



Vol 15, N° 3
Julio - Septiembre 2015

ISSN: 1317-2255
Deposito Legal: pp 20002FA828
Dep. legal ppi 201502ZU4642

Multiciencias

Multiciencias

R M Cs



Universidad del Zulia
Revista Arbitrada Multidisciplinaria



LUZ Punto Fijo

Núcleo LUZ-Punto Fijo
Programa de Investigación y Posgrado
Falcón-Venezuela

Multiciencias / Revista Arbitrada Multidisciplinaria del Núcleo LUZ-Punto Fijo

MULTICIENCIAS, Vol.15, N° 3, 2015 (328 - 338)

ISSN: 1317-2255 / Deposito Legal: pp 20002FA828 / Dep. legal ppi 201502ZU4642

Sistema de movimiento del cursor del computador para personas con discapacidad utilizando técnicas de visión artificial

Iñaki Aguirre-Gil y Jesús Gaviria

*Departamento de Control y Automatización Escuela de Ingeniería de Sistemas. Facultad de Ingeniería
Universidad de Los Andes, Mérida, estado Mérida. Venezuela*

iaguirre@ula.ve; gsusgp@hotmail.com

Resumen

En este trabajo se presentan dos aplicaciones basadas en técnicas de visión artificial con diferentes utilidades. La primera permite a personas con discapacidad que interactúen con el computador de manera sencilla, sustituyendo el movimiento del ratón por los movimientos controlados de abrir y cerrar el ojo. La segunda aplicación es diseñada con el objetivo de contribuir a la reducción de accidentes de carro mediante la construcción de un sistema que emita una señal de alarma cuando detecte que el conductor de un automóvil cierra el ojo por un periodo superior a un segundo. Ambas aplicaciones fueron modeladas mediante redes de Petri e implementadas con éxito en un computador personal.

Palabras clave: visión artificial; red petri; discapacidad; sistema de movimiento.

Computer cursor movement system for people with disabilities using vision techniques.

Abstract

In this paper we present two applications with different utilities based on computer vision techniques. The first application can help people with disabilities to interact with a computer in a simple substituting the computer mouse with controlled movements by opening and closing the eye. The second application is designed with the aim of contributing the reduction of car accidents by implementing an alarm signal that alerts the car driver when it closes the eye for a period longer than one second. Both applications were modeled using Petri nets and successfully implemented on a personal computer.

Keywords: artificial vision; petri nets; disability; movement system.

1. Introducción

Mejorar la calidad de vida de personas con discapacidad es un área de gran sensibilidad y responsabilidad ciudadana. La creación de una herramienta sencilla que resuelve el problema de mover el cursor del computador a través del movimiento ocular usando técnicas de visión artificial fue propuesto y desarrollado por Justo, (2009). El sistema resultó con un gran éxito para personas con limitaciones de movilidad y que pudiesen coordinar el movimiento ocular, sin embargo los usuarios al largo tiempo de uso, manifestaban cansancio mental y físico. En el presente artículo se muestran las mejoras del trabajo previo y se integra con el trabajo de Aguirre y Amaro (2012), el cual consiste en mover el cursor del computador presionando una tecla para controlar los movimientos del ratón del computador. El nuevo sistema pretende simplificar la solución del sistema de Justo, utilizando un solo grado de libertad para el movimiento completo del cursor, a través de parpadeo voluntario, posteriormente se integra a la herramienta de escritorio desarrollada por Aguirre y Amaro (2012) como otra opción para mover el cursor (izquierda, derecha, arriba, abajo, clic y doble clic).

La segunda aplicación presentada en el artículo consiste en utilizar las bondades del primer sistema para crear una herramienta que verifique cuando un conductor vehicular cierre los ojos por un periodo de tiempo superior a un segundo y el sistema emita una señal de alarma para despertar al conductor. Esta herramienta sería de gran utilidad para evitar una gran cantidad de accidentes vehiculares.

Según Aguirre (2011), las Redes de Petri constituyen una de las herramientas matemáticas más sencillas e

intuitivas para modelar un sistema de eventos discretos (SED); gracias a su capacidad de representación gráfica, permiten visualizar y establecer a través de grafos las relaciones existentes entre los diferentes elementos que integran un sistema. La implementación de ambos sistemas se realiza en un computador personal con una cámara web de bajo costo.

