



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica-Eléctrica

SISTEMA DE COMUNICACIÓN VISUAL

Vivian Lucía Orozco Molina

Asesorada por Dr. Juan Carlos Córdova Zeceña

Guatemala, octubre de 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

SISTEMA DE COMUNICACIÓN VISUAL

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA

FACULTAD DE INGENIERÍA

POR

VIVIAN LUCÍA OROZCO MOLINA

ASESORADA POR DR. JUAN CARLOS CÓRDOVA ZECEÑA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERA ELECTRÓNICA

GUATEMALA, OCTUBRE DE 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

| | |
|------------|----------------------------------|
| DECANO | Ing. Murphy Olympo Paiz Recinos |
| VOCAL I | |
| VOCAL II | Lic. Amahán Sánchez Álvarez |
| VOCAL III | Ing. Julio David Galicia Celada |
| VOCAL IV | Br. Kenneth Issur Estrada Ruiz |
| VOCAL V | Br. Elisa Yezminda Vides Leiva |
| SECRETARIO | Inga. Marcia Ivonne Véliz Vargas |

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

| | |
|-------------|--|
| DECANO | Ing. Murphy Olympo Paiz Recinos |
| EXAMINADOR | Ing. Enrique Edmundo Ruiz Carballo |
| EXAMINADORA | Inga. Ingrid Salomé Rodríguez De Loukota |
| EXAMINADOR | Ing. Manuel Fernando Barrera Pérez |
| SECRETARIO | Inga. Marcia Ivonne Véliz Vargas |

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

SISTEMA DE COMUNICACIÓN VISUAL,

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica con fecha 30 de marzo de 2005.

Vivian Lucía Orozco Molina

DEDICATORIA

Dedico este trabajo a Dios, a San Judas Tadeo, a mis padres: Roberto Orozco y Vivian Molina, a mi abuela Nina, a mi hermano Tito y a mis amigos.

AGRADECIMIENTOS

A la Universidad de San Carlos de Guatemala.

A José Antonio de León.

Al Dr. Juan Carlos Córdova

ÍNDICE GENERAL

| | |
|---|-----|
| ÍNDICE DE ILUSTRACIONES | IV |
| RESUMEN | V |
| OBJETIVOS | VI |
| INTRODUCCIÓN | VII |
| | |
| 1. SISTEMA DE COMUNICACIÓN VISUAL | |
| 1.1. Tecleo ocular | 1 |
| 1.1.1. Sistema de tecleo ocular | 1 |
| 1.1.1.1. Teclado virtual | 2 |
| 1.1.1.2. Rastreo ocular | 3 |
| 1.1.1.2.1. Historia de rastreo ocular | 4 |
| 1.1.1.2.2. Técnicas de rastreo ocular | 5 |
| 1.1.2. Proceso de Tecleo Ocular | 8 |
| 1.1.2.1. Enfoque | 9 |
| 1.1.2.2. Retroalimentación | 10 |
| 1.1.2.3. Selección | 11 |
| 1.1.3. Resumen | 12 |
| 1.2. Propuesta del Sistema de Comunicación Visual | 12 |
| | |
| 2. ELEMENTOS DEL SISTEMA | |
| 2.1. Cámara | 17 |
| 2.1.1. Tipos | 18 |
| 2.1.1.1. Cámaras digitales fotográficas | 18 |
| 2.1.1.2. Cámaras web | 19 |
| 2.1.1.3. Cámaras de video | 19 |

| | |
|--|----|
| 2.1.2. Lentes..... | 20 |
| 2.1.3. Sensores | 21 |
| 2.1.4. Selección de cámara | 25 |
| 2.2. Imagen | 26 |
| 2.2.1. Compresión | 26 |
| 2.2.2. Formato de archivo de imagen | 27 |
| 2.2.3. Formato de video..... | 28 |
| 2.2.4. Selección de la imagen | 29 |
| 2.3. Iluminación | 29 |
| 2.3.1. Selección de la iluminación | 33 |
| 2.4. Monitor | 33 |
| 2.5. Lenguaje de programación..... | 33 |
| 2.5.1. Selección del lenguaje de programación..... | 35 |
| 2.6. Montaje de la cámara | 35 |
| 2.6.1. Selección del montaje de la cámara..... | 36 |
| 2.7. Relación usuario-monitor-cámara | 36 |
| | |
| 3. FILTROS 2D Y PROGRAMA | |
| 3.1. Clasificaciones de filtros | 39 |
| 3.2. Convolución y correlación | 40 |
| 3.3. Filtros de interés..... | 42 |
| 3.4. Programa..... | 43 |
| 3.4.1. Versiones | 49 |
| | |
| 4. EVALUACIÓN | |
| 4.1. Criterio de desempeño | 59 |
| 4.1.1. Selección de indicadores de desempeño | 59 |
| 4.1.2. Estadística descriptiva..... | 60 |
| 4.2. Programa de evaluación..... | 62 |

| | |
|---|----|
| 4.3. Desempeño de las distintas versiones | 64 |
| CONCLUSIONES | 67 |
| RECOMENDACIONES | 69 |
| BIBLIOGRAFÍA | 71 |
| APÉNDICE..... | 75 |

ÍNDICE DE ILUSTRACIONES

FIGURAS

| | | |
|-----|--|----|
| 1. | Primera pantalla de EagleEyes..... | 3 |
| 2. | Segunda pantalla de EagleEyes..... | 3 |
| 3. | Imágenes de Purkinje..... | 6 |
| 4. | Teclado virtual del Sistema de Comunicación Visual..... | 13 |
| 5. | Diagrama de bloques del proyecto..... | 15 |
| 6. | Sensor CMOS..... | 23 |
| 7. | Cámara seleccionada..... | 26 |
| 8. | Primera configuración..... | 37 |
| 9. | Segunda configuración..... | 37 |
| 10. | Etapa de comparación en la versión A..... | 51 |
| 11. | Pasos de mejoramiento de una imagen del ojo..... | 54 |
| 12. | Pasos de mejoramiento de una imagen del iris..... | 54 |
| 13. | Proceso de calibración en la versión B..... | 57 |
| 14. | Detección de la orientación actual de la mirada en la versión B..... | 58 |

TABLAS

| | | |
|------|--|----|
| I. | Ubicación de caracteres en el teclado..... | 13 |
| II. | Estadísticas de los coeficientes de correlación..... | 64 |
| III. | Estadísticas de tiempo por palabra (segundos)..... | 65 |
| IV. | Costos materiales..... | 65 |

RESUMEN

El Sistema de Comunicación Visual permite al usuario escribir por medio de la vista. El sistema utiliza un interfaz gráfico en la computadora con representación de las letras, símbolos y funciones del teclado. Se emplea un dispositivo para capturar imágenes oculares. La ubicación y movimientos oculares se analizan por medio de un programa que las transforma en comandos que sustituyan las funciones del teclado, para luego ser transcritas a un documento.

El Sistema de Comunicación Visual corresponde a un sistema de tecleo ocular. En el presente trabajo se proporcionan los antecedentes relevantes de sistemas de este tipo. Además, se presentan los elementos teóricos específicos para la selección de los elementos del sistema y el diseño del programa. Se explican los elementos básicos del proceso llevado a cabo por el sistema; desde la captura de la imagen hasta el análisis de ésta por una computadora y la ejecución de acciones. El trabajo finaliza con una evaluación del sistema y sus distintas versiones.

OBJETIVOS

- **General**

Proporcionar una forma de comunicación escrita para personas con pérdida temporal o permanente del uso de las manos y/o de la voz.

- **Específicos**

1. Implementar algoritmos de adquisición y procesamiento de imágenes que posibiliten la interpretación de señales visuales para su uso en comunicaciones escritas.
2. Construir un sistema de tecleo ocular de bajo costo, fácil de utilizar y con un diseño cómodo para el usuario.

INTRODUCCIÓN

La capacidad de expresarse, de comunicarse, es más que el intercambio de información con otros, por medio de palabras, señales o símbolos; es una necesidad humana, es una parte esencial de quien se es y la forma en que el ser humano es percibido. Es algo que se necesita para sentirnos completos, además de sentirnos seres funcionales en la sociedad.

Debido a la pérdida temporal o permanente del uso de las manos y/o de la voz, a muchas personas se les dificulta o se les hace imposible comunicar sus ideas, sentimientos o necesidades.

El propósito de este trabajo de graduación es implementar un Sistema de Comunicación Visual que ayude a estas personas a comunicarse por escrito sin depender de otras personas.

El Sistema de Comunicación Visual es un sistema de tecleo ocular. Un sistema de tecleo ocular produce texto con la dirección de la mirada. Consiste en un teclado virtual en una pantalla, un dispositivo de rastreo ocular y un programa que ejecute acciones basados en los resultados de éste.

1. SISTEMA DE COMUNICACIÓN VISUAL

1.1. Tecleo ocular

El tecleo ocular consiste en producir texto usando la dirección de la mirada y movimientos oculares. Muchas veces las personas con discapacidad motora solo pueden usar la vista como medio de comunicación, es aquí donde más se observa la utilidad de un sistema de tecleo ocular.

Para poder diseñar un sistema de tecleo ocular, se debe conocer cómo trabaja la trayectoria de la mirada ('gaze path'). La trayectoria de la mirada consiste en la combinación de fijaciones y sacudidas. Las fijaciones consisten en mantener la mirada relativamente quieta (200 a 600 ms) y las sacudidas son movimientos rápidos de un punto a otro que duran de 30 a 120 ms.

Otro elemento a tomar en cuenta es el problema comúnmente encontrado del "toque de Midas": donde quiera que observa el usuario, la dirección de la mirada puede ser interpretada por el sistema como una orden, aunque esta no sea la intención del usuario.

1.1.1. Sistema de tecleo ocular

Un sistema de tecleo ocular generalmente consiste en un interfaz o teclado virtual en una pantalla, un dispositivo de rastreo ocular y un programa que analiza los resultados de éste.

1.1.1.1. Teclado virtual

Muchos sistemas existentes no utilizan el teclado tradicional de formato QWERTY para su teclado virtual. Esto se debe a varias razones; primero, el usuario puede no estar familiarizado con este formato y segundo, por la imprecisión de la mirada.

Un teclado organizado en orden alfabético o que agrupe las letras más comunes puede ser más fácil de utilizar.

Generalmente, mientras más grande el tamaño del la tecla o elemento, mayor es la precisión de selección. El tamaño adecuado dependerá del dispositivo de rastreo ocular, la resolución y tamaño de la pantalla y de la distancia entre el usuario y la pantalla.

Desafortunadamente, los elementos grandes reducen el espacio en la pantalla. Algunos sistemas que tienen pocos elementos en la pantalla utilizan un método jerárquico para seleccionar un carácter. Por ejemplo, en el sistema EagleEyes [Gips y Olivieri, 1996] existen dos niveles. El primer nivel tiene elementos con grupos de caracteres (Fig. 1), se debe seleccionar uno de estos elementos para poder seleccionar la letra de este grupo (Fig. 2), éste es el segundo nivel. También son comúnmente utilizadas técnicas de predicción de palabras.

Figura 1. Primera pantalla de EagleEyes

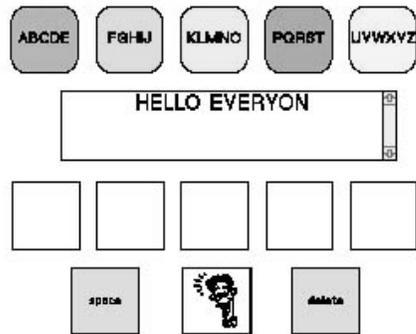
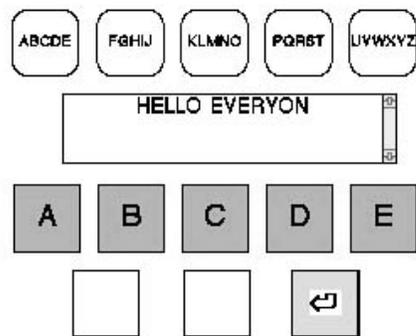


Figura 2. Segunda pantalla de EagleEyes



1.1.1.2. Rastreo ocular

Cuando observamos a alguien, nuestra mirada tiende a ir a los ojos de la otra persona porque sabemos que es la parte más reveladora de la cara humana. Con este simple acto mostramos la bidireccionalidad de los ojos, ya que con ellos percibimos información de nuestro entorno, pero también damos información. Con la mirada indicamos una dirección y qué nos llama la atención.

Debido a la habilidad humana de comunicarnos con la mirada, se han hecho investigaciones sobre el rastreo ocular y sus aplicaciones.

El rastreo ocular ('eye-tracking') consiste en el estudio de la mirada y movimientos oculares por medio de algún dispositivo.

1.1.1.2.1. Historia de rastreo ocular

Desde 1878 (Javal) se han hecho investigaciones sobre el seguimiento ocular, los métodos iniciales involucraban contacto directo con la córnea. En los años 30, Miles Tinker usó la fotografía para estudiar los movimientos oculares durante la lectura. Paul Fitts y algunos colegas usaron la cámara de filmación, en 1947, para estudiar el movimiento ocular de los pilotos al usar los controles de cabina en el momento de aterrizaje. Para esta época se inventó el primer rastreador ocular ('eye tracker') sujetado en la cabeza (Hartridge y Thompson, 1948), permitiendo al usuario mayor libertad de movimiento.

La introducción de la mini y microcomputadora en los años 70, dieron lugar al rastreo ocular en tiempo real. Surgieron grandes avances enfocados principalmente en los aspectos psicológicos y fisiológicos del rastreo ocular. La mayoría de las investigaciones eran realizadas por laboratorios militares en el campo de la aviación.

Debido a la proliferación de computadoras personales, el rastreo ocular se utilizó como instrumento de evaluación en el campo de interacción humano-computadora (Card, 1984), en los 80's. También se propuso por primera vez el uso del rastreo ocular como medio de comunicación entre el usuario y la computadora, enfocándose principalmente en usuarios con necesidades especiales (Hutchinson, 1989; Levine, 1981). Debido a los avances de la tecnología, se ha seguido investigando e invirtiendo en el desarrollo de mecanismos de rastreo ocular para la interacción humano-computadora (Vertegaal, 1999; Jacob, 1991; Zhai, Morimoto & Ihde, 1999).

1.1.1.2.2. Técnicas de rastreo ocular

Las técnicas actuales de rastreo ocular se pueden clasificar según la forma en que hacen contacto con el usuario.

1. Técnicas basadas en la luz reflejada: se mide la reflexión de alguna luz dirigida al ojo. Generalmente se utiliza luz infrarroja para evitar distraer al usuario y evitar la interferencia de otras fuentes de luz.

2. Técnicas basadas en el potencial eléctrico de la piel: Mide el potencial eléctrico de la piel alrededor de los ojos.

3. Técnicas basadas en lentes de contactos: Se aplica un lente de contacto especial que facilita el rastreo de su posición.

Todas las técnicas requieren calibración antes de utilizarse, y en la mayoría de los casos de re-calibración.

1. Técnicas basadas en luz reflejada

1.1 Rastreo del limbus

El limbus es la frontera entre el iris oscuro y la esclerótica blanca del ojo. Debido al contraste entre la esclerótica y el iris esta frontera se puede detectar y rastrear fácilmente. Esta técnica se basa en la posición y forma del limbus en relación a la cabeza. Por lo tanto, la cabeza se debe mantener casi estacionaria o el dispositivo debe estar sujetado a la cabeza del usuario. Debido al recubrimiento temporal de la parte inferior y superior del limbus por los parpados se considera que el rastreo del limbus es adecuado únicamente para un rastreo horizontal (Scott y Findlay, 1993).

