

ISSN 2070-836X

APORTE SANTIAGUINO

Revista de Investigación

Volumen 4 n.º 2, julio – diciembre 2011

*Ciencia,
cultura,
tecnología
e innovación*



Huaraz, Perú

latindex
Sistema Regional de Información en Línea
para Revistas Científicas de América
Latina, el Caribe, España y Portugal

ARTÍCULOS ORIGINALES

- Modelo probabilístico y regional de las descargas máximas instantáneas del río Santa en la región Ancash. [Probabilistic model and regional the instantaneous maximum discharge the river Santa in the Ancash region]..... 135
Abelardo Díaz Salas.
- Sistema de visión artificial para la detección de somnolencia de conductores, basado en el comportamiento ocular. [Artificial vision system for the detection of drivers' drowsiness, based on the ocular behavior]..... 145
Jesús E. Espinola Gonzales, Maximiliano E. Asís López, Vladimir G. Rodríguez Sabino.
- La efectividad del Control Interno en la Gestión de los Gobiernos Locales en el Callejón de Huaylas. Periodo 2006- 2007. [The effectiveness of Internal Control in the management of local government in the Callejón de Huaylas. Period 2006 - 2007]..... 152
Loel Salutor Bedón Pajuelo, Juan Alejandro Murga Ortiz, Manuel Morales Alberto.
- Constitucionalización de los Derechos de los Pueblos Indígenas en el Perú. [Constitutionalization of the Rights of the Indigenous Peoples in the Peru]..... 159
Luis Robles Trejo.
- Tutoría Virtual y Desempeño Académico en los Estudiantes de la Escuela Profesional de Educación de la FCSEC-UNASAM. [Virtual Tutoring and Academic Performance of the Students in the Educational Professional School at the FCSEC-UNASAM]..... 169
Rudecindo Penadillo L., Moisés Huerta R., Alberto Huamani G., Alfredo Zanabria P.
- Patologías Obstétricas durante el embarazo, parto y puerperio de las Adolescentes atendidas en el Hospital Víctor Ramos Guardia, Huaraz, 2009. [Obstetric Pathology during pregnancy, labor and postpartum in Adolescents treated in the Hospital Victor Ramos Guardia, Huaraz, 2009]. 177
Marcelo Arotoma O., Magna Guzmán A., Teresa Valencia V., Olga Cayra S.
- Evaluación del rendimiento y la acción conservante en carne de cerdo del aceite esencial de orégano (*Origanum vulgare L.*) cultivado en seis zonas altoandinas de Amazonas. [Performance Assessment and preservative action in pork and Oregano essential oil (*Origanum vulgare L.*) grown in six high Andes of Amazonas] 185
M. Ventura; C.E. Millones; E.A. Auquiñivin; E.R. Vásquez y A.R. Tafur.
- Caracterización de una colección de *Theobroma cacao L.* en Tingo María usando marcadores moleculares ISSR. [Characterization of a *Theobroma cacao L.* collection at Tingo Maria using ISSR molecular markers]. 195
Julio Chia W., Luis Garcia C., Mery Suni N. and Bertus Eskes.

Del aprendizaje colaborativo al aprendizaje por inmersión. [Of colaborative learning to immersive learning].....	203
<i>Kenneth Delgado S. G.</i>	
El cambio climático, la glaciología y los riesgos en la Cordillera Blanca Ancash – Perú. [Climate change, glaciology and risks in the cordillera Blanca Ancash - Perú].	208
<i>César A. Portocarrero R.</i>	

Sistema de visión artificial para la detección de somnolencia de conductores, basado en el comportamiento ocular

Artificial vision system for the detection of drivers' drowsiness, based on the ocular behavior

¹Jesús E. Espinola Gonzales.^a, ¹Maximiliano E. Asís López.^a, ¹Vladimir G. Rodríguez Sabino.^a