2. Antecedentes

Existen trabajos previos realizados en la Universidad de Los Andes en Venezuela, diseñados para ayudar a personas con discapacidad, entre los cuales se encuentran: Interfaces amigables para personas con discapacidad y poca movilidad de manera que puedan interactuar con el computador, Justo (2009) y Aguirre y Amaro (2012), Aguirre y otros (2011). En este trabajo se hacen modificaciones en la implementación de los trabajos citados para lograr compatibilidad entre sus aplicaciones.

En el trabajo de Justo (2009) se realiza el movimiento del cursor a partir del movimiento del globo ocular, mediante un periférico alternativo que en este caso es una cámara web modificada que enfoca al ojo. La aplicación en cuestión permite interactuar de manera directa con el computador teniendo los cuatro movimientos (arriba, abajo, derecha, izquierda) mediante el correspondiente movimiento del ojo, y el clic y doble clic al mantener el ojo cerrado un tiempo determinado. El sistema al ser utilizado por más de una hora hace que el usuario presente cansancio y es por ello que se busca una nueva opción que logre atenuar esa situación. El trabajo desarrollado e implementado

con éxito resultó un punto de partida para el posterior desarrollo de aplicaciones más sencillas de utilizar por los usuarios finales.

Una segunda aplicación desarrollada por Aguirre y Amaro (2012) permite hacer los movimientos de cursor del computador, clic y doble clic, con un solo grado de libertad, pero en este caso, presionando la tecla *Alt* del teclado en el momento que una imagen es mostrada al usuario con la dirección destino, pudiéndose asignar una segunda tecla en su configuración como es el botón central del mouse. Esta aplicación resultó exitosa en su funcionamiento pero no abarca a las personas que poseen una discapacidad motora importante.

Uniendo las características de ambos trabajos, se desea desarrollar una herramienta versátil y sencilla que permita mover el cursor del computador a partir de técnicas de visión artificial y utilizando un solo grado de libertad, en este caso el abrir y cerrar el ojo.

3. Descripción e implementación del Sistema Propuesto

3.1 El periférico alternativo

Uno de los componentes vitales del sistema es la cámara *web* de baja resolución, económica y cuyo *hardware* es modificado de la siguiente manera: luego de abrir la cámara *web*, se localiza y se extrae el filtro de luz infrarroja, remplazándolo por un negativo fotográfico. En la Figura 1 se puede observar el proceso de apertura, ubicación del lente de la cámara, ubicación del sensor de luz y el lente que contiene el filtro, adicionalmente se muestra, un filtro de luz infrarroja, el cual es un trozo de cristal y se cambia por negativo de fotografía del mismo tamaño. (Justo y Aguirre, 2009).



Figura 1. Cámara modificada para operar de modo infrarrojo

Fuente: Justo y Aguirre, 2009

La iluminación se logra adjuntando un LED emisor de infrarrojos junto a la cámara, el cual tiene un interruptor y una resistencia de 68 ohmios ya que está conectada a la fuente de poder de 5V por un cable USB y el voltaje máximo permitido para el LED es de 2V. Luego se procede a ensamblar la cámara y se busca que quede posicionada de manera que enfoque el ojo del usuario y se mantenga fija así la persona mueva la cabeza. Se fija la cámara a una gorra haciendo uso de un tornillo y una tuerca para que quede enfocado en la zona de interés. En la Figura 2 se observa el prototipo físico final obtenido.



Figura 2. Prototipo final montado sobre una gorra.

Fuente: Justo y Aguirre, 2009

Luego de tener todo en su lugar se puede observar en la Figura 3 la diferencia entre las imágenes capturadas antes y después de la modificación de la cámara, resaltando la pupila del ojo como elemento a estudiar.

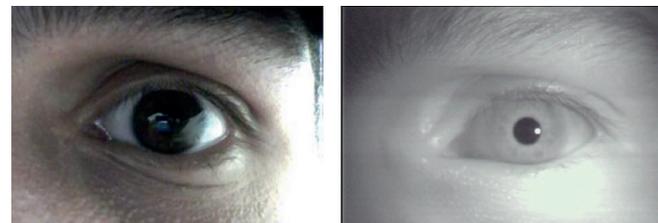


Figura 3. Imagen de la cámara antes y después de la modificación

3.2 Lógica de funcionamiento del Sistema Propuesto

El sistema está compuesto por dos aplicaciones, *MouseMover* desarrollado por Aguirre y Amaro (2012) y *VisionClic* desarrollado por Justo (2009). Al ejecutar el sistema Principal (*Mouse Mover*), se abre como una aplicación de alta prioridad en la esquina inferior derecha del monitor, cargando así todos sus

formularios y diferentes variables necesarias como son el refrescamiento de seis diferentes imágenes: flecha arriba, flecha abajo, flecha izquierda, flecha derecha, clic y doble clic. En paralelo se ejecuta otra aplicación que es la encargada de reconocer que se está presionando la tecla asignada e interactuando con la imagen mostrada.