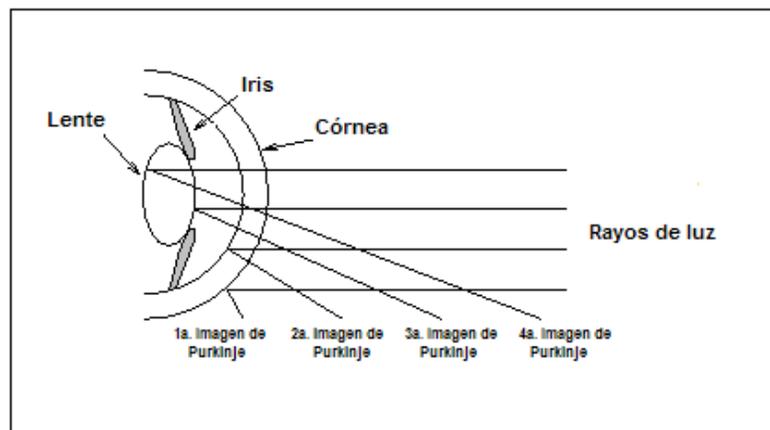
1.2 Rastreo de la pupila

Esta técnica es similar a la anterior pero utiliza la frontera entre la pupila y el iris. La cabeza debe mantenerse estacionaria con respecto al dispositivo. Debido a que la pupila se cubre mucho menos por los párpados que el limbus, se puede rastrear verticalmente además de horizontalmente. La frontera de la pupila es más aguda que la del limbus permitiendo una mayor resolución, pero a la vez el contraste entre la pupila y el iris es menor, dificultando la detección de bordes.

1.3 Relación entre el reflejo de la córnea y el de la pupila

Cuando se alumbra al ojo del usuario con luz (de preferencia infrarroja) hay varios reflejos sobre las fronteras del lente y la córnea, estos se llaman imágenes de Purkinje (véase figura 3). La primera imagen de Purkinje junto con el reflejo de luz de la retina se puede grabar utilizando una cámara sensible a infrarrojo. Con los movimientos oculares, la posición relativa de la primera imagen de Purkinje y el centro del reflejo de luz de la retina cambian y la dirección de la mirada se puede calcular de las posiciones relativas. El problema de esta técnica es obtener una buena vista del ojo ya que mover la cabeza puede desenfocar la imagen de video del ojo.

Figura 3. Imágenes de Purkinje



1.4 Reflejo de la córnea e imagen del ojo usando una red neuronal artificial

Esta técnica fue desarrollada recientemente y en ella los cálculos son realizados por una red neuronal artificial (Baluja y Pomerleau, 1994). Siempre se utiliza una imagen digitalizada del usuario pero con un ángulo más amplio del usuario donde se observe toda la cabeza. Una luz estacionaria se coloca en frente del usuario y el sistema inicia encontrando el ojo derecho del usuario por medio de la búsqueda en la imagen de video del reflejo de esta luz (primera imagen de Purkinje). Esta se distingue porque es un punto pequeño y brillante en medio de una región más oscura. Luego extrae una parte rectangular de la imagen de video centrado en la primera imagen de Purkinje y manda esto a la red neuronal. La salida de la red neuronal es un conjunto de coordenadas.

La red neuronal requiere una calibración más elaborada que otras técnicas ya que debe recolectar imágenes durante por lo menos tres minutos mientras el usuario rastrea con la mirada un cursor que se mueve en la pantalla. Luego hay una sesión de entrenamiento que dura treinta minutos utilizando las imágenes guardadas, pero el sistema no requiere de re-calibración.

La precisión de este sistema es menor que la de las otras técnicas pero el usuario tiene una mayor libertad de movimiento.

1.5 Rastreo de Imágenes de Purkinje

El rastreo de la primera y cuarta imagen de Purkinje se puede usar para rastrear la dirección de la mirada usando la técnica de la imagen dual de Purkinje (Muller et al., 1993), que utiliza las posiciones relativas de estos reflejos para calcular la dirección. Generalmente esta técnica es más precisa que las demás técnicas y la frecuencia de muestreo es alta, hasta 4 kHz.

Debido a que la cuarta imagen de Purkinje es débil, se debe controlar mucho la luz ambiental.

2. Técnica basada en el potencial eléctrico de la piel

La electro-oculografía se basa en el hecho que existe un campo electrostático que rota junto con el ojo. Al grabar pequeñas diferencias de potencial eléctrico alrededor de la piel del ojo, se puede detectar la posición del mismo. Esto se hace con electrodos colocados en la piel alrededor del ojo. Esta técnica es inadecuada para uso diario ya que requiere del contacto de electrodos con el usuario.

3. Técnicas basadas en lentes de contacto

En esta técnica el usuario debe usar unos lentes de contacto especiales. Existen dos técnicas basadas en lentes. La primera consiste en insertar en un lente de contacto una o mas superficies de espejo planas, las reflexiones de haces de luz se pueden utilizar para calcular la posición del ojo. La segunda técnica consiste en colocar una pequeña bobina de inducción en el lente, la posición exacta del lente se puede conocer a través del uso de campos electromagnéticos alrededor de la cabeza del usuario. Debido a riesgos de salud esta técnica es inadecuada para uso diario.

1.1.2. Proceso de tecleo ocular

Al utilizar un sistema de tecleo ocular el usuario sigue un proceso que se puede subdividir en tres partes: enfoque, retroalimentación y selección. Primero, el usuario debe enfocar su vista en una de las teclas virtuales. Segundo, el sistema resalta la letra enfocada proporcionando retroalimentación.

Tercero, si el usuario desea seleccionar la letra resaltada, él continúa viéndola, y de esta manera utiliza el tiempo como un comando de activación. Otra alternativa de selección, es el disparo de un interruptor como comando de activación. Además, el sistema puede dar retroalimentación auditiva indicando que se presionó exitosamente la tecla de forma virtual. De último, aparece la letra en un campo de texto en la pantalla.

A continuación se entra en más detalle del proceso de tecleo ocular, además de mencionar sistemas existentes que ejemplifican diferentes formas de realizarlo.

1.1.2.1. Enfoque

Antes de que el usuario pueda seleccionar algo, debe enfocarlo. Existen diferentes métodos de enfoque dependiendo de la técnica utilizada de rastreo ocular y de las necesidades del cliente para el cual fue diseñado.

Enfoque tipo mouse: Con los ojos se mueve un cursor en la pantalla. Dos ejemplos de esto son los sistemas EagleEyes [Gips et al., 1993] y EyeWriter [Lileg et al., 1999]. Ambos se basan en la electro-oculografía.

Enfoque basado en tiempo: Se debe observar el elemento fijamente por un tiempo definido ('dwell time'). Este tiempo puede ser modificada dependiendo de la experiencia del usuario. Este método y el tipo mouse son los más utilizados.

Enfoque basado en barrida: Este método se usa para personas a quienes se les dificulta sostener la mirada debido a su estado de salud. A veces solo pueden mover los ojos en una dirección.

En este caso los ojos se pueden usar como interruptores utilizando el método de barrida. Por ejemplo, un interruptor podría ser mirar hacia arriba. En la barrida por escalón ('step-scanning'), el usuario puede usar un interruptor para cambiar el enfoque de un elemento a otro y otro interruptor para seleccionar el elemento.

En la barrida automática, el usuario solo tiene un interruptor. El enfoque va automáticamente de un elemento a otro después de un tiempo predefinido. Lo único que tiene que hacer el usuario es activar el interruptor cuando el elemento deseado está enfocado.

Debido a la lentitud de realizar una barrida, es aconsejable hacer una barrida de grupos grandes de elementos antes de enfocar un elemento individual. La barrida se utiliza en el sistema VisionKey [Kahn et al., 1999] y en Eye-Switch Controlled Communication Aids [Kate et al., 1979].

1.1.2.2. Retroalimentación

El usuario necesita saber que ha enfocado correctamente y si está ejecutando la acción que desea realizar. Para esto, es indispensable una retroalimentación fácil de entender.

La retroalimentación debería decirnos qué elemento está enfocado en ese momento, además de decirnos cuando se puede seleccionar.

Una forma de mostrar el enfoque es mover un cursor, pero esto puede distraer la atención del usuario. Otra manera es resaltar el elemento, ya sea cambiando el color de fondo o enmarcándolo.

También puede haber retroalimentación auditiva en la forma de un sonido sencillo al enfocar o seleccionar un elemento.

El cursor o elemento resaltado pueden cambiar de color o forma dependiendo de la etapa del proceso. En el sistema Eye-Gaze Response Interface Computer Aid (ERICA) [Hutchinson et al., 1989], esto se logra con un rectángulo que va empequeñeciéndose.

1.1.2.3. Selección

Para seleccionar un elemento existen varias alternativas, las más comunes se describen a continuación.

Dwell time: Ver fijamente un elemento por cierto tiempo es el método más común de selección. Generalmente dura de 600 a 1000 milisegundos, y puede ser interrumpida viendo hacia otro lado antes de que termine el tiempo.

Parpadear y Guiñar: Si se utiliza un parpadeo como método de selección se debe separar de los parpadeos naturales que ocurren de forma inconsciente.

Actividad muscular: El mecanismo de selección podría ser alguna actividad muscular facial u otra.

Movimiento ocular: Una mirada o secuencia de miradas hacia la izquierda, derecha, arriba o abajo se puede interpretar como un comando de selección o de desección.

1.1.3. Resumen

En resumen, un sistema de tecleo ocular se basa en las técnicas desarrolladas de rastreo ocular. El sistema de tecleo ocular necesita un interfaz que dependa de las necesidades del usuario y de la precisión del dispositivo de rastreo ocular. Además, debe tener un programa de análisis de los movimientos oculares que indique al usuario, enfoque y selección, utilizando la retroalimentación.

1.2. Propuesta del Sistema de Comunicación Visual

El presente proyecto consiste en un interfaz PC /usuario, a través del cual la persona podrá escribir en una computadora, solamente utilizando el parpadeo y la orientación de la mirada.

En la pantalla aparece un teclado (Fig. 4) con únicamente nueve teclas. Esto se realizó así para poder distinguir de mejor forma las pequeñas diferencias de la orientación de la mirada. Las teclas son de color verde, y cuando el usuario observa una de las teclas, ésta se torna de color rojo. Para poder usar todos los caracteres del teclado, cada tecla contiene cinco letras (Tabla 1). Si la persona sigue observando fijamente la misma tecla, ésta cambiará su contenido, es decir la “etiqueta” de la letra cambiará a la siguiente. Cuando el usuario se haya ubicado en la letra deseada deberá cerrar los ojos. Al hacer esto se escuchará un sonido y la letra seleccionada se escribirá a un documento, además de aparecer en un cuadro de texto en la pantalla.

Figura 4. Teclado virtual del Sistema de Comunicación Visual

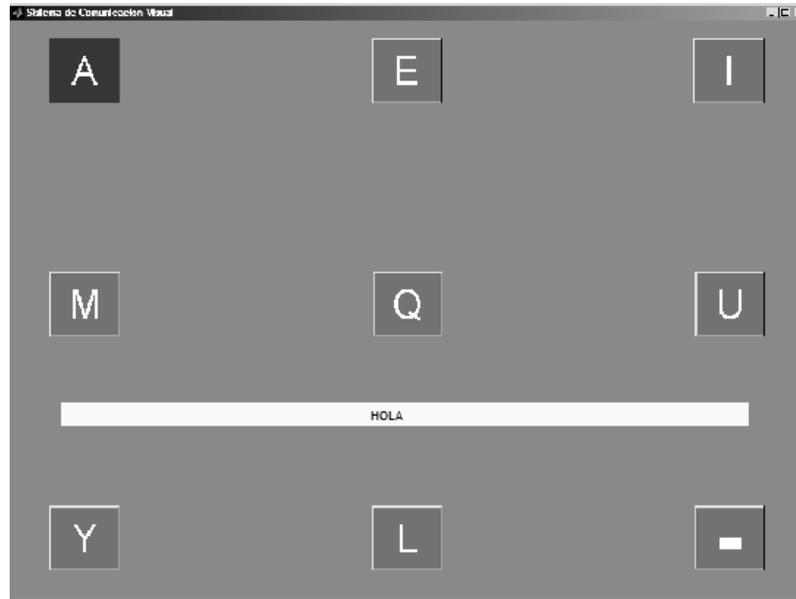


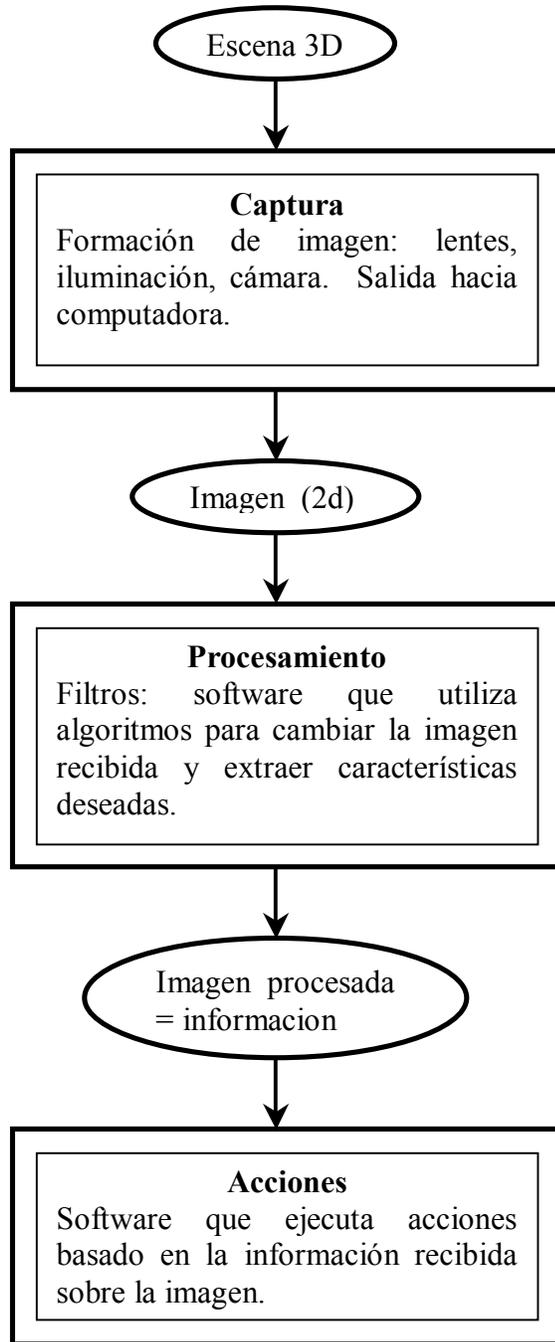
Tabla I. Ubicación de caracteres en el teclado

| NO. DE TECLA | CONTENIDO | | | | |
|--------------|-----------|---|------|---|-------|
| | 1 | A | B | C | D |
| 2 | E | F | G | H | 2 |
| 3 | I | J | K | L | 3 |
| 4 | M | N | O | P | 4 |
| 5 | Q | R | S | T | 5 |
| 6 | U | V | W | X | 6 |
| 7 | Y | Z | 0 | ? | 7 |
| 8 | . | ; | , | : | 8 |
| 9 | ENTER | _ | SUPR | 9 | SALIR |

Funcionamiento

El proyecto controla los parámetros de una cámara, captura imágenes cuando es necesario, procesa las imágenes capturadas para resaltar información de interés, analiza la imagen para determinar la dirección de la mirada y ejecuta acciones basadas en este resultado, tales como escribir en un documento y proveer retroalimentación al usuario. A continuación se encuentra un diagrama de bloques (Fig. 5) de los distintos procesos realizados en el proyecto.

Figura 5. Diagrama de bloques del proyecto



Las etapas principales de este proceso son la captura de imágenes, el procesamiento de imágenes y la ejecución de acciones.

1. Captura

Consiste en la formación de la imagen y su transmisión hacia la computadora. La formación de la imagen incluye la iluminación, el lente y la cámara. La iluminación debe ser diseñada para resaltar los rasgos críticos y el lente debe ser capaz de proveer una imagen precisa de ellas. La cámara graba de forma electrónica las imágenes. La información de la cámara es transferida hacia la computadora por medio de un interfaz que acepta los datos entrantes y maneja su transferencia.

