, Jim R., Practical Computer Vision Using C. Nueva York, Estados Unidos 1994

8. ANEXOS

ANEXO A: OPENCV

OpenCV (Open Source Computer Vision Library), es una biblioteca desarrollada por Intel. Es una colección de funciones en C y C++ que implementan algunos de los algoritmos más populares en el procesamiento digital de imágenes.
Fue originalmente desarrollada para proveer una infraestructura libre y abierta donde los esfuerzos distribuidos de la comunidad de visión por computadora pudieran ser consolidados y optimizados. Algunas áreas de aplicación son interacción hombre-máquina, identificación de objetos, segmentación y reconocimiento, rastreo de movimientos, reconocimiento de rostros, reconocimiento de gestos, monitoreo, entendimiento de movimiento, biométricas, robots móviles, etc.

Es principalmente una librería de alto nivel que implementa algoritmos como técnicas de calibración de cámaras, detección de características y rastreo, análisis de figuras, análisis de movimiento, reconstrucción en 3D, segmentación y reconocimiento de objetos.

La principal característica de la librería además de su funcionalidad es el desempeño. Los algoritmos están basados en estructuras de datos dinámicas y más de la mitad de las funciones fueron optimizadas a nivel de ensamblador para tomar ventaja de la arquitectura de los procesadores Intel.

ANEXO B: TARJETA DE EVALUACIÓN

El procesador tiene una capacidad de direccionamiento de 4 Gbytes, suficiente para mapear todos los registros de control de los periféricos y todas las memorias del EZ-kit. Este se basa en una arquitectura RISC[18] con un conjunto de instrucciones de 32 bits, con doble módulo de MAC[19] (Multiply-accumulate) de 16 bits. Trabaja a una velocidad máxima de reloj de 300MHz ó 600 millones de operaciones de multiplicación-adición por segundo (600 MMAC por segundo) con planes de alcanzar 1GHz.

Se caracteriza por su bajo consumo de energía 0.15mW/MMAC operando a 0,7V, además de contar con un conjunto de memorias ultrarrápidas internas al procesador que pueden ser usadas a la frecuencia del mismo, con lo cual se pueden tener accesos a memoria sin o con muy poca latencia.

Una desventaja característica es que la unidad aritmética y lógica es de punto fijo, a pesar de esto, usa numerosas técnicas para maximizar el desempeño en el procesamiento de señales e imágenes. Estas características incluyen bloques computacionales en paralelo, uso eficiente del DMA[20] e instrucciones especializadas para el procesamiento de video, se dispone de 2 alus de 8 bits solo para video. En la figura 8 se puede observar un diagrama en bloques del procesador y sus periféricos.

[18] Reduced instruction set computer
[19] Unidad de hardware designada para realizar operaciones de multiplicación y adición de manera eficiente.
[20] Direct Memory Access: Unidad que permite la eficiente interacción entre dispositivos de entrada salida, sin intervención del procesador.