De esta manera el cursor del ratón se moverá cuando se presione la tecla hacia la dirección en la que está la imagen en ese momento y de la misma manera, presionará clic y doble clic si se hace en el momento en que las imágenes respectivas son mostradas al usuario.

La aplicación principal cuenta con una barra de menú en la cual se encuentra la pestaña de ayuda, que abre una ventana de “Ayuda” y una de “Acerca de”. También tiene una pestaña de configuración que permite salir del sistema o abrir la ventana de configuración general, dentro de esta ventana se pueden editar algunas variables que cambiarán el comportamiento del sistema principal, cambiar la velocidad con que se muestran las imágenes en segundos, la rapidez con que se moverá el puntero cuando se presione la tecla asignada, cambiar la tecla por el botón central del Mouse, dejarla en su defecto que es la tecla *Alt* e invocar la segunda aplicación (*VisionClic*).

Si se solicita *VisionClic*, se muestra una ventana de bienvenida, luego se ejecuta un proceso de calibración del ojo abierto y del ojo cerrado, tal como se muestra en la Fig. 4.

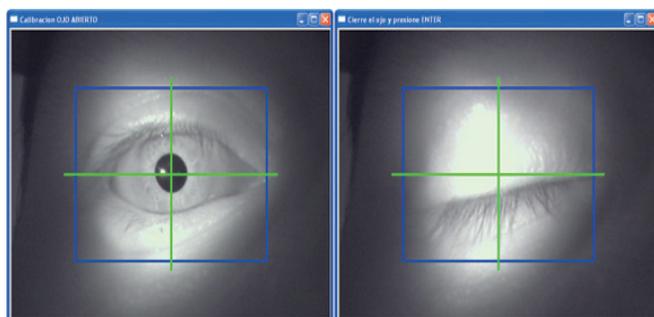


Figura 4. Calibración de imagen (a) Ojo abierto (b) Ojo cerrado.

Fuente: Justo y Aguirre, 2009

Al cerrar el ojo, *VisionClic* hace uso de un algoritmo de visión artificial para detectar esta acción y ejecutarla acción de presionar una tecla de manera de controlar el ratón con el parpadeo voluntario.

Una vez estudiada y analizada la lógica de funcionamiento del sistema propuesto se procede a realizar el modelo utilizando Redes de Petri.

3.3 Modelado utilizando Redes de Petri

Para Peterson (1981) las Redes de Petri son una herramienta que se utiliza para modelar y estudiar sistemas. La teoría de las Redes de Petri permite modelar un sistema y representarlo matemáticamente. A través de esta técnica es posible revelar información importante acerca del comportamiento dinámico del sistema, la cual puede ser utilizada para mejorar el sistema o para determinar fallas.

La representación gráfica se realiza a través de grafos, en los cuales, los estados son representados por nodos y las transiciones son representadas por arcos como lo sugieren Moody y Antsaklis (1998). Las funciones de entrada y salida determinan el destino y la procedencia de cada uno de los arcos como lo planteada García (2004). Por otro lado, al modelar un sistema complejo usando Redes de Petri, los estados y las transiciones son representados por etiquetas (letras y/o números), y luego el significado de cada estado y transición es detallado en tablas donde se asocia cada estado o transición con la etiqueta mostrada en el grafo. El modelado del sistema planteado en este artículo hace uso de Redes de Petri.