La salida de esta etapa es un archivo de imagen. La calidad y tamaño del archivo de imagen serán determinados por el tamaño de la imagen, la cantidad de compresión y el formato del archivo.

2. Procesamiento

En el procesamiento se utilizan varios algoritmos para mejorar elementos de la imagen que son de importancia específica a la aplicación. La imagen mejorada es analizada, se identifican y se miden algunos rasgos, con el objetivo de obtener datos importantes para la siguiente etapa. En resumen, el procesamiento de imágenes se utiliza para extraer información importante sobre los objetos bajo inspección.

3. Ejecución de acciones:

Dependiendo de la información obtenida de la etapa anterior sobre la imagen se toman decisiones y se realizan acciones predeterminadas. Estas acciones corresponderán a las etapas del proceso de tecteo ocular mencionados anteriormente, que son enfoque, retroalimentación y selección.

2. ELEMENTOS DEL SISTEMA

El Sistema de Comunicación Visual se compone de los siguientes elementos: cámara, imagen, iluminación, lenguaje de programación y monitor. Resultó necesario hacer un estudio de los diferentes elementos ya que estos dictan el comportamiento del sistema. Utilizando la información recolectada se pudo tomar decisiones con respecto a la selección y/o diseño de cada elemento. Se buscó que cada elemento cumpliera su función dentro del sistema y se encontrara en armonía con el objetivo de construir un sistema de tecleo ocular de bajo costo, fácil de utilizar y con un diseño cómodo para el usuario.

En este capítulo se profundiza en la teoría aplicable de cada elemento y se exponen las decisiones tomadas con respecto a su diseño y/o selección. El capítulo finaliza con una descripción de la disposición física de los elementos: cámara, monitor, y usuario.

2.1. Cámara

Debido que la imagen es la única fuente de información, es vital que su calidad satisfaga las necesidades del proyecto. La calidad de la imagen dependerá en gran parte de la cámara y entre sus propiedades podemos mencionar:

Resolución es el alto y ancho de la imagen medido en píxeles.

Brillo es el uso de colores claros en lugar de oscuros. Una imagen con mucho brillo tiene más colores claros que oscuros.

Contraste indica el margen de valores entre los colores más claros y los más oscuros de una imagen. Una imagen con mucho contraste tendrá valores de luz y oscuridad bien definidos; una imagen con poco contraste tendrá la mayoría de los colores en tonos medios, con un efecto gris u oscuro.

Saturación es la intensidad de color de la imagen. Una imagen saturada tiene colores intensos; una imagen con saturación baja tiene colores apagados o atenuados.

Equilibrio de blanco es la proporción de componentes rojo, azul y verde requerida para mostrar blanco verdadero en una cámara determinada.

Exposición es el resultado de abrir el objetivo (apertura) y el tiempo que permanece abierto (la velocidad del obturador). véase 2.1.1.1.

2.1.1. Tipos

Las cámaras varían de tamaño, precio, tipo de sensores, interfaces y otras características. Debido a las necesidades del proyecto se hará énfasis en las cámaras digitales fotográficas, las cámaras web y en las cámaras de video.

2.1.1.1. Cámaras digitales fotográficas

Para formar la imagen las cámaras digitales utilizan un lente que enfoca luz a través de la apertura de un diafragma y un obturador hacia un plano focal, donde se encuentra un sensor de imagen. El lente u objetivo es esencialmente uno o varios pedazos de vidrio, u otra sustancia transparente, que canaliza la luz que reflejan los objetos situados ante él. La apertura del diafragma controla la cantidad de luz que llega al plano focal, y se mide con una escala de la forma $f/\text{número}$, donde el número determina el diámetro de la apertura. El obturador se abre y cierra para determinar la cantidad de tiempo que entra la luz, el tiempo que permanece abierto se denomina velocidad de obturación, y es variable.

Al presionar el botón de disparo se ajusta la apertura y la velocidad de obturación para una exposición correcta. La imagen formada es capturada por el sensor de imagen. Cada píxel del sensor de imagen acumula una carga eléctrica proporcional a la intensidad de luz que cae sobre él. Al terminar la exposición la carga de cada píxel se mide y se convierte a un número digital. Estos números pueden ser utilizados para reconstruir la imagen estableciendo el color y brillo de los píxeles correspondientes de una pantalla u hoja impresa.

La imagen es guardada en un dispositivo removible de almacenamiento, por ejemplo una tarjeta de memoria flash.

Algunos parámetros importantes de los lentes se mencionan en el inciso 2.1.2. y los sensores de imagen se describen en el inciso 2.1.3..

2.1.1.2. Cámaras web

Una cámara web consiste en una cámara digital que se puede conectar a la computadora para adquirir imágenes en tiempo real. Se conecta a la computadora por medio de un cable hacia el puerto USB. Las cámaras web son pequeñas y de bajo costo.

2.1.1.3. Cámaras de video

Las cámaras de video utilizan los mismos sensores de imagen que las cámaras digitales fotográficas. De acuerdo a la salida que proporcionan, hay dos tipos principales de cámaras de video: analógica y digital; aunque, dentro de estas categorías hay varias permutaciones. Por ejemplo, la información de video puede ser transmitida desde la cámara a la computadora por medio de diferentes interfaces.

Cámaras analógicas

Las cámaras analógicas tienen una salida de señal de video en forma analógica. La señal analógica es digitalizada por una tarjeta de adquisición de imágenes. El estándar analógico para la representación de video es el más común, dado que está basado en el estándar de televisión.

Las cámaras analógicas son muy utilizadas en aplicaciones de visión artificial debido a su bajo costo, su cableado de fácil adquisición, por la madurez de su tecnología y sus estándares conocidos.

Cámaras digitales

Las cámaras digitales digitalizan la señal antes de enviarla a la tarjeta de adquisición de imágenes. De esta manera, son menos susceptibles a la introducción de ruido en la transmisión.

2.1.2. Lentes

Algunos parámetros básicos se describen en seguida.

Formato de la cámara: El formato de la cámara nos provee con las dimensiones del sensor de imagen. Los lentes deben cubrir un área igual o mayor al formato de la cámara.

Campo de vista: El campo de vista (FOV, field of view) es el área del objeto que es proyectada desde el lente hacia el sensor de imagen y debe cubrir todos los rasgos a medir. Los rasgos dentro del campo de vista deben aparecer suficientemente grandes para ser medidos.

Magnificación requerida: La magnificación requerida es de:

Ecuación 1

$$Mag = \frac{W_{camara}}{W_{FOV}}$$

donde W_{camara} es el ancho del sensor de la cámara y W_{FOV} es el ancho del campo de vista.

Distancia focal: Es la distancia entre el lente y el plano de imagen de la cámara cuando el objeto está en el infinito.

2.1.3. Sensores

La imagen es capturada utilizando un sensor de imagen en el plano de la imagen. El sensor de imagen es el corazón de una cámara digital

La mayoría de los sensores de imágenes son sensores de estado sólido (semiconductores) y están hechos de silicio. Cuando la luz (fotones) cae sobre el silicio, los electrones en el silicio se excitan y los lazos covalentes que los unen a los átomos de silicio, se rompen. El número de electrones liberados es proporcional a la luz incidente. En el diseño de las arquitecturas de sensores de imagen se utilizan métodos para controlar y medir este efecto fotoeléctrico. Naturalmente, han surgido diferentes diseños. Actualmente, los sensores de imágenes mas comunes son: el sensor basado en dispositivos de acoplamiento de carga (CCD, *charge coupled device*) y el sensor CMOS (semiconductor de oxido de metal complementario).

Ambos sensores tienen áreas fotosensibles, llamadas píxeles, localizadas geométricamente en la superficie. Cada píxel actúa como un convertidor optoelectrónico, porque acumula una carga eléctrica proporcional a la luz incidente durante un período de tiempo, denominado tiempo de integración. De esta forma, un píxel contiene una parte de la información de la imagen.

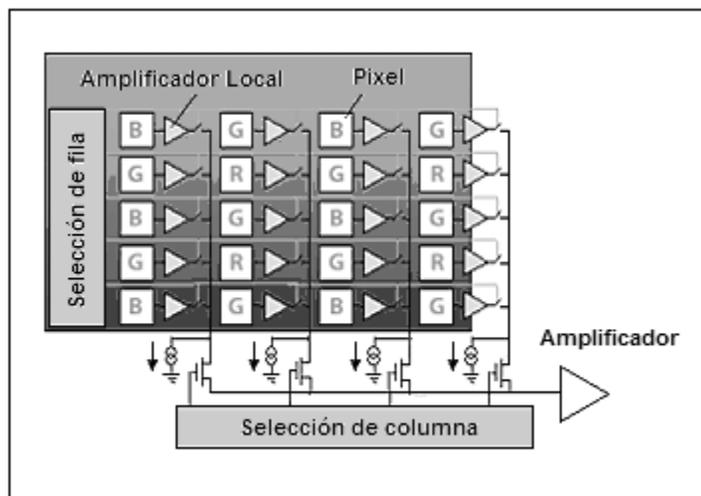
Un sensor de imagen necesita cumplir cinco tareas básicas: absorber fotones, generar una carga a partir de estos fotones, recolectar esta carga, transferir esta carga y convertirla en un voltaje. Las primeras tres tareas se realizan de forma similar en los sensores CCD y CMOS, pero las últimas dos se realizan de manera diferente.

CCD: Las cargas del arreglo de píxeles están acopladas para poder ser transferidas hacia la salida del sensor. Las cargas son desplazadas de una fila horizontal de píxeles hacia la siguiente fila horizontal de píxeles, desde arriba hacia abajo del arreglo. Es decir, que hay varios registros de desplazamiento paralelos (verticales). Cuando una fila de carga se mueve verticalmente, la siguiente fila se desplaza hacia los píxeles recién vaciados. La última fila es un registro de desplazamiento serial u horizontal. Las cargas en esta fila son amplificadas y luego transferidas hacia fuera del sensor como una señal analógica. Esta señal es enviada a un chip de conversión analógico a digital (CAD) y la información digital resultante es la información capturada de la imagen por el sensor antes de cualquier procesamiento posterior.

Actualmente, los sensores CCD, proveen mayor resolución y tienen menos ruido.

CMOS: Cada píxel del arreglo en un sensor CMOS puede tener su propio amplificador y circuitería de salida. Con esto se elimina la necesidad de acoplamiento de carga. Los dispositivos CMOS consumen menos energía que un dispositivo CCD y son más baratos de fabricar. Además, los sensores CMOS pueden integrar otras funciones en el mismo sensor como la conversión de analógico a digital. Debido a esto el tamaño de la cámara se reduce.

Figura 6. Sensor CMOS



Además del tipo de sensor que utilizan, las cámaras digitales se pueden clasificar según la geometría de los píxeles en el sensor.

Cámaras lineales: Pueden ser manufacturadas con tecnología CCD o CMOS. Una cámara lineal (“linescan”) utiliza únicamente una línea de píxeles para crear imágenes. El concepto de barrido lineal se asocia a estas cámaras, es decir, la construcción de una imagen línea a línea, utilizando un sensor lineal. El sensor captura una fila de datos, la lee, y luego captura la fila de datos que sigue. Las cámaras lineales tienen típicamente de mil a cuatro mil píxeles, pudiendo proveer de esta forma imágenes de alta resolución.

Tienen altas tasas de transferencia de datos, pero su costo es elevado y la iluminación debe ser intensa.

Cámaras matriciales: Utiliza varias líneas de píxeles para que compongan un cuadro para crear imágenes. Generalmente, una cámara matricial o de barrida de área (“area scan”) produce una imagen de una área con una razón de ancho a alto de aproximadamente 4 a 3. Esto se debe a especificaciones estándares de televisión. Actualmente, algunas cámaras tienen sensores cuadrados (1:1).

Otro parámetro importante de las cámaras digitales es su formato. El tamaño de un sensor de imagen es su formato. El formato es importante porque determina el tamaño y proporciones de la imagen resultante. El nombre de un formato se mide en pulgadas aunque no tiene relación con las dimensiones físicas del sensor. Por ejemplo, un sensor con formato de ½ pulgada es de 4.8 mm x 6.4 mm.

Cámaras de color

Hasta ahora se ha visto que los sensores de imagen de estado sólido nos dan información únicamente de la intensidad de la luz incidente. En las cámaras de color, se muestrean los componentes rojos, verdes y azules para reproducir color.

Las cámaras de color caen en dos categorías: cámaras de chip único y cámaras de tres chips.

Cámaras de chip único: Se utiliza solo un sensor y un arreglo de filtros de colores sobre los píxeles.

Las cámaras de chip único requieren de circuitería y algoritmos de interpolación más complicados, pero suelen ser más pequeñas, más económicas y tener mayor sensibilidad a la luz baja que las diseñadas con tres chips. Aunque, tiene menos resolución y no es tan precisa en su reproducción del color.

Cámaras de tres chips: La luz que entra al cuerpo de la cámara se divide primero en tres haces y luego se enfoca en tres sensores de imagen, donde cada uno tiene un filtro de color. La función de respuesta en frecuencia de los filtros de color se puede diseñar para simular los del ojo humano. Los beneficios de este diseño incluyen un procesamiento simple de la señal, reproducción precisa del color y una alta resolución. Dentro de sus desventajas están su alto precio, mayor tamaño y poca sensibilidad a la luz baja.

2.1.4. Selección de cámara

Se seleccionó una cámara web debido a su bajo costo, su interfaz con la computadora que nos permite adquirir imágenes en tiempo real, y su pequeño tamaño que facilita su montaje.

La cámara seleccionada es de marca Logitech y modelo 'QuickCam for Notebooks Pro', utiliza un sensor CCD y tiene una resolución máxima de 640x480. Puede capturar hasta 30 cuadros por segundo. Aun con poca iluminación produce una imagen de buena calidad. Su tamaño es de 8 cm de alto por 3.5 cm de largo por 1.5 cm de ancho.

Figura 7. Cámara seleccionada



Se pueden modificar las siguientes propiedades de la cámara:

- Resolución
- Brillo
- Contraste
- Saturación
- Equilibrio de blanco
- Exposición

2.2. Imagen

Al momento de tomar imágenes, hay varias decisiones que se pueden tomar sobre el tamaño de la imagen, cantidad de compresión y el formato del archivo de imagen. Estas decisiones afectarán la calidad de imagen, la memoria utilizada y el tiempo de procesamiento.

2.2.1. Compresión

Los archivos de imágenes a veces son muy grandes para almacenar, transmitir y editar fácilmente. La compresión vuelve a los archivos de imágenes más manejables y pequeños.

Durante la compresión, datos redundantes o sin valor son eliminados o guardados de una forma más pequeña. Esto reduce drásticamente el tamaño del archivo, cuando la imagen es desplegada este proceso es invertido.

Hay dos formas de compresión –sin pérdidas y con pérdidas.

Compresión sin pérdidas: La imagen se descomprime de tal forma que la calidad es igual a la imagen original. Desafortunadamente, los archivos en este tipo de compresión siguen siendo muy grandes.

Compresión con pérdidas: Este proceso sí tiene pérdidas y deteriora la calidad de la imagen. Entre mayor la compresión, mayor será la deterioración. Pero, es más práctico debido a su reducido tamaño.

2.2.2. Formato de archivo de imagen

Es importante conocer el formato del archivo de imagen de salida de la cámara, ya que este será la entrada a la etapa de procesamiento. Se debe poder manipular y procesar de forma satisfactoria la imagen.

Hay diferentes opciones de formato de archivo de imagen.

JPEG: Es el formato más popular. La imagen se almacena utilizando compresión con pérdidas. Tiene como ventaja que se puede variar la cantidad de compresión y por ende la calidad de imagen.

TIFF: Utiliza compresión sin pérdidas.

RAW: Este formato almacena los datos directamente del sensor de imagen sin pasar por el procesamiento de la cámara.