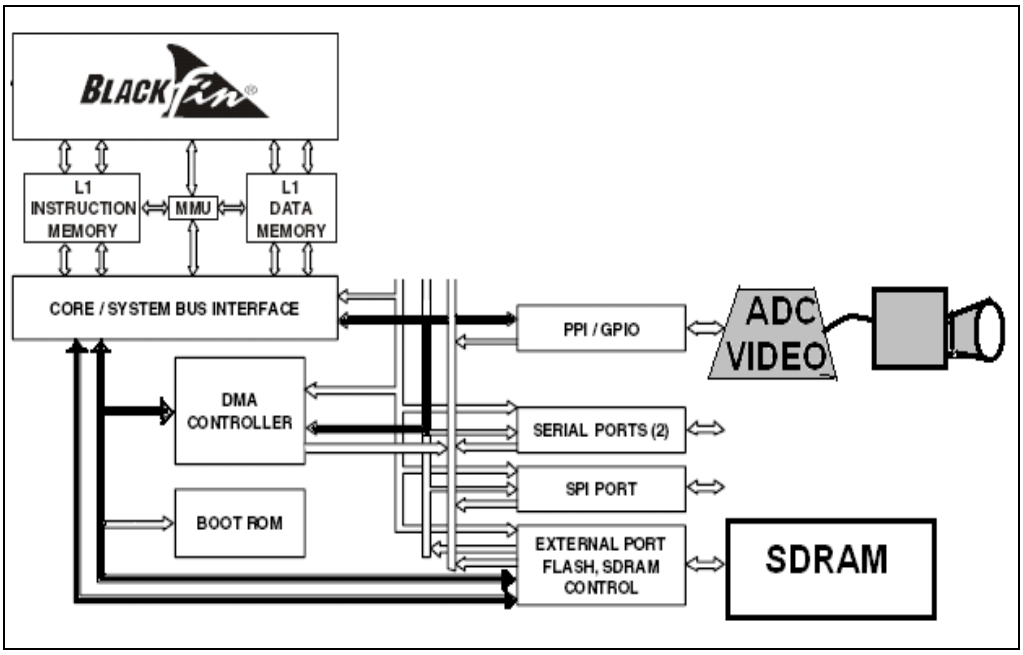


Figura 8. Arquitectura de la tarjeta de desarrollo[21]

Las memorias del procesador están organizadas de manera jerárquica, donde la memoria de primer nivel es la memoria interna al procesador, esta es la más rápida pero a la vez es más pequeña que las otras memorias del EZ-kit.

En total se tienen 68kB de memoria de primer nivel (banco de datos A de L1 con 32 KB, y su banco de datos B con 32 KB y scratchpad memory[22] de 4 KB). Un inconveniente de trabajar con la memoria de primer nivel L1 es que los 32 KB del banco A y los 32 KB del banco B no se encuentran mapeados de forma consecutiva, debido a esto para usar los 64 KB de la memoria de primer nivel sería necesario trabajar el DMA con descriptores, o analizar la imagen por partes,

[21] extraída del HARDWARE REFERENCE MANUAL ADSP BF 533 Blackfin Processor (Pág. 1-2).

[22] Memoria interna de alta velocidad utilizada para almacenamiento temporal de información preliminar

lo cual resulta posible dado que la interrupción del DMA podría ser configurada para que se genere una vez se transfiera cada fila de la imagen, en vez de la imagen completa.

La memoria de segundo nivel es la memoria externa al procesador, esta se conecta al bus del sistema a través del EBIU (External Bus Interface Unit).

Dentro de la memoria de segundo nivel L2 se encuentra la SDRAM (Syncronus Dynamic Ram), esta es el bloque de memoria más grande con la que cuenta el procesador en el Ez-kit. Dado que en la memoria SDRAM es posible almacenar toda la imagen, se configura el DMA para que transfiera la información de luminancia de la imagen a esta memoria que se encuentra mapeada en la primera posición de memoria del procesador (0x0). Se realiza así un submuestreo netamente con hardware, descartando la información cromática y las señales de sincronismo, proceso que además hace las veces de un filtro pasa-bajos que pre-procesa la imagen).

En la figura 10 se muestra una gráfica detallada que nos indica como es el mapeo de todas las memorias, en ella se puede visualizar claramente como está distribuida la memoria interna y la memoria externa **[6]**.