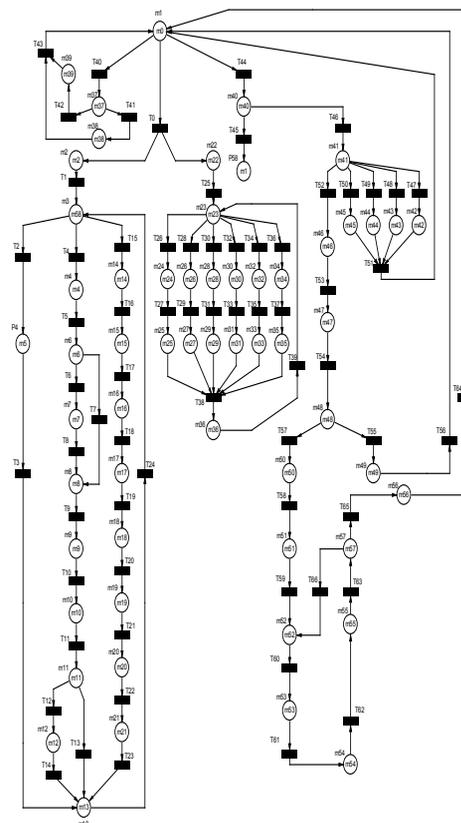


Figura 5. Modelado del sistema con Redes de Petri

Fuente: Justo y Aguirre, 2009

Tabla 1. Descripción de los lugares en la Red de Petri

Lugar	Descripción
m1	Inicializar Variables, Cargar Imágenes y Formularios de la Aplicación
m2	Iniciar Temporizador 1
m3	Leer Valores de tecla, conmutador, band e Iniciar Temporizador 2
m4	Detener Temporizador 2
m5	Hacer band=1
m6	Leer valor de band
m7	Esperar 1000ms / hacer band=0 / hacer tecla = no presionada / hacer conmutador=1
m8	Esperar 20ms
m9	Tomar Posición del Cursor
m10	Mover cursor un valor_pixel hacia la dirección D
m11	Leer Valores de tecla y conmutador
m12	Hacer conmutador=0
m13	Esperar 20ms (Temporizador 1)
m14	Detener Temporizador 2
m15	Esperar 1000ms
m16	Leer Posición del Cursor
m17	Hacer botón = presionado
m18	Esperar 10ms
m19	Hacer botón = no_presionado
m20	Esperar 300ms
m21	Estado que repite m17, m18, m19 y m20 (Solo se activa cuando Imagen D = 6)
m22	Iniciar Temporizador 2
m23	Leer Valor de imagen_variable
m24	Activar Imagen 1 y Desactivar las demás.
m25	Hacer imagen_variable=2
m26	Activar Imagen 2 y Desactivar las demás.
m27	Hacer imagen_variable=3
m28	Activar Imagen 3 y Desactivar las demás.
m29	Hacer imagen_variable=4
m30	Activar Imagen 4 y Desactivar las demás.
m31	Hacer imagen_variable=5
m32	Activar Imagen 5 y Desactivar las demás.
m33	Hacer imagen_variable=6
m34	Activar Imagen 6 y Desactivar las demás.
m35	Hacer imagen_variable=1
m36	Esperar tiempo ms (Temporizador 2)
m37	Desplegar Pestaña de Ayuda
m38	Abrir Formulario de Instrucciones
m39	Abrir Formulario de Acerca de
m40	Desplegar Pestaña de Opciones
m41	Abrir Formulario de Configuración
m42	Preparar nuevo valor_pixel
m43	Preparar nuevo Temporizador 2
m44	Preparar nuevo valor tecla
m45	Preparar nuevo valor tecla
m46	Preparar para ejecutar Programa
m47	Mostrar Ventana de Bienvenida
m48	Verificar la variable Capture, para validar captura desde la cámara web
m49	Mostrar Advertencia que indica que no tiene cámara conectada al computador
m50	Mostrar ventana de web cam en tiempo real para calibración de ojo abierto
m51	Mostrar ventana de web cam en tiempo real para calibración de ojo cerrado
m52	Restar Imagen capturada en calibración con imagen tomada en tiempo real
m53	Aplicar umbral establecido a la imagen resultante de la resta
m54	Calcular Histograma resultante
m55	Comparar histograma resultante con el histograma de una imagen totalmente negra
m56	Preparar Valor de tecla Nueva
m57	Si se superó el umbral; Si la bandera es 0, presionar la tecla asignada y cambiar la bandera a 1; si la bandera es 1, Soltar la tecla asignada y cambiar la bandera a 0