PNG: Es un formato universal que utiliza compresión sin pérdidas.

2.2.3. Formato de Video

Las cámaras analógicas de video pueden proveer una señal estándar (RS-170, NTSC, CCIR, PAL, etc.) o una señal no estándar. Las cámaras estándares son más fáciles de configurar, mientras que las cámaras no-estándar proveen características diferentes, como una mayor resolución o velocidad de cuadros. Para una cámara no estándar se debe usar un dispositivo de adquisición de imágenes que sea configurable a dicha cámara.

Las cámaras de video digitales tienen diferentes opciones de interfaces: paralelo digital (RS 422 y LVDS), camera link y IEEE 1394 (firewire).

Técnicas de 'Scanning'

Los datos del sensor son leídos o desplegados por diferentes métodos. El método que se utilice va a depender de la aplicación.

Entrelazado: Este método se desarrolló para enviar señales de televisión y proveer una imagen más uniforme. Consiste en leer primero las líneas impares (1, 3, 5,...) durante el primer tiempo de medio cuadro y luego las pares (0, 2, 4,...) durante el segundo tiempo de media cuadro y luego integrarlas. Desafortunadamente, si ocurre algún movimiento en la escena entre barridas, el cuadro de salida aparecerá borroso.

Progresivo: Este método lee cada línea en el orden natural en que aparece en la imagen (0, 1, 2,...). Esta técnica es útil para aplicaciones de detección de movimiento. Debido a que evita el procesamiento necesario para sistemas de televisión, es popular para aplicaciones que utilicen una computadora.

2.2.4. Selección de la imagen

Resolución:

Se seleccionó la mayor resolución posible de la cámara, es decir 640x480.

Formato:

El programa es el encargado de la adquisición de imágenes y no el software incluido con la cámara. Debido a esto, el programa maneja los valores digitales de la imagen y no utiliza un formato tradicional para archivos de imagen. La información de la imagen se almacena en un arreglo. Si la imagen está en escala de grises, el arreglo será bidimensional y tendrá el tamaño $M \times N$, donde M es el alto de la imagen y N es el ancho de la imagen, medido en píxeles. Si la imagen está a colores, el arreglo será tridimensional y tendrá el tamaño $M \times N \times 3$. En las imágenes a color cada píxel está representado por 3 valores, uno por cada componente de su color: rojo, verde y azul.

El arreglo contiene datos de tipo uint8, los cuales son enteros de 8 bits sin signo. Estos datos pueden tener cualquier valor entero entre 0 y 255.

2.3. Iluminación

Una buena iluminación disminuye la cantidad de procesamiento necesaria. Para formar imágenes de utilidad se debe proveer luz de una forma que resalte las características deseadas mientras minimice las características indeseadas. Además, se debe procurar controlar la luz ambiental ya que ésta pueda cambiar en cualquier instante.

A continuación se describen algunos aspectos a tomar en cuenta en el diseño de la iluminación.

Tipos de Reflexión:

Los objetos reflejan la luz de dos formas:

Especular: La luz de cada rayo incidente es reflejada en una sola dirección. Las reflexiones especulares son brillantes, pero pueden causar una saturación en la cámara. Además, no son confiables ya que un pequeño cambio del ángulo entre el iluminador, el objeto y el lente puede causar que desaparezca la reflexión especular completamente.

Difusa: La luz de cada rayo incidente se refleja desde varios ángulos. Las reflexiones difusas son tenues pero estables.

Técnicas de Iluminación:

En una imagen es importante el contraste. El contraste es una medida de la diferencia entre dos regiones de una imagen. La iluminación debe ser diseñada para enfatizar esa diferencia. La iluminación dependerá del objeto y de los rasgos que se desean resaltar. Es importante conocer varios parámetros para seleccionar la técnica de iluminación adecuada.

Ángulo Sólido de Iluminación: El ángulo sólido de iluminación es el área de una esfera unitaria centrada en el objeto que ocupa la iluminación. El ángulo sólido se mide en estereorradianes, con 4π estereorradianes en una esfera completa. La iluminación de un ángulo sólido pequeño se denomina puntiforme; y de un ángulo sólido grande, difusa.

Iluminación puntiforme: La iluminación puntiforme (point-like) es fácil de implementar ya que los iluminadores son pequeños y pueden ser puestos a una distancia del objeto. Algunos ejemplos de iluminadores puntiformes son las lámparas incandescentes, haces de fibra óptica, y diodos emisores de luz (LEDs).

La iluminación puntiforme ofrece una alta intensidad y eficiencia. Es buena para crear bordes finos en la imagen, formar sombras y acentuar rasgos de la superficie. No es recomendada para superficies especulares, pero sí para superficies difusas.

Iluminación difusa: Debido a que una iluminación difusa debe cubrir una gran parte del objeto, es más complicada de implementar que una iluminación puntiforme. En la iluminación difusa se minimiza la textura de la superficie y la sensibilidad a los ángulos del objeto. También es causa de bordes borrosos en las imágenes. Se recomienda para ambas superficies, especulares y difusas.

Dirección de Iluminación

Campo brillante: En esta iluminación, la luz entra perpendicular o casi perpendicular a la superficie. La iluminación normal de una casa es de campo brillante.

Un caso especial de este tipo de iluminación es la iluminación coaxial. Aquí el objeto se ilumina precisamente desde la dirección del lente. Esto requiere de un divisor de haz adentro o en frente del lente. La iluminación coaxial se utiliza para inspeccionar los rasgos de superficies planas especulares, para tomar una imagen de rasgos profundos y para eliminar sombras.

Campo Oscuro: Si el objeto se ilumina desde un sitio aproximadamente paralelo a la superficie, la iluminación es denominada campo oscuro. La textura y otras elevaciones sobre la superficie se miran con claridad, mientras que la mayor parte del objeto aparece oscuro.

Luz trasera: En esta iluminación, el iluminador está atrás del objeto. Puede ser difusa o puntiforme. Existen dos usos para la luz trasera (backlight), la primera es la observación de objetos traslúcidos y la segunda es para la formación de siluetas de objetos opacos.

Color de Iluminación: La mayoría de aplicaciones utiliza luz sin filtrar. Pero en algunos casos, la iluminación con un solo color puede proveer un mejor contraste. Un espectro reducido provee una mejor resolución, pero reduce la cantidad de iluminación.

Técnicas Especiales

Polarización: Es la orientación de las ondas que comprenden la iluminación. Se utiliza un polarizador de luz y un polarizador sobre el lente (donde se le llama analizador) para eliminar la luz de reflexión especular.

Iluminación estructurada: Significa proyectar desde aproximadamente 45 grados un patrón de luces sobre la superficie del objeto. Se utiliza para obtener información tridimensional del objeto. La altura se puede determinar de la forma del patrón dispersada desde el objeto y la imagen.

2.3.1. Selección de iluminación

La cámara seleccionada tiene poca sensibilidad a los cambios de luz y le permite al sistema trabajar bajo diferentes condiciones de iluminación sin mayor alteración en su operación. No se necesitó de una iluminación muy cercana que causara incomodidad al usuario.

La iluminación puede ser luz de día, fluorescente o incandescente como la que se encuentra generalmente en las habitaciones. La iluminación usada fue de tipo incandescente centrada en el techo de una habitación.

Mientras el usuario esté utilizando el sistema no se deben variar las condiciones iniciales de iluminación. Se debe procurar que la iluminación sea uniforme sobre la cara del usuario.

2.4. Monitor

El teclado virtual, como se mencionó en el capítulo uno, consiste de nueve teclas distribuidas en la pantalla de un monitor. El tamaño del teclado varía conforme el tamaño del monitor. Entre mayor sea el tamaño del teclado y la distancia entre teclas mayor será la precisión para distinguir la orientación de la mirada. La precisión también aumentará con la disminución de distancia entre el teclado virtual y el usuario.

2.5. Lenguaje de programación

Al considerar el lenguaje de programación a utilizar se debe evaluar las librerías que incluye y las funciones propias del lenguaje, ya que estas herramientas influirán en el desarrollo del algoritmo.

Además, debe ser compatible con el sistema operativo y la cámara, con funciones que controlen propiedades de la última.

Utilizando el lenguaje de programación se debe poder crear un programa que construya el interfaz de usuario, adquiera y procese imágenes y tome las decisiones necesarias. El corazón del proyecto es el programa, el programa controla la operación del sistema.

Para satisfacer los requerimientos del programa, el lenguaje de programación debe proveerle al programador la capacidad de cumplir las siguientes funciones:

- Construcción de un entorno gráfico
- Adquisición de imágenes
- Procesamiento de imágenes
- Escritura a archivos

Construcción de un entorno gráfico:

El entorno gráfico es de vital importancia ya que únicamente por medio de la vista se comunica el usuario. El entorno gráfico tomará la forma de un teclado virtual.

Adquisición de imágenes:

Se deben adquirir imágenes de forma periódica de la cámara.

Procesamiento de imágenes:

Las imágenes capturadas deben ser filtradas para detectar cualquier movimiento ocular.

Escritura a archivos.

Es necesario transcribir a un documento de Word lo que escriba de forma virtual el usuario, por tanto se deben utilizar funciones que manipulen este tipo de archivo.

2.5.1. Selección del lenguaje de programación

El lenguaje de programación que se seleccionó fue MATLAB. MATLAB es el nombre abreviado de "MATrix LABoratory". Es un lenguaje de programación para realizar cálculos numéricos con vectores y matrices. Como caso particular puede también trabajar con números escalares, con cadenas de caracteres y con otras estructuras de información más complejas.

La razón principal de seleccionar MATLAB reside en las herramientas que proporciona para la adquisición y el procesamiento de imágenes. Éstas se encuentran en el "image acquisition toolbox" e "image processing toolbox" respectivamente.

Además, para crear el entorno gráfico se puede utilizar el GUIDE de MATLAB. Esta provee herramientas para crear GUIs, 'Graphical User Interface'. Permitiendo diseñar la forma del entorno gráfico así como asociar funciones a los diferentes elementos del GUI. MATLAB también incluye funciones para la manipulación de archivos.

2.6. Montaje de la cámara

Un adecuado montaje de la parte óptica del sistema puede mejorar y mantener el rendimiento del mismo. Entre las características deseables del montaje están:

Protección: El lente debe ver a través de una ventana fácilmente limpiable. Se debe prevenir que las superficies ópticas se contaminen y que entre luz indeseada.

Estabilidad: La estabilidad mecánica es importante, ya que la óptica debe ver la escena apropiada, desde el ángulo adecuado y con los ajustes necesarios de enfoque. La óptica del sistema debe poder ser estable por largos períodos de tiempo.

Ajuste: Se debe poder ajustar el sistema fácilmente a la posición y a las características físicas del usuario y del monitor.

2.6.1. Selección del montaje de la cámara

En la implementación del sistema se utilizaron dos diferentes configuraciones para el montaje de la cámara. La primera utilizó la cámara montada encima de un casco y la segunda montada encima de la computadora. Se presentarán los resultados de ambas configuraciones en el capítulo cuatro.

En la primera configuración, la cámara estaba posicionada enfrente de uno de los ojos a una distancia de aproximadamente 5 centímetros y el casco utilizado podía ajustarse al tamaño de la cabeza del usuario. Con esta configuración se obtiene una mayor resolución del ojo, pero la cámara causa cierta obstrucción a la vista de ese ojo.

2.7. Relación usuario-monitor-cámara

El usuario se debe centrar enfrente de la pantalla de la computadora para poder observar de mejor forma el teclado virtual y para que las diferencias en la orientación de la mirada sean más marcadas.

A continuación se encuentran dos figuras (Fig. 8 y 9) del Sistema de Comunicación Visual, utilizando las dos configuraciones de montaje de la cámara mencionadas.

Figura 8. Primera configuración

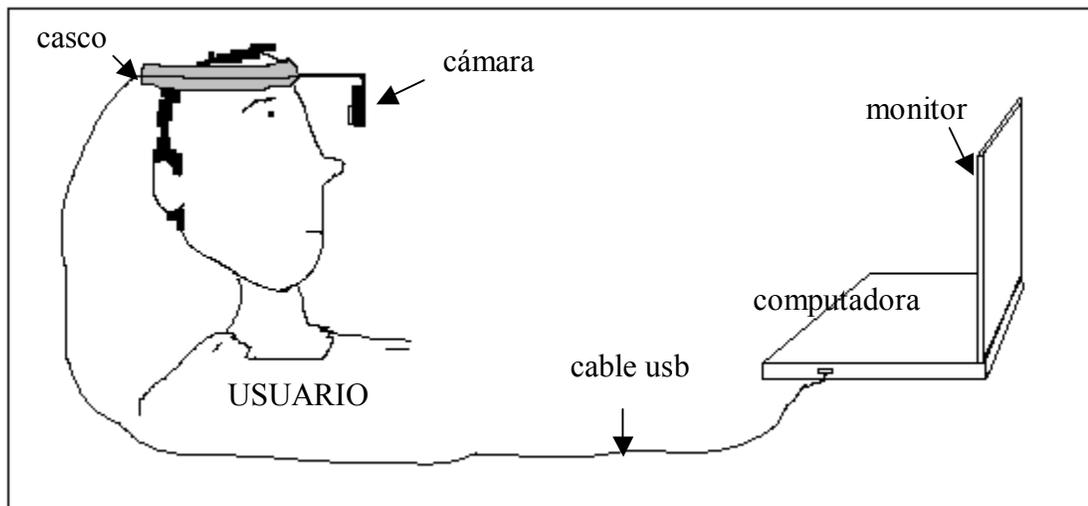
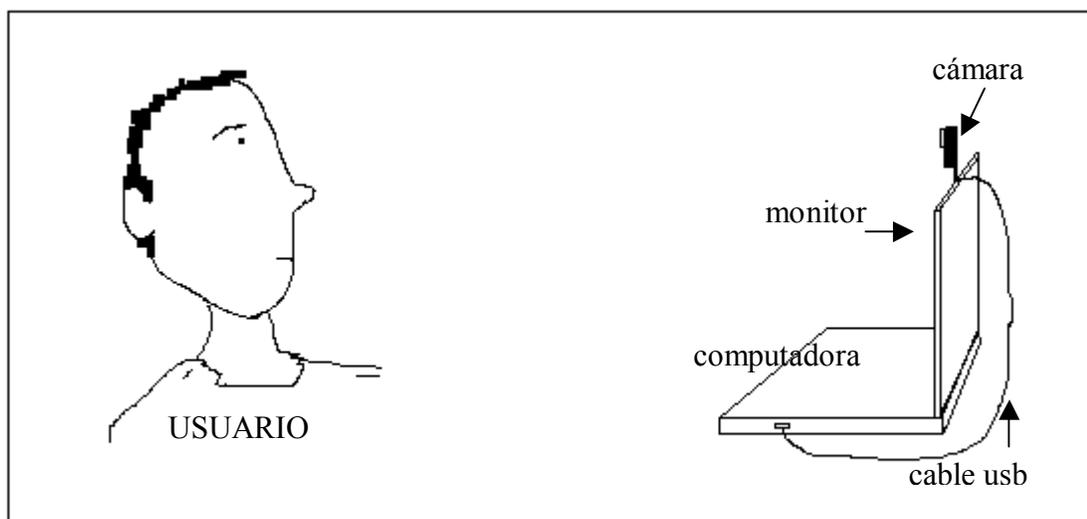


Figura 9. Segunda configuración



3. FILTROS 2D Y PROGRAMA

Una imagen es una percepción en dos dimensiones (2D) de una escena tridimensional. El proyecto utiliza la computadora para capturar e interpretar información visual, en la forma de imágenes. Por medio de imágenes, se deducirán las propiedades del entorno y se tomarán decisiones basados en ellas.

Es obligatorio transformar estas imágenes para remover ruido y resaltar rasgos de valor a la aplicación. Es decir, se debe realizar un procesamiento de imágenes. A la función que transforma la imagen se le denomina filtro 2D.