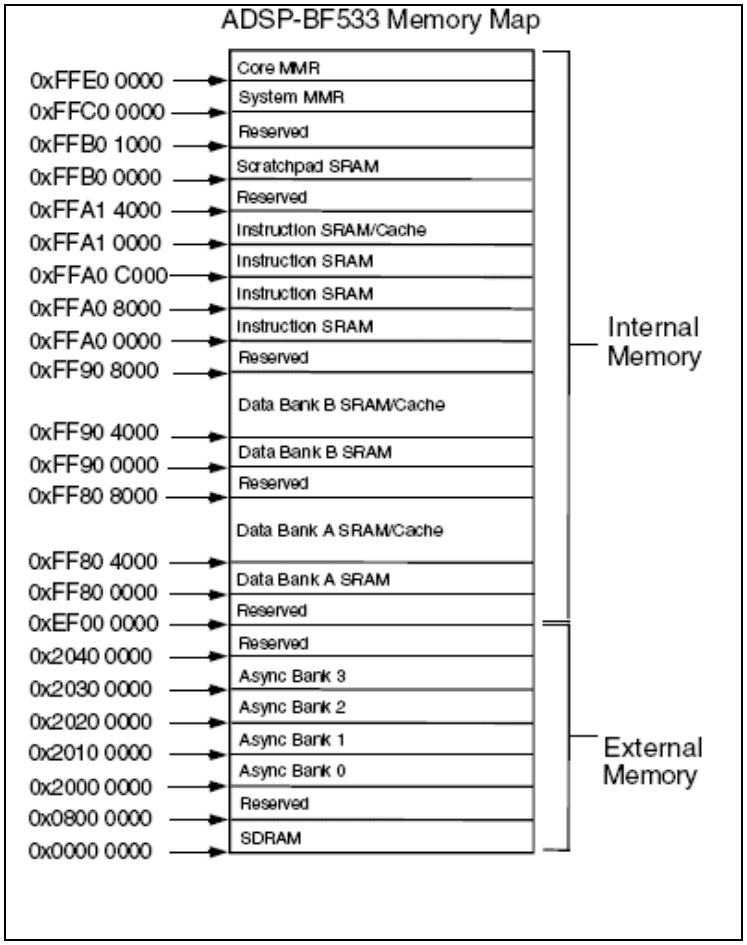
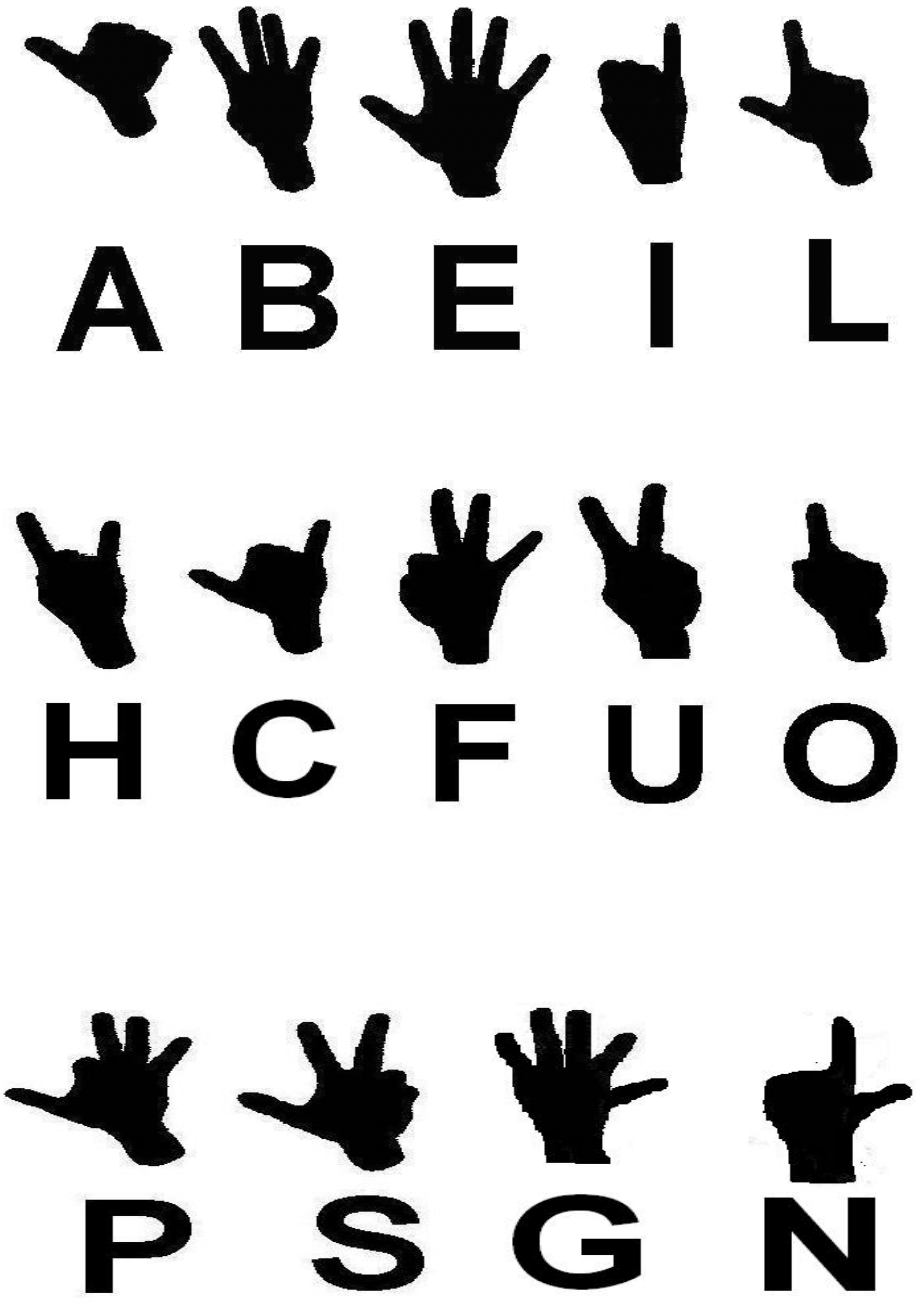