Tabla 2. Descripción de las transiciones en la Red de Petri

Transición	Descripción
T0	Variables Inicializadas, Imágenes y Formularios Cargados.
T1	Temporizador 1 Iniciado
T2	tecla = no presionada
T3	band = 1
T4	(Tecla= presionada & imagen(D) = activada) (conmutador = 1)
T5	Temporizador 2 detenido
T6	band=1
T7	band=0 & tecla = no presionada & (Se ha esperado 1ms)
T8	band=0
T9	Se ha esperado 20ms
T10	Posición del cursor tomada
T11	Posición del cursor escrita
T12	tecla= presionada & conmutador = 1
T13	tecla = no presionada
T14	conmutador = 0
T15	Tecla = presionada & Imagen(D) = activa
T16	Temporizador 2 detenido
T17	Se ha esperado 1000ms
T18	Posición del cursor tomada
T19	Tecla = presionada
T20	Se ha esperado 10ms
T21	Tecla = arriba
T22	Se ha esperado 300ms
T23	1
T24	Se ha esperado 20ms (Temporizador 1)
T25	Temporizador 2 Iniciado
T26	Imagen_variable = 1
T27	Imagen D = activa, Imágenes (Todas - D) = desactivadas
T28	Imagen_variable = 2
T29	Imagen D = activa, Imágenes (Todas - D) = desactivadas
T30	Imagen_variable = 3
T31	Imagen D = activa, Imágenes (Todas - D) = desactivadas
T32	Imagen_variable = 4
T33	Imagen D = activa, Imágenes (Todas - D) = desactivadas
T34	Imagen_variable = 5
T35	Imagen D = activa, Imágenes (Todas - D) = desactivadas
T36	Imagen_variable = 6
T37	Imagen D = activa, Imágenes (Todas - D) = desactivadas
T38	1
T39	Se ha esperado tiempo_1 (Temporizador 1)
T40	Clic sobre Ayuda
T41	Clic sobre Instrucciones
T42	Clic sobre Acerca de
T43	Formulario de Instrucciones o Ayuda Cerrado
T44	Clic sobre Opciones
T45	Clic sobre Salir
T46	Clic sobre Configuración
T47	Velocidad de Cursor Modificado
T48	Tiempo de Muestra de Imagen Modificado
T49	Clic Sobre Radio Button Tecla Alt
T50	Clic Sobre Radio Button Boton Central Mouse
T51	Clic sobre Aceptar/Cancelar
T52	Clic Sobre VisionClic
T53	Clic Sobre Aceptar
T54	Clic Sobre Iniciar
T55	Si Capture == 0
T56	Clic Aceptar
T57	Si Capture ≠ 0
T58	Presiona cualquier tecla
T59	Presiona cualquier tecla
T60	Imágenes restadas
T61	Umbral aplicado
T62	Histograma calculado
T63	Comparación hecha, valor retornado
T64	Tecla Enviada a Programa Principal
T65	Tecla Presionada
T66	Si se capturan imágenes similares a la almacenada en proceso de calibración de ojo cerrado por un tiempo aproximado de un segundo.

3.4 Implementación del Sistema

La implementación de la herramienta se realiza bajo el sistema operativo Windows 7 Home Premium, haciendo uso del programa Microsoft Visual Studio 2010 con los lenguajes C++ y Visual Basic. Adicionalmente se utiliza la librería OpenCV 2.4 que brinda una plataforma de funciones de visión artificial como lo describe Amaro (2010).

3.4.1 Descripción Visual Detallada de la Herramienta Propuesta en Ejecución

Se ejecuta el programa principal *MouseMover 2.0* y se comienzan a mostrar las imágenes que registrarán el comportamiento del cursor cuando el usuario cierre el ojo. La secuencia de las imágenes que se muestran a continuación (Figuras 6 a la 11) muestran los diferentes movimientos del cursor del computador.

Movimiento hacia abajo: al seleccionar esta opción al cerrar el ojo por un segundo, el cursor del computador se moverá hacia abajo mientras se tenga presionada esta opción o cuando se presione nuevamente la opción.



Figura 6. Indicador de movimiento hacia abajo

Movimiento hacia arriba: al seleccionar esta opción al cerrar el ojo por un segundo, el cursor del computador se moverá hacia arriba mientras se tenga presionada esta opción o cuando se presione nuevamente la opción.



Figura 7. Indicador de movimiento hacia arriba

Movimiento hacia la izquierda: al seleccionar esta opción al cerrar el ojo por un segundo, el cursor del computador se moverá a la izquierda mientras se tenga presionada esta opción o cuando se presione nuevamente la opción.



Figura 8. Indicador de movimiento hacia la izquierda

Movimiento hacia la derecha: al seleccionar esta opción al cerrar el ojo por un segundo, el cursor del computador se moverá a la derecha mientras se tenga

presionada esta opción o cuando se presione nuevamente la opción.