Este capítulo empieza con teoría de algunos filtros de interés, luego explica el procedimiento general que ejecuta el programa, y finaliza dando a conocer las diferentes versiones de éste.

Dentro del marco del capítulo se señalan las funciones implementadas, incluyendo los filtros utilizados.

El código de los distintos programas se encuentra en el apéndice.

3.1. Clasificaciones de filtros

Existen tres clasificaciones de filtros:

1. Por su linealidad

Los filtros lineales, también conocidos como filtros de convolución, se pueden representar en álgebra lineal como una multiplicación de matrices. Los filtros no-lineales son todos los demás filtros que no se pueden representar usando una formulación de matrices.

2. Por su respuesta en frecuencia

Se utilizan filtros pasabajos para remover ruido o suavizar imágenes. Los filtros pasaaltos responden a los cambios abruptos de la intensidad de luz en una imagen y se pueden usar para mejorar el detalle de una imagen.

3. Filtros de frecuencia o espaciales

Los filtros espaciales alteran los valores de píxeles con respecto a las variaciones de la intensidad de luz en su vecindad, mientras que los filtros de frecuencia operan en el dominio de la frecuencia sobre imágenes que han sido transformados por Fourier. Después de la operación de filtrado, se aplica la transformada inversa de Fourier.

3.2. Convolución y correlación

Dos operaciones matemáticas elementales al filtrado de imágenes son la convolución y la correlación.

Convolución

El filtrado lineal de una imagen se realiza a través de esta operación.

La idea básica de la convolución consiste en una ventana de tamaño finito que es barrida sobre la imagen. La ventana es una matriz de pesos que se denomina "kernel" de convolución o filtro. El filtro se rota 180° alrededor de su origen o elemento central, luego el elemento central se sitúa sobre el píxel cuyo valor se desea filtrar. El valor de píxel resultante es la suma de píxeles dentro de la ventana multiplicados por su peso. El peso de cada píxel es el valor de píxel de la ventana que cae sobre él. Si el filtro $h[j, k]$ es cero afuera de la ventana rectangular $\{j=0,1,\dots,J-1; k=0,1,\dots,K-1\}$, entonces la convolución del filtro con una imagen 'a' se puede escribir como una suma finita de productos de la siguiente forma:

Ecuación 2

$$c[m, n] = a[m, n] * h[m, n] = \sum_{j=0}^{j-1} \sum_{k=0}^{k-1} h[j, k] a[m - j, n - k]$$

Correlación

La correlación de dos funciones reales es una operación de similares características a la convolución con la salvedad de que no se gira alrededor del origen los valores del filtro o "kernel" de correlación. La expresión matemática para esta operación es:

Ecuación 3

$$c[m, n] = \sum_{j=0}^{j-1} \sum_{k=0}^{k-1} h[j, k] a[m + j, n + k]$$

Al igual que la convolución, la correlación es una operación básica del procesamiento de imágenes digitales.

La correlación se utiliza en los procesos de reconocimiento de patrones. El aplicarle correlación a una imagen, donde posiblemente se encuentre una característica a localizar, con una imagen de esa característica, producirá máximos locales (valores elevados de píxel) en posiciones candidatos. Por tanto, disponer de algoritmos que calculen de una forma eficiente estas operaciones es de gran interés.

Coefficiente de correlación

El coeficiente de correlación es una medida de la fuerza de la relación entre dos variables. Los valores del coeficiente de correlación varían en el rango de -1 a 1, indicando una correlación perfecta negativa en -1, ausencia de correlación en 0, y una correlación perfecta positiva en +1. Se puede estimar qué tan parecida es una imagen a otra con el coeficiente de correlación. El coeficiente de correlación de una imagen A con una imagen B, donde ambas imágenes son matrices de igual tamaño, se calcula con:

Ecuación 4

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}}$$

Donde \bar{A} y \bar{B} son las medias de sus respectivas matrices.

3.3. Filtros de interés

A continuación se describirán algunos filtros de interés para la implementación del Sistema de Comunicación Visual.

Umbralización

La umbralización nos permite separar a los objetos bajo consideración del fondo. La umbralización convierte la imagen en una imagen binaria, con valores de píxel de 0 y 1. Este proceso funciona estableciendo a 1 a todos los píxeles cuyos valores caen dentro de cierto rango, llamado intervalo de umbralización, y poniendo a 0 todos los demás valores de píxel.

Detección de bordes

La detección de bordes se refiere al proceso de identificar y localizar discontinuidades agudas en una imagen. Las discontinuidades son cambios abruptos en la intensidad de píxel, lo cual caracteriza a los bordes en una escena. Para detectar los bordes se puede convolucionar la imagen con un filtro 2D, que esté construido para ser sensible a grandes gradientes donde retorna un valor de 1, mientras retorna valores de 0 para las regiones uniformes.

Reconocimiento de patrones

El reconocimiento de patrones localiza regiones en una imagen que se parezcan a un modelo predeterminado. En el reconocimiento de patrones existe un punteo que muestra qué tan parecido es el modelo a la región encontrada.

3.4. Programa

Para obtener una vista panorámica del programa, se resume el programa completo en un algoritmo denominado SCV (Sistema de Comunicación Visual). Los subalgoritmos del programa se encuentran en cursiva.

Después del algoritmo general se define el papel de cada subalgoritmo dentro del programa, y si se considera necesario se presenta la estructura de éste.

Algoritmo: SCV

Variables:

i : índice, número de índice actual.

IR_i : imagen de referencia número i .

T_i : Tecla i , objeto para manipular propiedades de las teclas de la pantalla.

ia : índice anterior, número de índice anterior.

IN : imagen nueva.

m : mensaje escrito.

c : cuenta, numero que indica cuantas veces se ha observado de manera seguida la misma tecla.

$L_{i,c}$: Carácter o palabra c de la tecla i .

Inicialización

$IR_{(1:10)} \leftarrow \text{Calibración}(T_{(1:9)})$

Mientras $L_{i,c} \neq \text{"salir"}$ haga

$IN \leftarrow \text{Adquisición}$

$IN \leftarrow \text{Procesamiento}(IN)$

$i \leftarrow \text{Comparación}(IN, IR_{(1:10)})$

$c, m, L_{i,c} \leftarrow \text{Acciones}(i, m, c, T_{(1:9)})$

$ia \leftarrow i$

Fin Mientras

Finalización

SUBALGORITMOS:

1) Inicialización

En la inicialización se declaran variables a utilizar, se asignan valores a éstas, y tomando en cuenta el tamaño de la pantalla, se despliega el teclado virtual. También es aquí donde se crea el objeto de entrada de la cámara y se fijan algunas propiedades de ésta.

2) Calibración

El proceso de calibración sirve para obtener las diez imágenes de referencia del ojo o segmento de él. Durante este proceso se le indica al usuario por medio de mensajes en la pantalla, instrucciones a seguir respecto a la orientación de su mirada. Como se puede observar en el subalgoritmo *calibración*, las imágenes de referencia también pasan por una etapa de procesamiento.

Subalgoritmo: Calibración

Variables:

IR_i : imagen de referencia i , T_i : Tecla i ,

Escribir a pantalla ("Por favor observe la tecla roja")

Para $i \leftarrow 1$ hasta 9 haga

 color de fondo de $T_i \leftarrow$ rojo

 pausa de 4 segundos

$IR_i \leftarrow$ *Adquisición*

$IR_i \leftarrow$ *Procesamiento* (IR_i)

color de fondo de $T_i \leftarrow$ azul
Fin Para
Escribir a pantalla ("Cierre los ojos por 5 segundos")
pausa de 4 segundos
 $IR_{10} \leftarrow$ *Adquisición*
 $IR_{10} \leftarrow$ *Procesamiento* (IR_{10})
Quitar instrucciones de la pantalla

Retornar ($IR_{(1:10)}$)

3) Adquisición

Esta función nos devuelve una imagen del ojo recién capturada por la cámara. La adquisición de una foto se utiliza primero en el proceso de calibración para obtener las imágenes de referencia, y luego se realiza periódicamente para obtener la dirección de la mirada actual.

4) Procesamiento

Se procesa la imagen obtenida para resaltar los rasgos más importantes y eliminar rasgos sin importancia. Esto facilita su posterior análisis. Debido a que este subalgoritmo, igual que el de comparación, varía considerablemente en las distintas versiones, se profundizará en estos en la sección de versiones.

5) Comparación

Para determinar la dirección de la mirada se compara la imagen actual, después de que ha sido procesada, con las imágenes de referencia procesadas. Esta función retorna un índice de 1 a 10.

Un índice i , de 1 a 9, indica que el usuario está observando la tecla virtual i ; y un índice de 10, indica que el usuario tiene el ojo cerrado.

6) Acciones

La información sobre la dirección de la mirada en la forma del índice i se utiliza para ejecutar ciertas acciones. Entre éstas, la retroalimentación al usuario por medio de cambios en el teclado virtual y la reproducción de sonido, y la escritura del mensaje tecleado a archivo.

Para mejor comprensión, revisemos el funcionamiento del sistema y las acciones que éste comprende: Las teclas son de color verde, y cuando el usuario observa una de las teclas, ésta se torna color rojo. Para poder usar todos los caracteres del teclado, cada tecla contiene cinco caracteres. Si la persona sigue observando fijamente la misma tecla, esta cambiará su contenido, es decir la "etiqueta" de la tecla cambiará al siguiente carácter. Cuando el usuario se haya ubicado en el carácter deseado deberá cerrar los ojos. Al hacer esto se escuchará un sonido y el carácter seleccionado se escribirá a un documento junto con cualquier carácter escrito anteriormente, además de aparecer en un cuadro de texto en la pantalla. Una de las teclas también contiene la palabra "salir" para cuando el usuario se desee salir del programa, así como la palabra "supr" para borrar caracteres indeseados.

Subalgoritmo: Acciones

Variables:

i : índice, T_i : Tecla i , i_a : índice anterior, m : mensaje escrito, c : cuenta, $L_{i,c}$: Carácter o palabra c de la tecla i ;

```

Si  $i \neq 10$  entonces
    rojo  $\leftarrow$  color de fondo de  $T_i$ 
    Si  $i == i_a$ 
        Si  $c == 5$ 
             $c \leftarrow 1$ 
        De otra manera
             $c \leftarrow c + 1$ 
        Fin Si
    de otra manera
         $c \leftarrow 1$ 
    Fin Si
    etiqueta de  $T_i \leftarrow L_{i,c}$ 
De otra manera
    Si  $L \neq \text{"salir"}$ 
        Si  $L \neq \text{"supr"}$ 
             $m \leftarrow$  concatenar ( $m, L_{i,c}$ )
        De otra manera
             $m \leftarrow$   $m$  menos 1 carácter
        Fin Si
    reproducción de un sonido corto
    escribir a pantalla ( $m$ )
    escribir a archivo ( $m$ )
Fin Si

Retornar ( $c, m, L_{i,c}$ )

```

7) Finalización

Se limpian las variables utilizadas, se elimina el objeto de entrada de la cámara y se remueve la figura del teclado virtual de la pantalla.

3.4.1. Versiones

Aunque en el proceso de mejorar el proyecto se crearon muchas versiones del programa, aquí solo se presentan las de mayor éxito.

Las versiones del programa se diferencian en la implementación de algunos de los subalgoritmos mencionados anteriormente. Cada versión fue específicamente diseñada para una de las configuraciones de los elementos: monitor y cámara, que se presentaron en la sección 2.7. A continuación se nombran las distintas versiones, se describe su funcionamiento y se señalan las diferencias entre ellas.

Versión A:

La versión A fue diseñada para la primera configuración de elementos. Su etapa de procesamiento de imagen consiste en la conversión de la imagen original de tipo RGB a una imagen en escala de grises, a esta imagen de intensidades se le resta su promedio.

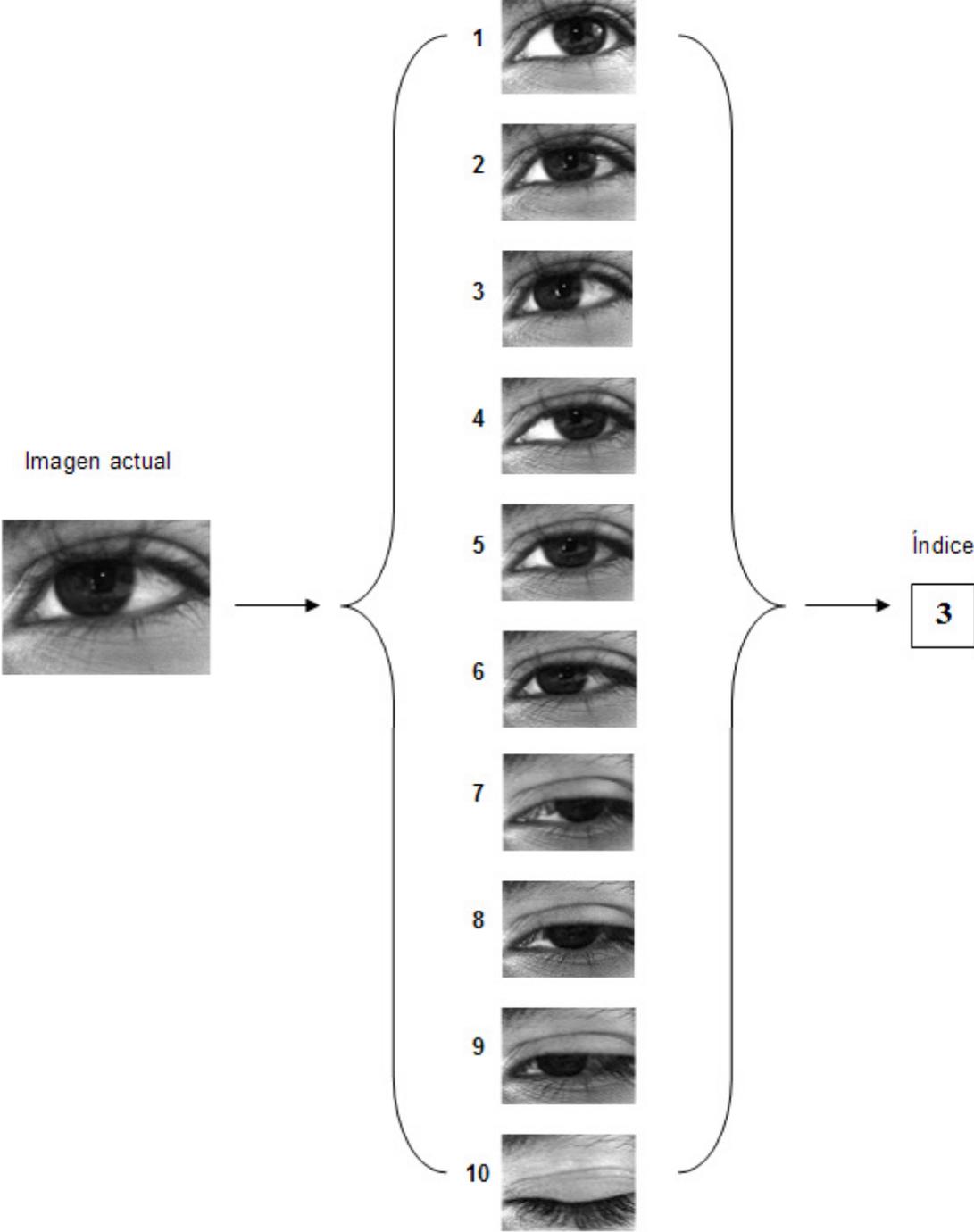
La conversión de RGB a escala de grises se efectúa debido a que la información sobre el color de la imagen no se utiliza posteriormente. Además, ésta conversión disminuye el espacio que ocupa la imagen y facilita su manipulación. Al restarle su promedio a la imagen, eliminamos el contenido de la imagen que no nos provee de información útil.