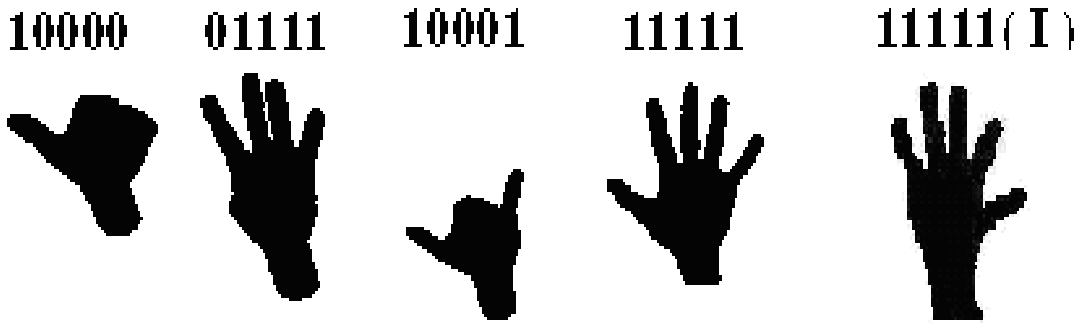
Figura 10. Mapa de memoria[23]

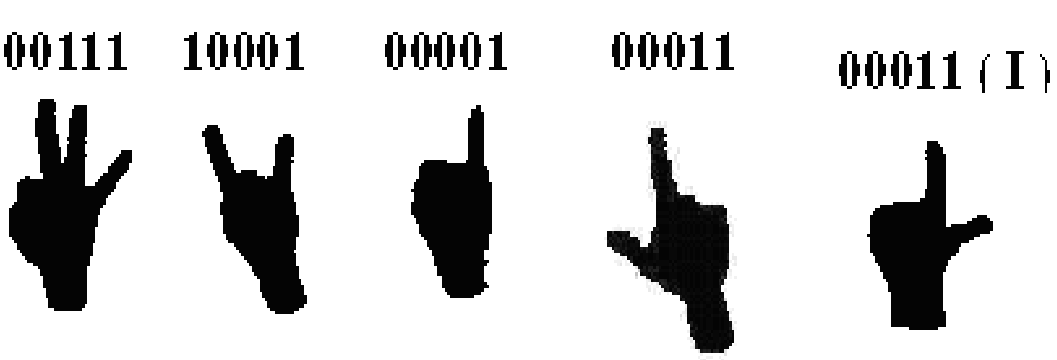
[23] extraída del HARDWARE REFERENCE MANUAL ADSP BF 533 Blackfin Processor (Pág. 6-3).

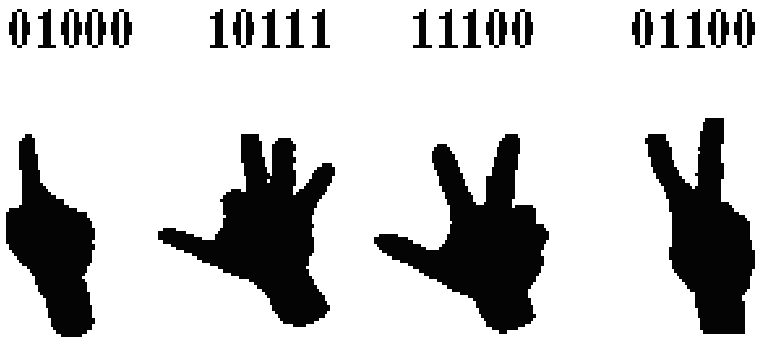
ANEXO C: ALFABETO PARA INTERACTUAR CON LA APLICACIÓN.



Representación binaria de los gestos.







ANEXO D. VECTORES BASE EXTRAÍDOS DE IMÁGENES.

Letra U	
157	189
162	193
159	192
162	192
158	188
155	187
161	193
157	178
159	181
162	185
166	190
155	180
161	186
161	186
172	196
168	192
168	193
171	196
169	194
164	188
175	195
175	193
170	187
169	189
162	191
163	195
166	198
161	190
170	188
172	190
PROMEDIO	**PROMEDIO**
164,5862069	189,862069
DESVIACIÓN ESTÁNDAR	**DESVIACIÓN ESTÁNDAR**
1,041722	0,859048