Figura 9. Indicador de movimiento hacia la derecha

Doble clic del ratón: al seleccionar esta opción al cerrar el ojo por un segundo, hace la operación equivalente a realizar un doble clic en el ratón.



Figura 10. Indicador de doble clic

Clic del ratón: al seleccionar esta opción al cerrar el ojo por un segundo, se hace la operación equivalente a realizar un clic en el ratón o presionar la tecla *Enter*.



Figura 11. Indicador de clic

Haciendo uso de estos seis movimientos prefijados por la aplicación, el usuario interactúa de una manera sencilla y eficaz con el computador. Ya sea para actividades de formación, de expresión o de distracción.

4. Aplicación: Sistema de Seguridad VisionWarning

Al ser el parpadeo voluntario la razón de control del movimiento del cursor del computador, se propone una segunda aplicación dirigida a los conductores de automóviles y que advierta al usuario cuando se esté quedando dormido, es decir, cuando el ojo se cierre por un aproximado de un segundo y medio, de manera de evitar un gran número de accidentes automovilísticos, Gaviria (2013).

4.1 Lógica de Funcionamiento

La ejecución de la aplicación abre una ventana de bienvenida que introduce un proceso de calibración de ojo abierto y luego de ojo cerrado.

El sistema entra en un ciclo al ejecutar el algoritmo de detección de ojo cerrado, si el usuario cierra el ojo por un segundo y medio, simulando que posiblemente se esté durmiendo, se dispara una advertencia de sonido y un aviso en la pantalla.

4.2 Modelo en Redes de Petri del Vision Warning

En este apartado se muestra el comportamiento del sistema *VisionWarning* modelado con sistemas de eventos discretos, particularmente utilizando Redes de Petri. Adicionalmente se muestran las tablas de estados y lugares necesarios para entender el modelo.

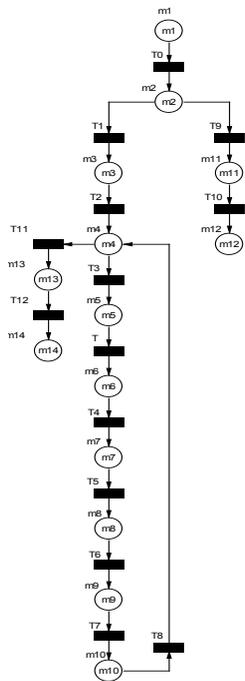


Figura 12. Modelado del Sistema VisionWarning con Redes de Petri

Fuente: Justo y Aguirre, 2009

Tabla 3. Descripción de los lugares en la Red de Petri

Lugar	Descripción
m1	Mostrar Ventana de Bienvenida
m2	Verificar la variable Capture, para validar captura desde la cámara web
m3	Mostrar ventana de web cam en tiempo real para calibración de ojo abierto
m4	Mostrar ventana de web cam en tiempo real para calibración de ojo cerrado
m5	Capturar Imagen
m6	Restar Imagen capturada en calibración con imagen tomada en tiempo real
m7	Aplicar umbral establecido a la imagen resultante de la resta
m8	Calcular Histograma resultante
m9	Comparar histograma resultante con el histograma de una imagen totalmente negra
m10	Disparar alarma de sonido y mostrar señal de advertencia
m11	Mostrar advertencia que indica que no tiene cámara conectada al computador
m12	Salir del Sistema
m13	Muestra menú que se despliega al hacer clic sobre el icono de la aplicación donde está la opción de cerrar la aplicación
m14	Salir del Sistema

Fuente: Justo y Aguirre, 2009

Tabla 4. Descripción de las transiciones en la Red de Petri

Transición	Descripción
T0	Clic Sobre Iniciar
T1	Si Capture \neq 0
T2	Presiona cualquier tecla
T3	Presiona cualquier tecla
T4	Imagen Capturada
T5	Imágenes restadas
T6	Umbral aplicado
T7	Histograma calculado
T8	Comparación hecha, valor retornado
T9	Si se capturan imágenes similares a la almacenada en proceso de calibración de ojo cerrado por un tiempo aproximado de un segundo y medio.
T10	Si Capture == 0
T11	Clic Aceptar
T12	Si se ejecuta clic derecho sobre el icono de la aplicación que se encuentra en la barra de Windows
T13	Si se presiona clic en cerrar aplicación

Fuente: Justo y Aguirre, 2009

4.3 Pruebas y Resultados del Sistema de Seguridad VisionWarning

Se realizaron varias pruebas con diferentes usuarios y se concluyó que el prototipo desarrollado es una herramienta fácil de usar, amigable y funciona de manera eficiente. Es totalmente factible la modificación de variables como el aviso visual y el auditivo, pudiéndose cambiar de ámbito como por ejemplo a un automóvil, logrando una gran extensibilidad del prototipo. La idea es contribuir con este prototipo a reducir el número de accidentes automovilísticos. En las Figuras 13 y 14 puede observarse la detección del ojo cerrado y la señal de aviso.