Antes de restarle el promedio a la imagen debemos cambiar el tipo de datos de uint8 a double para poder tener valores de píxel con signos positivos y negativos. El rango de los valores posibles de píxel de la imagen procesada será entonces de -255 a 255 en lugar de 0 a 255.

La etapa de comparación se realiza calculando el coeficiente de correlación de la imagen actual con cada una de las imágenes de referencia. Esta etapa devuelve el índice de la imagen de referencia con el cual se obtuvo el mayor coeficiente de correlación. Se ahorró cuantioso y valioso tiempo computacional al modificar la función de correlación corr2 de MATLAB para su uso en esta aplicación. La función modificada únicamente contiene la ecuación para calcular el coeficiente de correlación; ya no contiene la verificación de la validez de las entradas. Esta modificación no representó ningún peligro, dado que se sabía de antemano que las entradas suministradas a la función serían válidas.

La etapa de comparación se ilustra en la figura 10.

Figura 10. Etapa de comparación en la versión A



Versión B:

Esta versión fue creada para ser utilizado con la segunda configuración. Difiere en funcionamiento con la versión A en el método de presionar de forma virtual la tecla seleccionada. En lugar de parpadear, el usuario debe dirigir la mirada hacia un extremo de su campo de vista, por ejemplo mirar hacia arriba. Se seleccionó un extremo para evitar cualquier confusión con la orientación de la mirada cuando el usuario esté observando el teclado. En el proceso de calibración se selecciona la dirección de presión de tecla al tomar la imagen de referencia.

Debido que la cámara está localizada encima del monitor, la imagen que provee inicialmente es de la cara completa del usuario. En el proceso de calibración se cambia el tamaño de la imagen vista por la cámara a una región pequeña alrededor del ojo. La imagen inicial tiene un tamaño de 640x480, el tamaño de la imagen final será mucho menor.

El cambio del tamaño de la región de interés de la cámara se consigue pidiéndole a una persona que asista al usuario en el recorte de una imagen del iris de la imagen de cara completa que aparece en la pantalla, el usuario no requerirá de más asistencia después de este pequeño procedimiento. Denominaremos a esta variable de la imagen del iris recortado, después de pasar por una fase de mejoramiento, como IrisIni (iris inicial). Conociendo la posición de IrisIni se puede calcular una región prudente alrededor de él, donde se observe todo el ojo de usuario. Esta región establece el tamaño del cuadro capturado por la cámara. Luego, se prosigue con la obtención de diez imágenes del ojo, donde las primeras nueve imágenes indican la orientación de la mirada del usuario hacia una de las teclas y la décima es la imagen de presión de tecla.

Estas imágenes son mejoradas y luego, utilizando un procedimiento de reconocimiento de patrones, donde se utiliza IrisIri como patrón, se busca el iris en cada imagen del ojo. Estas imágenes del iris serán las imágenes de referencia resultantes del proceso de calibración.

La etapa de procesamiento se divide en dos fases, la primera fase consiste en mejorar la imagen del ojo y facilitar la localización del iris, ésta localización es el objetivo de la segunda fase. Llamaremos a la primera fase, Mejoramiento, y a la segunda fase, Reconocimiento de Patrón.

Los pasos de la fase de Mejoramiento son:

Paso 1: Se convierte la imagen del ojo a escala de grises.

Paso 2: Se filtra la imagen de forma lineal utilizando un kernel de correlación de 3x3, el kernel o filtro es un filtro de promedio, cada píxel resultante tiene el valor promediado de sus vecinos. Este tipo de filtro elimina el ruido causado por variaciones locales pequeñas.

Paso 3: Se complementa la imagen. La esclerótica del ojo aparecerá de un tono gris oscuro mientras que el iris tendrá un tono claro.

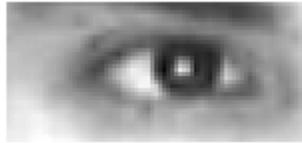
Paso 4: Se rellenan los agujeros, donde un agujero se define como un área de píxeles oscuros rodeado de píxeles claros. Esto se utiliza para emparejar el tono del interior del iris.

Paso 5: Se aumenta el contraste de la imagen.

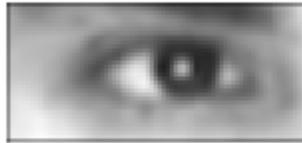
La fase de mejoramiento se utiliza en dos casos:

- Como parte de la etapa de procesamiento donde mejora la imagen capturada del ojo (Fig. 11).
- En el proceso de calibración para mejorar la imagen del iris recortado (Fig. 12). La imagen resultante es la variable IrisIri.

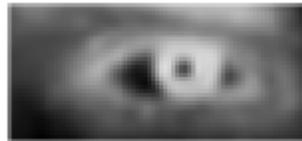
Figura 11. Pasos de mejoramiento de una imagen del ojo



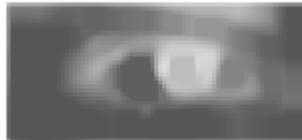
Paso 1



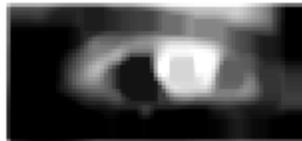
Paso 2



Paso 3

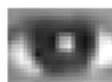


Paso 4



Paso 5

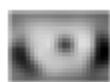
Figura 12. Pasos de mejoramiento de una imagen del iris



Paso 1



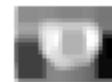
Paso 2



Paso 3



Paso 4



Paso 5

La fase de Reconocimiento de Patrón recibe como entrada una imagen patrón y la imagen donde se desea localizar el patrón, tiene como salida la sección de la imagen que tiene mayor similitud con el patrón. En este caso se utiliza la imagen de IrisIri como patrón, y la imagen del ojo, después de la fase de mejoramiento, como imagen donde se desea localizar el iris. Esta fase se realiza barriendo la imagen de IrisIri sobre la imagen del ojo y calculando el coeficiente de correlación de IrisIri con cada sección de la imagen del ojo o subimagen del mismo tamaño que IrisIri. La subimagen con el coeficiente de correlación más alto será el iris localizado y la salida de esta fase.

La etapa de comparación es igual a la de la versión A, pero recibe como entradas la imagen del iris actual (imagen de salida de la etapa de procesamiento) y las imágenes de referencia del iris (imágenes de salida de la etapa de calibración). Esta etapa tiene como salida el índice de la imagen de referencia con el cual se obtuvo el mayor coeficiente de correlación.

Para comprender mejor el funcionamiento de la versión B, se presenta un ejemplo ilustrativo de éste.

En la figura 13 se muestra el proceso de calibración, inicialmente la cámara captura una imagen de cara completa (Fig. 13.a), de la cual se recorta una imagen del iris (Fig. 13.b). Para crear la imagen IrisIri (Fig. 13.c), se mejora esta imagen del iris. Además, la posición de la imagen del iris nos permite establecer una región de captura de la cámara que solo contenga el ojo del usuario (Fig. 13.d).

Utilizando IrisIni para la fase de reconocimiento de patrones, se procesan 10 imágenes del usuario con distintas orientaciones de la mirada: las primeras nueve corresponden a las nueve teclas del teclado y la décima a la dirección de presión de tecla. El resultado de este procesamiento constituye las imágenes de referencia del iris (Fig. 13.e).

En la figura 14 se muestra la detección de la orientación actual de la mirada. Después de que se ha capturado una nueva imagen (Fig. 14.a), se extrae la imagen de su iris (Fig. 14.b) por medio de la etapa de procesamiento. Luego, se realiza la etapa de comparación con las diez imágenes de referencia (Fig. 14.c) y se obtiene el índice (Fig. 14.d) de la imagen de referencia con el mayor coeficiente de correlación. En la figura, el usuario observa la dirección de presión de tecla.

Figura 13. Proceso de calibración en la versión B

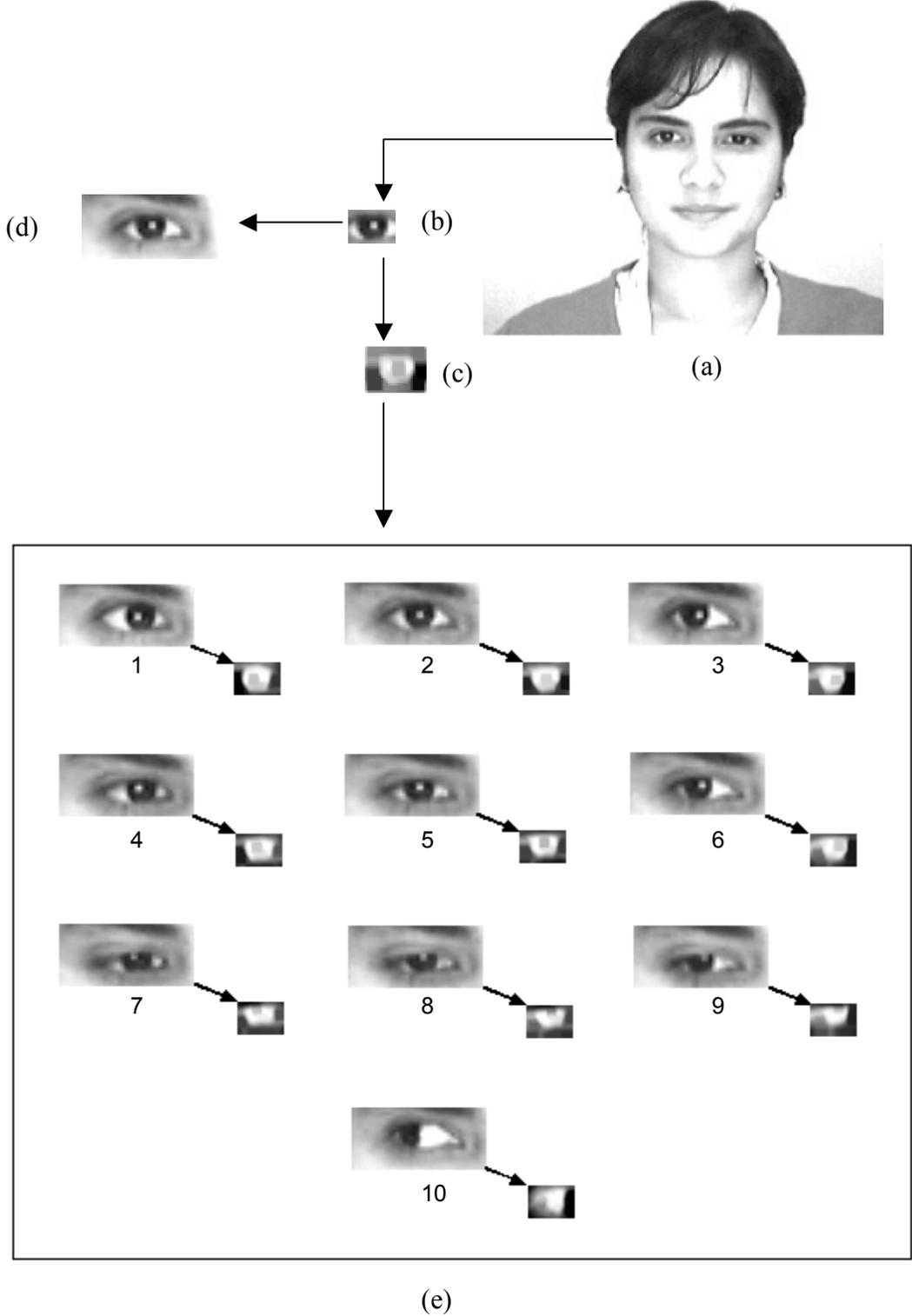
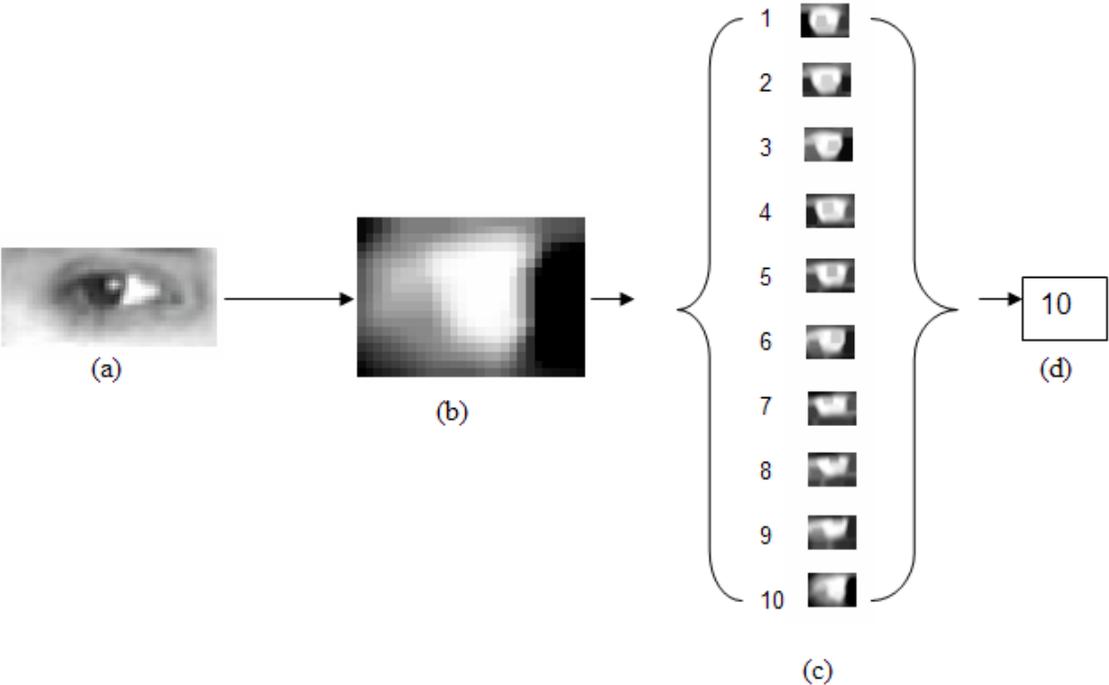


Figura 14. Detección de la orientación actual de la mirada en la versión B



4. EVALUACIÓN

En este capítulo se presenta una evaluación de las distintas versiones del sistema, para la evaluación fue necesario definir un criterio de desempeño y basados en éste crear un programa de evaluación. Como parte de la evaluación se incluyen algunas experiencias sobre el funcionamiento de las diferentes versiones y un análisis económico.

4.1 Criterio de desempeño

Previo a crear el programa de evaluación se debía seleccionar unos indicadores de desempeño. Se decidió que los indicadores seleccionados debían basarse en la velocidad y precisión del sistema. Además, sería necesario describir estadísticamente las medidas observadas de estos indicadores.

4.1.1. Selección de indicadores de desempeño

Para indicar la velocidad se escogió medir el tiempo que tarda el usuario en escribir una palabra. Para indicar la precisión con que el sistema decide cual es la tecla deseada se eligió usar el valor del coeficiente de correlación que utiliza el programa.

4.1.2. Estadística descriptiva

La estadística descriptiva resume las medidas observadas de un sistema. Al resumir un conjunto de observaciones se transmite una gran cantidad de información de la forma más sencilla posible.

El resumen generalmente contiene:

1. Una estadística que muestra cómo los valores de un conjunto de observaciones son similares. Esta estadística se denomina de tendencia central y nos indica el valor alrededor del cual se agrupan los datos. Las tres medidas más usuales de tendencia central son la media, la mediana, y la moda.

2. Una estadística que muestra cómo los valores difieren. Este tipo de estadística se denomina de variabilidad o dispersión y nos indica si los valores están próximos entre sí o por el contrario están muy dispersos. Ejemplos de este tipo de estadística son el rango, la varianza y la desviación estándar.

A continuación se definen las estadísticas descriptivas utilizadas en la evaluación del proyecto:

Media:

La media aritmética o promedio de un conjunto de números es la suma de todos los miembros del conjunto dividido por el número de elementos en el conjunto. La media aritmética es bastante influenciada por valores extremos.