RESUMEN

El cansancio en los conductores de vehículos, que muchas veces se manifiesta en somnolencia, es la causa de muchos accidentes. En algunos países el índice de accidentes ocasionados por esta causa alcanza hasta un 35%. En el presente trabajo se presenta un sistema de visión artificial capaz de detectar el estado de somnolencia. El proceso consiste en la obtención de imágenes a través de una cámara de video; estas imágenes son procesadas mediante un algoritmo que se ofrece en este trabajo; como resultado se tiene una respuesta por parte del sistema que indica si el conductor tiene los ojos cerrados o abiertos. Para lograr los objetivos de este trabajo se ha empleado principalmente el Software Matlab, la herramienta de modelado UML y la metodología RUP.

Palabras clave: Visión Artificial, procesamiento de imágenes digitales, detección de cerrado y abierto .

ABSTRACT

The weariness in the drivers of vehicles, often is demonstrated in drowsiness, is the reason of many accidents. In some countries the index of accidents caused by this reason reaches up to 35 %. In this work a artificial vision system capable of detecting the condition of drowsiness is offered. The process consists of the obtaining image across a video camera, these images are processed by means of an algorithm offered in this work; as a result the system offers a response that indicates if the driver has the closed or opened eyes. To achieve the objectives of this work the Matlab software, UML modeling tool and the RUP methodology has been used.

Key words : Artificial Vision, digital image processing, detection of open and closed eyes.

¹ Facultad de Ciencias, Universidad Nacional Santiago Antúnez de Mayolo.

^a Lic. en Matemática.

INTRODUCCIÓN

El índice de accidentes, que tienen como causa la somnolencia de los conductores de vehículos, es bastante alto. En algunos países se ha llegado a reportar hasta un 35% de las causas. (Branzan A., et al., 2008)

La somnolencia produce la disminución de la capacidad de vigilia afectando negativamente a los sentidos, mermando la capacidad de percepción, reconocimiento y control del vehículo, y a la vez, incrementando la probabilidad de sufrir algún accidente o un sobresalto en la carretera. (Ji Q. and Yang X., 2002).

Una de las formas de manifestación de la somnolencia es a través del comportamiento ocular, el cual ha sido relacionado con el nivel de alerta de los individuos en conductores de vehículos, caracterizados y cuantificados con diferentes variables: frecuencia de parpadeos, duración del cierre de párpados y el porcentaje de ojos cerrados en el tiempo (Hamada T., et al., 2003).

Actualmente, los fabricantes de automóviles están incorporando elementos de alta tecnología en sus modelos con la finalidad de aumentar la seguridad. Sin embargo, esto incrementa su costo y resultan ser inaccesibles para la mayoría de la población, especialmente en nuestro país.

Por esta razón, existe la necesidad de implementar sistemas de visión artificial que incorporen técnicas apropiadas para detectar la somnolencia de conductores de vehículos, para de esta manera minimizar las probabilidades de que ocurra un accidente.

El objetivo del presente trabajo ha sido el estudio y desarrollo de una plataforma tecnológica basada en visión artificial que monitoree el comportamiento de los parpadeos de modo que pueda detectarse estados de baja alerta (somnolencia), en tiempo diferido y tiempo real.

Con la adecuada toma de imágenes de los ojos del conductor y la creación de algoritmos matemáticos, basados en el comportamiento ocular, así como la implementación del software, permitió la creación de un sistema de visión artificial que detecte el estado de somnolencia del conductor.

El proceso consiste en la obtención, mediante una cámara de video, de imágenes digitales de la zona de los ojos del conductor, lo cual es procesado por el sistema, usando el algoritmo desarrollado en esta investigación. Como respuesta, el sistema identifica los ojos cerrados y emite una alarma.