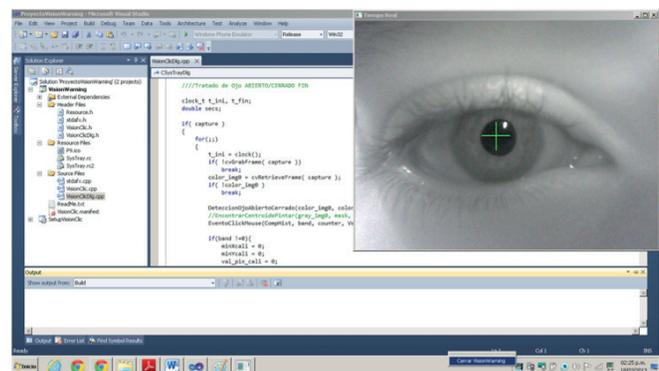


Figura 13. Sistema VisionWarning ojo abierto. (Gaviria, 2013)

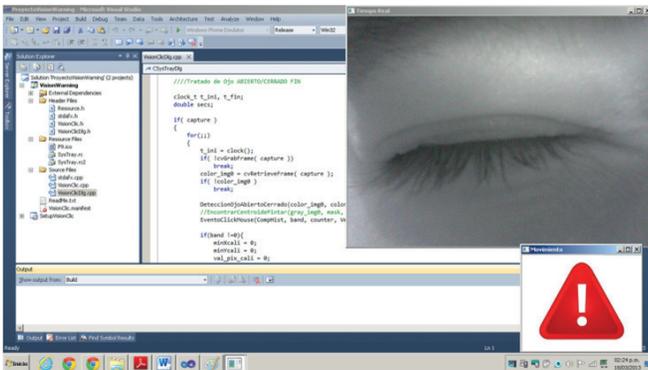


Figura 14. Sistema VisionWarning detectando el ojo cerrado y emitiendo la señal de alarma.

Fuente: Gaviria, 2013

5. Análisis de Resultados

El desarrollo y las pruebas realizadas de las aplicaciones descritas en el presente artículo tuvieron los resultados esperados como mejorar la calidad de vida de personas con discapacidad y reducir los accidentes automovilísticos. Se realizaron pruebas de funcionamiento de las dos aplicaciones en miras de obtener características específicas de comportamiento, obteniéndose aplicaciones estables, cómodas, con una interfaz de usuario amigable y que logra una operatividad sin complicaciones.

En las pruebas realizadas continuamente se observa que el ángulo y posición de colocación de la cámara respecto al ojo debe ser correcto, es decir, que el ojo sea enfocado de frente de tal manera que las pestañas no bloqueen el área donde se hace el tratado de la imagen.

De igual manera el proceso de calibración es vital al momento de obtener resultados por parte de la aplicación, por eso se debe hacer con mucha cautela y tener en cuenta que la cámara enfoca dependiendo de la luz que hay en el ambiente. Este proceso de calibración se debe hacer con calma tanto para el ojo abierto como para el ojo cerrado y enfocando bien el área del ojo que se va a tratar.

Como se trata de una resta de imágenes, se realizaron pruebas donde no necesariamente la cámara enfoca al ojo, sino una parte para que el usuario tuviese más rango de visión, así como pruebas donde no se enfoca al ojo en lo absoluto sino que en la imagen de ojo abierto se enfoca una superficie blanca y en la de ojo cerrado se coloca un dedo sobre la superficie.

Estas pruebas funcionan siempre y cuando el enfoque de la cámara sea el mismo para todo momento,

es decir, la cámara debe estar sobre la gorra en la misma posición y la persona no debe mover mucho la cabeza, y para el caso del dedo en la superficie, el dedo debería estar en el mismo lugar donde se calibró como ojo cerrado, de lo contrario no servirá, se recomienda siempre usarlo en la posición correcta, enfocando el ojo ya que provee un poco más de flexibilidad motora para el usuario que está usando la aplicación.