Letra H	
166	203
171	212
170	210
172	211
150	188
160	204
171	213
163	204
163	202
167	208
166	211
158	205
160	205
174	226
165	220
170	221
168	218
171	220
172	222
172	221
151	198
172	219
158	205
159	207
168	213
PROMEDIO	**PROMEDIO**
165,4583333	210,9583333
DESVIACIÓN ESTÁNDAR	**DESVIACIÓN ESTÁNDAR**
1,306133	1,744377

Letra C	
127	225
121	220
119	221
116	217
118	217
128	227
128	223
132	229
134	229
134	233
141	233
132	230
138	220
138	224
134	233
137	231
PROMEDIO	**PROMEDIO**
129,8125	225,75
DESVIACIÓN ESTÁNDAR	**DESVIACIÓN ESTÁNDAR**
1,88791128	1,36787929

Letra O
170
172
174
173
178
175
162
182
183
174
174
176
175
177
172
PROMEDIO
174,46
DESVIACIÓN ESTÁNDAR
1,308626

Letra L	
118	186
114	181
112	183
118	188
109	181
109	183
108	180
117	187
115	184
117	185
116	183
109	175
114	180
124	191
123	190
125	192
124	188
127	192
126	190
124	190
122	190
122	191
120	188
PROMEDIO	**PROMEDIO**
117,9565217	**186**
DESVIACIÓN ESTÁNDAR	**DESVIACIÓN ESTÁNDAR**
1,276744	**0,979215**

Letra i
203
215
219
198
217
197
192
195
204
205
196
193
207
194
202
196
203
205
220
207
204
223
PROMEDIO
204,3181818
DESVIACIÓN ESTÁNDAR
1,93637

ANEXO E. VECTORES BASE EXTRAÍDOS DE UNA SECUENCIA DE IMÁGENES

- Vectores base para gestos con una rama

I
201
202
215
215
190
203
190
198
200
201
201,5

A
119
112
108
107
117
111
114
113
113
116
113

O
172
164
160
161
163
153
154
165
162
156
161

- Vectores base para gestos con dos ramas

U	
158	191
163	197
167	192
185	208
158	184
193	218
171	205
156	192
145	178
151	180
164,7	194,5

C	
101	218
114	232
110	232
114	232
111	218
109	217
105	215
106	216
114	230
112	232
109,6	224,2

H	
159	225
174	215
155	214
162	210
148	215
153	211
162	219
156	216
155	214
162	211
158,6	215

N	
181	261
173	256
185	255
179	256
182	261
171	258
186	253
178	256
179	258
179	256
179,3	257

L	
101	183
94	179
104	186
110	192
111	194
102	178
102	177
103	180
103	186
104	185
103,4	184

- Vectores base para gestos con tres ramas

F		
186	212	256
178	212	249
163	190	232
173	199	235
168	198	240
156	187	232
163	196	235
181	206	248
180	209	243
178	208	248
172,6	201,7	241,8

S		
104	175	200
101	175	202
97	170	198
94	172	199
99	173	201
104	177	203
100	177	208
97	175	202
95	165	190
99	173	200
99	173,2	200,3

- Vectores base para gestos con cuatro ramas

B			
146	181	198	235
150	182	198	233
156	186	205	237
155	186	203	235
155	188	209	240
156	183	202	233
154	183	201	234
143	181	199	230
143	181	199	230
143	181	199	230
150,1	183,2	201,3	233,7

P			
86	172	205	235
85	172	209	236
86	187	212	235
83	188	209	232
86	196	215	239
86	200	217	240
77	182	197	228
80	84	201	231
81	175	211	233
80	169	207	232
83	172,5	208,3	234,1

- Vectores base para gestos con cinco ramas

E				
88	168	201	218	246
75	143	167	193	252
86	158	190	207	251
86	152	187	202	238
81	153	189	204	239
77	154	189	209	242
85	153	180	196	238
78	149	187	205	240
86	153	188	205	239
78	109	128	168	275
82	149,2	180,6	200,7	246

G				
120	152	175	198	270
130	158	179	205	274
118	157	180	206	281
117	157	176	203	273
121,25	156	177,5	203	274,5

ANEXO F. ESTADÍSTICA DE DURACIÓN DEL PROGRAMA EN VISUAL C++

Profile: Function timing, sorted by time

Date: Thu May 12 12:38:04 2005

Program Statistics

Command line at 2005 May 12 12:25: "C:\Trabajo_de_Grado \morphology"

Total time: 85288,406 millisecond

Time outside of functions: 123,242 millisecond

Call depth: 4338

Total functions: 40

Total hits: 108764837

Function coverage: 90,0%

Overhead Calculated 5

Overhead Average 5