Si denotamos el conjunto de datos por $X=\{x_1, x_2, \dots, x_n\}$ entonces la media aritmética es:

Ecuación 5

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i .$$

Mediana:

Si tenemos n valores x_1, x_2, \dots, x_n habiendo sido ordenados de forma creciente, entonces se define la mediana como el valor que deja a cada lado (por encima y por debajo) la mitad de los valores de la muestra.

Matemáticamente la mediana toma por valor:

Ecuación 6

$$x_{\frac{n+1}{2}}, \text{ si } n \text{ es impar}$$

y

Ecuación 7

$$\frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2}, \text{ si } n \text{ es par}$$

Como medida descriptiva, tiene la ventaja de no estar afectada por las observaciones extremas, ya que no depende de los valores que toma la variable, sino del orden de las mismas.

Desviación estándar:

La desviación estándar de un conjunto de datos es una medida de cuánto se desvían los datos de su media. Una desviación estándar elevada indica que los datos se encuentran alejados de la media y una desviación estándar reducida significa que se encuentran cerca de la media. Se calcula con:

Ecuación 8

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$$

4.2 Programa de evaluación

La creación de un programa de evaluación fue necesaria para obtener información que indicara el desempeño de la versión siendo evaluada. El programa de evaluación fue utilizado durante el proceso de experimentación, facilitando la observación del rendimiento de las diferentes versiones. Estas observaciones dieron lugar al refinamiento y desarrollo del proyecto.

El programa de evaluación de cada versión funciona de igual manera que la versión a evaluar en las etapas de inicialización, calibración, adquisición procesamiento y comparación. Difiere únicamente en algunas acciones que realiza como el almacenamiento de información y el control del teclado.

El programa de evaluación despliega en las primeras 8 teclas, 8 letras aleatorias del alfabeto, y la 9ª. tecla es la tecla de suprimir. A diferencia del programa normal, cada tecla contiene solo una letra. En la pantalla aparece una instrucción que pide al usuario escribir determinada palabra.

La palabra está formada por un conjunto aleatorio de 5 letras de las 8 posibles en el teclado. Cuando haya escrito correctamente la palabra, el usuario escuchará un sonido y se refrescará el teclado virtual con otro conjunto de 8 letras y una nueva palabra. Este proceso se repetirá 20 veces. Al terminar de escribir las 20 palabras, el programa se cerrará y se podrá abrir un archivo donde se encuentran los datos recolectados por el programa de evaluación. Este archivo estará guardado en la misma carpeta que la versión de prueba, y su nombre contiene la fecha y hora de su creación.

Se diseñó el funcionamiento del programa de evaluación de esta manera para evitar que el evaluador se acostumbrara a la posición de determinadas letras y a escribir palabras que le fueran fáciles. Sin esta aleatoriedad¹, la evaluación no sería realmente objetiva debido a la experiencia adquirida por parte del evaluador. El tamaño de la palabra promedio es de 5 letras², y por eso se instruye al usuario escribir palabras de este tamaño. Además, uno de los datos buscados es el tiempo que le toma al usuario escribir una palabra.

Los datos recolectados y almacenados en un archivo son:

- Un vector de todos los coeficientes de correlación calculados.
- La media de los coeficientes de correlación.
- La mediana de los coeficientes de correlación.
- La desviación estándar de los coeficientes de correlación.
- Un vector del tiempo por palabra de todas las palabras.
- El tiempo promedio por palabra.
- La mediana del vector de tiempos.
- La desviación estándar del vector de tiempos.

¹ La aleatoriedad se obtuvo con el programa RandLetras presentado en el apéndice.

² Este dato se obtuvo con la creación del programa LetrasPalabrasProm que calcula el tamaño promedio de una palabra dentro de un texto voluminoso. Se encuentra en el apéndice.

Además, el programa de evaluación reduce el tamaño del teclado virtual, que normalmente ocupa toda la pantalla, para poner a la vista algunas imágenes y datos. Esta ventana de información se encuentra en el lateral derecho del teclado virtual. Las imágenes desplegadas consisten en la imagen actual y la imagen de referencia que se considera de mayor parecido. Esto permite observar la calidad de ambas imágenes y realizar una comparación de éstas. Los datos constan del coeficiente de correlación y el índice de la tecla seleccionada. El coeficiente de correlación nos indica la fuerza con que las dos imágenes se parecen. Si el índice de la tecla seleccionada no era la deseada, se puede deducir de las imágenes la causa del error. La ventana de información es actualizada con cada nueva imagen capturada.

4.3 Desempeño de las distintas versiones

En ambas versiones se pudo apreciar lo importante de la etapa de calibración, una buena calibración asegura un mejor rendimiento del sistema. Utilizando los programas de evaluación se recopilaban las estadísticas de 20 pruebas de cada versión y se realizó un promedio de éstas. Las estadísticas de los coeficientes de correlación (Tabla 2) muestran que la versión B obtiene un mayor parecido entre sus imágenes, aunque su desviación estándar es mayor.

Las estadísticas de tiempo por palabra (tabla 3) permiten ver que se obtiene una mayor velocidad de escritura con la versión A.

Tabla II. Estadísticas de los coeficientes de correlación

| | Versión A | Versión B |
|----------------------------|------------------|------------------|
| Media | 0.9189 | 0.9319 |
| Mediana | 0.9264 | 0.9532 |
| Desviación Estándar | 0.0384 | 0.063 |

Tabla III. Estadísticas de tiempo por palabra (segundos)

| | Versión A | Versión B |
|----------------------------|------------------|------------------|
| Media | 25.0994 | 30.6826 |
| Mediana | 23.164 | 28.321 |
| Desviación Estándar | 6.3582 | 8.0302 |

Se experimentó mayor comodidad con la versión B, ya que éste no utiliza un casco para el montaje de la cámara y no obstruye la vista. Sin embargo, la versión A es menos sensible al movimiento del usuario.

4.4 Análisis económico

Para la implementación del proyecto se utilizó como recursos materiales únicamente una cámara digital y el soporte en forma de casco para poder montar la cámara en la posición necesaria para capturar las imágenes del ojo del usuario. A continuación se detallan los costos materiales del proyecto.

Tabla IV. Costos materiales

| <i>Descripción</i> | <i>Costo</i> |
|---------------------------|---------------------|
| 1 Cámara web | Q200.00 |
| 1 Casco | Q40.00 |
| Total | Q240.00 |

CONCLUSIONES

- El tecleo ocular es y ha sido un método de comunicación para personas discapacitadas.
- El aumento del tamaño de un elemento del teclado virtual aumenta la precisión de selección.
- La utilización de una cámara web reduce el costo de implementar un sistema de tecleo ocular.
- Es necesario el filtrado de imágenes para resaltar rasgos de valor a la aplicación.
- La configuración del sistema con la cámara montada en la cabeza obstruye la vista.
- La versión A del sistema permite escribir con mayor velocidad que la versión B.

RECOMENDACIONES

1. Investigar cuántas personas en Guatemala y/o en otros países necesitan o se beneficiarían de un sistema de tecleo ocular.
2. Estudiar el potencial de un usuario para usar exitosamente el Sistema de Comunicación Visual basado en su diagnóstico médico.
3. Realizar pruebas del Sistema de Comunicación Visual con una mayor cantidad de personas por un período de tiempo más extenso. Utilizar la información recopilada para observar la curva de aprendizaje del usuario, es decir si mejora su rendimiento con el tiempo.
4. Integrar a la versión B del Sistema de Comunicación Visual un rastreo automático del ojo en la etapa de calibración.
5. Incorporar a ambas versiones recalibraciones automáticas basadas en la posición inicial del usuario con respecto al monitor y la cámara y su posición actual.
6. Estudiar la posibilidad de crear un mouse controlado por la mirada.

BIBLIOGRAFÍA

1. Aulikki H. y otros. **Design issues of iDict: a gaze assisted translation aid**. Proceedings of Eye Tracking Research and Applications. Florida, E.E.U.U. 9-14. 2000.
2. Bates, R., 2002. **Computer input device selection methodology for users with high-level spinal cord injuries**. Proceedings of the first Cambridge Workshop on Universal Access and Assistive Technology (CWUAAT). Trinity Hall, University of Cambridge. Inglaterra. 2002.
3. Blanco, Y. y otros. **Síntesis visual del habla**. Departamento de Ingeniería Eléctrica y Electrónica. Universidad Pública de Navarra.
4. Farid, M. y otros. **Computer display control and interaction using eye-gaze**. School of Computer Science, Queens University Belfast. Irlanda del Norte, Reino Unido.
5. Duchowski, A. **Eye-tracking methodology: Theory and practice**. Inglaterra: Springer, 2003. 251 pp
6. Jacob, R. **New human-computer interaction techniques**. Human-Computer Interaction Lab, Naval Research Laboratory, Washington D.C. E.E.U.U
7. Jacob, R. **Eye movement based interaction techniques and the elements of next generation**. Dept. of Electrical Engineering and Computer Science, Tufts University. Massachusetts, E.E.U.U.
8. Pesta, J. **Assistive, adaptive, amazing technologies**. Technos Quarterly. Vol. 3 No. 2. 1994
9. Taub, H., Schilling, D.J. **Principles Of Communications Systems**. 2ª Edición. McGraw-Hill, Singapore, 1986
10. Torres, F. et. al. **Robots y sistemas sensoriales**. 2ª. Edición. España: Pearson Educación, 2002. 480 pp.

11. Umbaugh, S. **Image enhancement through gray-scale modification.** Computer Vision and Image Processing, Prentice Hall
12. Umbaugh, S. **Morphology overview.** Computer Vision and Image Processing, Prentice Hall
13. Vieth, Y. **Optimised sensor architectures for machine vision applications.** DALSA Corporation . Ontario, Canada.

BIBLIOGRAFÍA ELECTRÓNICA

1. Gips, J. y Olivieri, P. 1996. **EagleEyes: An eye control system for persons with disabilities**. Proceedings of the Eleventh International Conference on Technology and Persons with Disabilities (CSUN '96). California, E.E.U.U. Disponible en:
<http://www.cs.bc.edu/~eagleeye/papers/paper1/paper1.html>
2. Glenstrup, Arne J. y Theo Engell-Nielsen. **Eye controlled media: Present and future state**. Tesis Psicología de la Información, DIKU, Universidad de Copenhagen , 1995. Disponible en:
<http://www.diku.dk/~panic/eyegaze/article.html>
3. Majaranta, Päivi y Kari-Jouko Räihä. **Twenty years of eye typing: Systems and Design issues**. Proceedings of ETRA 2002, Eye Tracking Research and Applications Symposium 2002, 25-27 March 2002, Luisiana, E.E.U.U. 15-22. 2002 Disponible en
<http://www.cs.uta.fi/~curly/publications/ETRA2002-Majaranta-Raiha.pdf>
4. MATLAB 7.0. Help
5. Milekic, Slavko. **The more you look the more you get: Intention-based interface using gaze-tracking**. The University of the Arts, E.E.U.U. Disponible en:
<http://www.archimuse.com/mw2003/papers/milekic/milekic.html>
6. West, P. **A Roadmap for Building a Machine Vision System**. Automated Vision Systems.
<http://www.imagenation.com/pdf/roadmap.pdf>
7. West, P. **Should a Frame Grabber Be Part of Your Picture? Five Questions to Ask About Your Vision Application**. CyberOptics Semiconductor. 2004
http://www.machinevisiononline.org/public/articles/CyberOptics_Should_Grabber_Picture.pdf
8. **Choosing to Use a Custom Lens**. Melles Griot
<http://www.mellesgriot.com/pdf/pg6.pdf>

9. **Creating a Custom Solution.** Melles Griot.
<http://www.mellesgriot.com/pdf/pg7.pdf>
10. **Choosing a Frame Grabber for Imaging or Machine Vision.** Data Translation Inc.
http://www.machinevisiononline.org/public/articles/Choosing_Imaging_Boards.pdf
11. **Detailed Medical and Technical Information. Eyegaze Communications System**
<http://www.eyegaze.com/2Products/Disability/Medicaldata.html>
12. **Edge detection** National Instruments
<http://zone.ni.com/devzone/conceptd.nsf/webmain>
13. **Eye typing**
<http://www.cs.uta.fi/hci/gaze/eyetyping.php>
14. **Estadística descriptiva**
<http://ftp.medprev.uma.es/libro/node12.htm>
http://en.wikipedia.org/wiki/Descriptive_statistics
15. **Image Filtering Overview.** National Instruments
<http://zone.ni.com/devzone/conceptd.nsf/webmain/E69D110920B99E838625685E0081B881>
16. **Image Analysis and processing.** National Instruments
<http://zone.ni.com/devzone/conceptd.nsf/webmain>
17. **Machine Vision Lens Fundamentals.** Melles Griot.
<http://www.mellesgriot.com/pdf/pg11-19.pdf>
18. **Machine Vision Lighting Fundamentals.** Melles Griot.
<http://www.mellesgriot.com/pdf/pg20-24.pdf>
19. **Machine Vision Overview.** Melles Griot
<http://www.mellesgriot.com/pdf/pg4-5.pdf>
20. **Mathworks**
www.mathworks.com
21. **Mediana**
<http://es.wikipedia.org/wiki/Mediana>

22. **Mounting and Enclosures Fundamentals.** Melles Griot.
<http://www.mellesgriot.com/pdf/pg30.pdf>
23. **Pattern matching strategies.** National Instruments
<http://zone.ni.com/devzone/conceptd.nsf/webmain>
24. **Region of interest (ROI) manipulations.** National Instruments
<http://zone.ni.com/devzone/conceptd.nsf/webmain>
25. **Video Camera for Machine Vision.** Melles Griot.
<http://www.mellesgriot.com/pdf/pg25-29.pdf>
26. **Lighting presentation**
http://www.illuminationtech.com/seminars/photon_basics.pdf#search='machine%20visiontutorial'

APÉNDICE

CÓDIGO UTILIZADO

FUNCIONES COMUNES A AMBAS VERSIONES

% FUNCIÓN DE CORRELACIÓN MODIFICADO

```
function r = corr2mod(varargin)
```

```
A=varargin{1};
```

```
B=varargin{2};
```

```
a=double(A);
```

```
b=double(B);
```

```
a = a - mean2(a);
```

```
b = b - mean2(b);
```

```
r = sum(sum(a.*b))/sqrt(sum(sum(a.*a))*sum(sum(b.*b)));
```

%FUNCIÓN DE COMPARACIÓN

```
function [indice, coefi]= Comparacion2mod(ImagRef, TD)
```

```
    for i=1:length(ImagRef)
```

```
        rescorr(i)=corr2mod(ImagRef{i},TD);
```

```
    end
```

```
    coefi=max(rescorr); %mayor coeficiente de correlación
```

```
    indice= find(rescorr==max(rescorr));%encuentra tecla con máxima correlación
```

%FUNCIÓN DE FINALIZACIÓN

```
function AICerrar
```

```
    %profile viewer
```

```
    delete(gcf)
```

```
    clear all
```

```
    close all
```

```
end
```

```
%FUNCIÓN DE LETRAS ALEATORIAS PARA VERSION DE PRUEBA:  
RANDLETRAS
```

```
function [palabra, letras]=RandLetras  
alfabeto=char(65:90);  
i=randperm(25);  
letras=alfabeto(i(1:8));  
j=randperm(8);  
palabra=letras(j(1:5));  
letras=[letras '-'];
```