Respecto al programa desarrollado *MouseMover 2.0*, resultó ser completamente eficiente y altamente funcional. Se prueba que la configuración de tiempo de muestra de imagen se puede reducir o incrementar para *VisionClic* al igual que la velocidad del cursor.

Se probaron las funciones básicas como abrir una carpeta, abrir un archivo, navegar en un explorador de internet y lo más importante, se probó su funcionamiento sobre un teclado virtual que a su vez escribe sobre un programa que reproduce con voz natural lo anteriormente escrito. De esta manera se le da mayor funcionalidad y extensibilidad al programa y al usuario.

Respecto al uso de recursos, la aplicación principal ocupa entre 30MB y 35MB de la memoria RAM, por lo que lo hace un programa bastante liviano y fluido. Si se invoca a *VisionClic* se sumarán otros 30MB al uso de recursos, pero esto no influirá en el desempeño de la computadora personal.

En sentido general, el sistema de movimiento del cursor para personas con discapacidad diseñado e implementado en este proyecto tiene un perfecto desempeño, haciendo uso de un solo grado de libertad, de una forma eficiente, segura, usando pocos recursos y teniendo una interfaz agradable para el usuario.

Consideraciones finales

El periférico alternativo resultó ser eficiente, brindando al usuario con discapacidad la posibilidad de interactuar con el computador de una manera sencilla y dando una mejor calidad de vida.

Las aplicaciones desarrolladas poseen una gran rentabilidad, ya que son mucho menos costosas que productos similares existentes en el mercado, lo cual muestra un potencial para su producción y distribución.

Las pruebas realizadas muestran que las aplicaciones son totalmente eficaces, cumpliendo todos los objetivos de la investigación.

Referencias

- [1] AGUIRRE, Iñaki (2011). Análisis y Descripción de Técnicas de Automatización, Talleres Gráficos de la ULA, Mérida, Venezuela.
- [2] AGUIRRE, I; JUSTO, F; AMARO, M, RAMÍREZ, A (2011). “Descripción de Tres Aplicaciones de bajo costo para Ayuda a Personas con Discapacidad”, en: **Revista Ciencia e Ingeniería**, Vol. 32, No. I.
- [3] AGUIRRE, I; AMARO, M (2012). “Sistema de Movimiento del cursor del computador para Personas con Discapacidad”, en: **Revista Ciencia e Ingeniería**, Vol. 33, No.2, pp. 3-10.
- [4] AMARO, Manuel (2010). Diseño e implementación de un sistema que permita el control de la trayectoria de un vehículo, a través de un Sistema de Visión Artificial, Proyecto de Grado, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela.
- [5] JUSTO, F; AGUIRRE, I. (2009). Creación de una Herramienta que Permita Mover el Cursor de un Computador a Partir del Movimiento Ocular, Utilizando Técnicas de Visión Artificial, Seventh Latin American And Caribbean Conference for Engineering and Technology, LACCEI 2009, San Cristóbal, Venezuela.
- [6] JUSTO, Francisco (2009). Creación de una Herramienta que Permita Mover el Cursor de un Computador a Partir del Movimiento Ocular, Utilizando Técnicas de Visión Artificial, Proyecto de Grado, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela.
- [7] GAVIRIA, Jesús (2013). Sistema de Movimiento del Cursor del Computador para Personas con Discapacidad Utilizando Técnicas de Visión Artificial, Proyecto de Grado, Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela.
- [8] GARCÍA, E. (2004). **Automatización de Procesos Industriales**, Alfa Omega. España.
- [9] MOODY, J; ANTSAKLIS, P (1998). **Supervisory Control of Discrete Event System using Petri Nets**. Publicado por Kluwer Academic Publishers. Norwell, MA. Estados Unidos.
- [10] Microsoft Developer Network (2009). Librerías, definición de funciones y ejemplos [Documentos WWW], recuperado: <http://msdn.microsoft.com> en marzo 2009.
- [11] PETERSON, J. (1981). **Petri Net Theory and Modeling of Systems**. Prentice-Hall, Inc.



UNIVERSIDAD
DEL ZULIA

Multiciencias

Vol 15, N° 3

Edición por el Fondo Editorial Serbiluz.

Publicada en septiembre de 2015.

Universidad del Zulia. Maracaibo-Venezuela

www.luz.edu.ve

www.serbi.luz.edu.ve

produccioncientifica.luz.edu.ve