MATERIALES Y MÉTODOS

Selección del Hardware

Para este proyecto, por razones económicas, se eligió una Webcam. Las características más relevantes de esta cámara se enumeran a continuación:

Marca: Micronics ; Modelo: Voyager- MIC W313

CARACTERÍSTICAS:

Captura de Imagen y Video: Hasta 5 Mpixels 2560x2048(Interpolado)

Interface: USB 2

Formato de Video: 24 BIT RGB

Rango de Foco: 3 cm hasta el infinito. CMOS de alta resolución.

Plataforma: Notebooks o PC. Compatible con todos los Windows.

Selección del Software

MATLAB, Laboratorio de Matrices, es un lenguaje de alto nivel desarrollado por MathWorks cuyas funciones están optimizadas para el uso de matrices. Gracias a la diversidad de sus librerías aplicables a diversos campos del conocimiento, este programa es ampliamente utilizado en la educación y en la industria.

Su elemento básico de trabajo son las matrices. El programa permite realizar de un modo rápido la resolución numérica de problemas en un tiempo mucho menor que si se resolvieran con lenguajes de programación tradicionales como Fortran, Basic o C.

MATLAB permite la implementación de código de forma más sencilla y amigable que otros lenguajes de programación, gracias a un conjunto amplio de funciones especializadas denominadas Toolbox.

Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos los toolboxes de adquisición y procesamiento de imágenes y el Matlab Guide que nos permitirá realizar nuestro programa en ambiente visual.

El Toolbox de adquisición de imágenes es una colección de funciones que sirven, entre otras cosas, para adquirir imágenes de diversos dispositivos (desde frame grabbers profesionales hasta USB webcams), para visualizar videos en vivo, para la adquisición de imágenes mediante triggers, para importar datos hacia el entorno de trabajo de MATLAB, etc. En otras palabras, es la herramienta que nos permitirá realizar la aplicación en tiempo real.

El Toolbox de Procesamiento de Imágenes proporciona a MATLAB un conjunto de funciones que amplía las capacidades del producto para realizar desarrollo de aplicaciones y de nuevos algoritmos en el campo del proceso y análisis de imágenes.

El entorno matemático y de creación de MATLAB es ideal para el procesamiento de imágenes, ya que estas imágenes son, al fin y al cabo, matrices. Este toolbox incorpora funciones para:

- Diseño de filtros.
- Mejora y retocado de imágenes.
- Análisis y estadística de imágenes.
- Operaciones morfológicas, geométricas y de color.
- Transformaciones 2D, etc.

Modelado UML.

El paradigma orientado a objetos enfatiza la creación de clases que encapsulan tanto los datos, como los algoritmos que se utilizan para manejar los datos. Si se diseñan y desarrollan adecuadamente las clases orientadas a objetos, estas pueden reutilizarse en diferentes aplicaciones y arquitecturas de sistemas basados en computadoras.

A finales de la década de 1990, Grady Booch, James Rumbaugh e Ivar Jacobson colaboraron para combinar y recopilar las mejores características de sus métodos, adoptando los siguientes objetivos:

- Representar sistemas completos, en lugar de solamente la parte de software, usando conceptos orientados a objetos.
- Establecer una relación explícita entre conceptos y código ejecutable.
- Crear un lenguaje de modelado que pudiera ser usado por humanos y máquinas.

El resultado fue un método unificado, denominado UML (Unified Modeling Language, o Lenguaje de Modelado Unificado), que con bastante aceptación y éxito se ha empleado por número importante de desarrolladores de software para las labores de análisis y diseño de sistemas orientados a objetos.

UML presenta las ventajas de poder utilizarse como un lenguaje estándar, principalmente gráfico, para visualizar, especificar, construir y documentar los productos de un sistema de software. Se puede emplear con todos los procesos, a lo largo del ciclo de vida de desarrollo y a través de diferentes tecnologías de implementación.

Proceso de desarrollo

Aunque UML es bastante independiente del proceso de desarrollo que se siga, los mismos creadores de UML han propuesto su propia metodología de desarrollo, denominada el Proceso Unificado de Desarrollo.