VERSION A

```
function varargout = Teclado(varargin)  
% TECLADO Application M-file for Teclado.fig
```

```
%+++++  
%          INICIALIZACIÓN  
%  
%DESPLIEGUE Y POSICIONAMIENTO DEL TECLADO VIRTUAL
```

```
if nargin == 0
```

```
    fig = openfig(mfilename, 'reuse');  
    set(0, 'Units', 'Pixels');  
    pantalla=get(0, 'ScreenSize');  
    pos=[5, 30, 5/6*pantalla(3)-10, pantalla(4)-50];  
    set(fig,'Position', pos);  
    set(fig,'CloseRequestFcn', 'AlCerrar');  
    handles = guihandles(fig);  
    guidata(fig, handles);  
    if nargout > 0  
        varargout{1} = fig;  
    end
```

```
elseif ischar(varargin{1})
```

```
    try  
        if (nargout)  
            [varargout{1:nargout}] = feval(varargin{:});  
        else  
            feval(varargin{:});  
        end  
    catch  
        disp(lasterr);  
    end
```

```

end

%DECLARACIÓN DE VARIABLES

global letra TD cam Tecla cadena y y2 mensaje letras palabra
global C coefi mensaje2 contaMat Cf t;

cadena="";
mensaje=handles.texto;
letra = "";
mensaje2=handles.TextoIni;
pos2=[pos(1)+pos(3), pos(2), pantalla(3)-pos(3)-pos(1), pos(4)-18];
contaMat=1;
Cf=0;
t=0;
%variables de la figura de cada tecla
Tecla(1)=handles.Tecla1;
Tecla(2)=handles.Tecla2;
Tecla(3)=handles.Tecla3;
Tecla(4)=handles.Tecla4;
Tecla(5)=handles.Tecla5;
Tecla(6)=handles.Tecla6;
Tecla(7)=handles.Tecla7;
Tecla(8)=handles.Tecla8;
Tecla(9)=handles.Tecla9;
Tecla(10)=handles.Tecla9;
y=wavread('blip.wav'); y2=wavread('chimes.wav');
cuando=datestr(clock, 'ddmmmTHHMM');
[palabra, letras]=RandLetras;

%CREACIÓN Y CONTROL DEL OBJETO DE ENTRADA DE LA CÁMARA
cam=videoinput('winvideo');
preview(cam);

handles=guihandles(fig);

%+++++
%          CALIBRACIÓN
ImagRef=MatrizFotos;

%%+++++
%%          PROGRAMA PRINCIPAL
tic
restaurarteclado(0)

```

```

guidata(fig, handles);
C = 1; %bandera para salir del programa

while (C ~= 21)

    pause(1)
    %ADQUISICIÓN
    TD=getsnapshot(cam);%adquisición de nueva imagen
    %PROCESAMIENTO
    TD=procesamiento(TD);
    %COMPARACIÓN
    [t_indice, coefi]= Comparacion2mod(ImagRef, TD);
    %-----

    Cf(contaMat)=coefi;
    contaMat=contaMat+1;

    %despliegue de imagen de referencia y la nueva imagen
    figure(1), set(gcf,'Position', pos2, 'Color', 'b', 'Name',...
    'Comparación de imagenes', 'Toolbar', 'none');
    subplot(2,1,1), imshow(TD, [-256 256])
    title('Imagen Nueva')
    xlabel(['coeficiente: ', num2str(coefi)], 'Color', [1,0.4,0.6])
    set(gca,'XTickLabel', {}, 'YTickLabel', {});
    text(100,100, ['Tecla: ', num2str(t_indice)], 'Color', 'y', 'FontSize', 14);
    text(100,140, ['Nivel: ', num2str(C)], 'Color', 'y', 'FontSize', 14);
    subplot(2,1,2), imshow(ImagRef{t_indice}, [-256 256])
    title('Imagen de Referencia'), axis off

    %ACCIONES
    handles=guihandles(fig);
    TecladoFcn(t_indice);
    %se guardan algunas variables de interes
    mediaC=mean(Cf); medianaC=median(Cf); stdC=std(Cf);
    mediaT=mean(t); medianaT=median(t); stdT=std(t);
    Info=struct('Version', {'A1'}, 'Fecha', {cuando}, 'Tiempos', {t}, ...
    'MediaT', {mediaT}, 'MedianaT', {medianaT}, 'StdT', {stdT}, 'Coeficientes',...
    {Cf}, 'MediaC', {mediaC}, 'MedianaC', {medianaC}, 'StdC', {stdC});
    save(cuando,'Info')
    guidata(fig, handles);
    %-----

end

```

```

%FINALIZACIÓN
closepreview(cam)
delete(cam)
delete(handles.figure1)
close all
clear

%+++++
%DECLARACIÓN DE FUNCIONES

%+++captura fotos patrones y forma una matriz de fotos+++
function FotosRef=MatrizFotos
global Tecla cam mensaje2
    pause(60);
    FotosRef=cell(1,10);
    FotoMala=getsnapshot(cam);
    set(mensaje2, 'String', 'PORFAVOR OBSERVE LA TECLA ROJA');
    for i=1:9
        set(Tecla(i), 'BackgroundColor', 'r');
        pause(6);
        FotosRef{i}=getsnapshot(cam);
        FotosRef{i}=procesamiento(FotosRef{i});
        set(Tecla(i), 'BackgroundColor', [0.251,0.502,0.502]);
    end
    set(mensaje2, 'String', 'Cierre los ojos por 10 segundos');
    pause(6);
    FotosRef{10}=getsnapshot(cam);
    FotosRef{10}=procesamiento(FotosRef{10});

end

%+++se encarga de manejar la figura del teclado y escribir a archivo
%+++las letras tecleadas de forma virtual
function TecladoFcn(t_ind)
global Tecla palabra letras letra cadena y2 mensaje C cam t
    if t_ind ~= 10
        restaurarteclado(t_ind)
        set(Tecla(t_ind),'BackgroundColor','r')
        letra=letras(t_ind);
    else
        erase=strcmp(letra, '-');
        if erase==1
            cadena=cadena(1: length(cadena)-1);
            set(mensaje, 'String', cadena);
        end
    end
end

```

```

        wavplay(y)
    else
        cadena=[cadena, letra];
        bien=strcmp(cadena, palabra);
        set(mensaje, 'String', cadena);
        if bien==1
            wavplay(y2)
            [palabra, letras]=RandLetras;
            cadena="";
            t(C)=toc;
            tic
            C=C+1;
            restaurarteclado(0);
        else
            wavplay(y)
        end
    end
    restaurarteclado(0);
end
end

% restaura el teclado a sus letras y colores originales
function restaurarteclado(t_indice)
    global Tecla letras mensaje2 palabra
    for i=1:9
        if i~=t_indice
            set(Tecla(i),'String', letras(i));
        end
        set(Tecla(i),'BackgroundColor',[0.251,0.502,0.502])
    end
    set(mensaje2, 'String', ['Escriba: ' palabra])
end

end

```

VERSIÓN B

%INICIALIZACIÓN

%DESPLIEGUE Y POSICIONAMIENTO DEL TECLADO VIRTUAL
if nargin == 0

```

    fig = openfig(mfilename, 'reuse');
    set(0, 'Units', 'Pixels');
    pantalla=get(0, 'ScreenSize');
    pos=[5, 30, 5/6*pantalla(3)-10, pantalla(4)-50];
    set(fig,'Position', pos);
    set(fig,'CloseRequestFcn', 'AlCerrar');
    handles = guihandles(fig);
    guidata(fig, handles);
    if nargout > 0
        varargout{1} = fig;
    end

elseif ischar(varargin{1})
    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:});
        else
            feval(varargin{:});
        end
    catch
        disp(lasterr);
    end

end

%DECLARACIÓN DE VARIABLES

global letra TD cam Tecla cadena y y2 mensaje letras palabra
global C coefi mensaje2 contaMat Cf t pos2;

cadena="";
mensaje=handles.texto;
letra = "";
mensaje2=handles.Textolni;
pos2=[pos(1)+pos(3), pos(2), pantalla(3)-pos(3)-pos(1), pos(4)-18];
contaMat=1;
Cf=0;
t=0;
%variables de la figura de cada tecla
Tecla(1)=handles.Tecla1;
Tecla(2)=handles.Tecla2;
Tecla(3)=handles.Tecla3;
Tecla(4)=handles.Tecla4;

```

```

Tecla(5)=handles.Tecla5;
Tecla(6)=handles.Tecla6;
Tecla(7)=handles.Tecla7;
Tecla(8)=handles.Tecla8;
Tecla(9)=handles.Tecla9;
Tecla(10)=handles.Tecla9;
y=wavread('blip.wav'); y2=wavread('chimes.wav');
cuando=datestr(clock, 'ddmmmTHHMM');
[palabra, letras]=RandLetras;

%CREACIÓN Y CONTROL DEL OBJETO DE ENTRADA DE LA CÁMARA
cam=videoinput('winvideo',1,'RGB24_640x480');
fuente=getselectedsource(cam);
%brillo elevado, imagen obtenida en escala de grises
set(fuente, 'brightness', 10000, 'PanMode', 'Manual', 'ZoomMode',...
    'Manual', 'TiltMode', 'Manual')
preview(cam);

handles=guihandles(fig);
%+++++
%
%          CALIBRACIÓN
[Iris, IrisIni]=MatrizFotos;

%+++++
%
%          PROGRAMA PRINCIPAL
tic
profile on
restaurarteclado(0)
guidata(fig, handles);
C = 1; %bandera para salir del programa

while (C ~= 21)

    pause(1)
    %ADQUISICIÓN
    TD=getsnapshot(cam);%adquisición de nueva imagen
    %PROCESAMIENTO
    TD=Procesamiento2(TD); %mejoramiento de nueva imagen
    [corrM, IrisN]=patternMatchingNoFigMod(TD, IrisIni);%reconocimiento de
patrones
    %COMPARACIÓN
    [t_indice, coefi]= Comparacion2mod(Iris, IrisN);
    %-----

```

```

Cf(contaMat)=coefi;
contaMat=contaMat+1;

%despliegue de imagen de referencia y la nueva imagen
figure(1), set(gcf,'Position', pos2, 'Color', 'b', 'Name',...
'Comparación de imagenes', 'Toolbar', 'none');
subplot(2,1,1), imshow(IrisN)
title('Nueva Imagen')
set(gca,'XTickLabel',{}, 'YTickLabel', {});
text(10,10, ['Tecla: ', num2str(t_indice)], 'Color', 'y', 'FontSize', 14);
text(10,30, ['Nivel: ', num2str(C)], 'Color', 'y', 'FontSize', 14);
subplot(2,1,2), imshow(Iris{t_indice})
title('Imagen de Referencia')
xlabel(['coeficiente: ', num2str(coefi)], 'Color', [1,0.4,0.6])
axis off

%ACCIONES
handles=guihandles(fig);
TecladoFcn(t_indice);
%se guardan algunas variables de interes
mediaC=mean(Cf); medianaC=median(Cf); stdC=std(Cf);
mediaT=mean(t); medianaT=median(t); stdT=std(t);
Info=struct('Version', {'B1'}, 'Fecha', {cuando}, 'Tiempos', {t}, ...
'MediaT', {mediaT}, 'MedianaT', {medianaT}, 'StdT', {stdT}, 'Coeficientes',...
{Cf}, 'MediaC', {mediaC}, 'MedianaC', {medianaC}, 'StdC', {stdC});
save(cuando,'Info')
guidata(fig, handles);
%-----

end

%FINALIZACIÓN
closepreview(cam)
delete(cam)
delete(handles.figure1)
close all
clear

%+++++
%DECLARACIÓN DE FUNCIONES

%+++captura fotos patrones y forma una matriz de fotos+++
function [Iris, IrisIni]=MatrizFotos
global Tecla cam mensaje2 pos2

```

```

pause(30);
FotosRef=cell(1,10);
fotos=cell(1,10);
Iris=cell(1,10);
FotoMala=getsnapshot(cam);

foto=getsnapshot(cam);
[IrisIni, rect]=imcrop(foto);
rect2=[rect(1)-rect(3), rect(2)-rect(4)/2, 3*rect(3), 2*rect(4)];
set(cam, 'ROIPosition', rect2);
IrisIni=Procesamiento2(IrisIni);
pause(2)
set(mensaje2, 'String', 'PORFAVOR OBSERVE LA TECLA ROJA');
for i=1:9
    set(Tecla(i), 'BackgroundColor', 'r');
    pause(4);
    FotosRef{i} = getsnapshot(cam);
    FotosRef{i} = Procesamiento2(FotosRef{i});
    [corrM, Iris{i}]=patternMatchingNoFigMod(FotosRef{i}, IrisIni);
    set(Tecla(i), 'BackgroundColor', [0.251,0.502,0.502]);

    figure(1), set(gcf,'Position', pos2, 'Color', 'b', 'Name',...
    'Imagen de Referencia', 'Toolbar', 'none');
    subplot(2,1,1), imshow(Iris{i})
    title(['Imagen de Referencia: ', num2str(i)])
    set(gca,'XTickLabel', {}, 'YTickLabel', {});
    subplot(2,1,2), imshow(IrisIni)
    title('Iris Inicial')
    axis off;

end
set(mensaje2, 'String', 'Cierre los ojos por 10 segundos');
pause(4);
FotosRef{10} = getsnapshot(cam);
FotosRef{10} = Procesamiento2(FotosRef{10});
[corrM, Iris{10}]=patternMatchingNoFigMod(FotosRef{10}, IrisIni);

end

%+++se encarga de manejar la figura del teclado y escribir a archivo
%+++las letras tecleadas de forma virtual
function TecladoFcn(t_ind)
global Tecla palabra letras letra cadena y y2 mensaje C cam t
if t_ind ~= 10

```

```

    restaurarteclado(t_ind)
    set(Tecla(t_ind),'BackgroundColor','r')
    letra=letras(t_ind);
else
    erase=strcmp(letra, '-');
    if erase==1
        cadena=cadena(1: length(cadena)-1);
        set(mensaje, 'String', cadena);
        wavplay(y)
    else
        cadena=[cadena, letra];
        bien=strcmp(cadena, palabra);
        set(mensaje, 'String', cadena);
        if bien==1
            wavplay(y2)
            [palabra, letras]=RandLetras;
            cadena="";
            t(C)=toc;
            tic
            C=C+1;
            restaurarteclado(0);
        else
            wavplay(y)
        end
    end
    restaurarteclado(0);
end
end
end

% restaura el teclado a sus letras y colores originales
function restaurarteclado(t_indice)
    global Tecla letras mensaje2 palabra
    for i=1:9
        if i~=t_indice
            set(Tecla(i),'String', letras(i));
        end
        set(Tecla(i),'BackgroundColor',[0.251,0.502,0.502])
    end
    set(mensaje2, 'String', ['Escriba: ' palabra])
end
end
end

```

%FUNCIÓN DE LA FASE DE MEJORAMIENTO

```
function I=Procesamiento2(ojo)
ojo=rgb2gray(ojo);
%filtro que promedia
h=fspecial('average');
I=imfilter(ojo, h);
%Se complementa la imagen (fondo negro e iris blanco)
%para poder usar más facilmente |imfill|
I=imcomplement(I);
I=imclearborder(I);
%Se rellenan los hoyos
I1=imfill(I, 'holes');
se=strel('disk', 3);
I=imclose(I1, se);
I=imadjust(I);
```

%FUNCIÓN DE LA FASE DE RECONOCIMIENTO DE PATRONES

```
function [corrMax, Iris, x1, y1]=patternMatchingNoFigMod(Imagen, Ref)

[m, n]=size(Imagen)
[m1, n1]=size(Ref)
corrMax=0;
for x=1:m-m1
    for y=1:n-n1
        imagenDeComp=Imagen(x:x+m1-1, y:y+n1-1);
        corrAct=corr2mod(Ref, imagenDeComp);
        if corrAct>corrMax
            corrMax=corrAct;
            Iris=imagenDeComp;
            x1=x; y1=y;
        end
    end
end
end
```

FUNCIÓN PARA ENCONTRAR EL TAMAÑO PROMEDIO DE UNA PALABRA EN UN TEXTO GRANDE: LETRASPALABRAPROM

```
function PProm=letrasPalabraProm
fid = fopen('alicia.txt', 'r');
texto=fread(fid);
E=length(find(texto==' '));
PProm=round((length(texto)-E)/(E+1));
```