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Además, el Proceso Unificado utiliza el UML para expresar gráficamente todos los esquemas de un sistema software. Pero, realmente, los aspectos que definen este Proceso Unificado son tres: es iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura:

- Dirigido por casos de uso: Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.
- Centrado en la arquitectura: En la arquitectura de la construcción, antes de construir un edificio este se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, etc. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.
- Iterativo e incremental: Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases.

Cada recorrido por las fases se denomina iteración en el proyecto en la que se realizan varios tipos de trabajo (denominados flujos). Además, cada iteración parte de la anterior incrementando o revisando la funcionalidad implementada. Se suele denominar proceso.

RESULTADOS

Se ha realizado pruebas tanto en tiempo real como en tiempo diferido en condiciones óptimas de luz, es decir con una buena iluminación; y una base de

datos de 30 fotos de individuos con ojos cerrados y 30 con ojos abiertos. Estos son los resultados:

	Error relativo	Error Porcentual
Ojos abiertos	0.10	10%
Ojos cerrados	0.067	6.7%

Algoritmo de Detección de Ojos abiertos o Cerrados

En la figura 1, se encuentra un diagrama de flujo básico de las funciones principales del sistema.

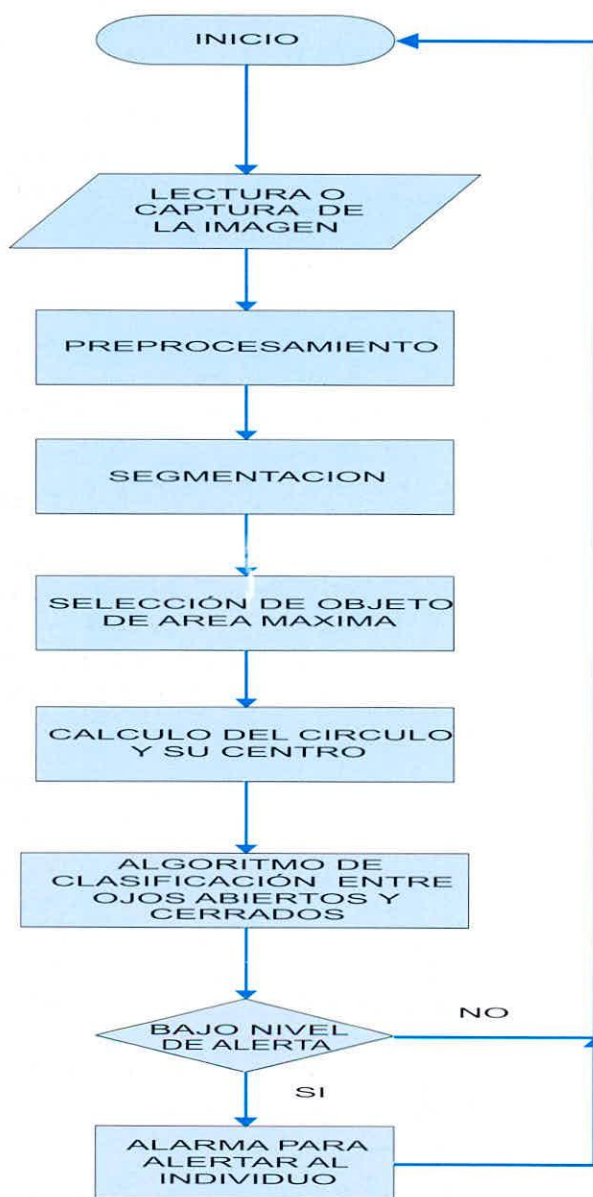


Figura 1. Organigrama Básico del Sistema

Lectura o captura de las imágenes

Para poder obtener la imagen capturada por la cámara es necesario contar con un programa (driver) que acceda los datos (imagen) desde la tarjeta de digitalización de imágenes. Sin embargo, el driver viene ya con la compra de la tarjeta y afortunadamente en Matlab ya existe un programa, el cual se enlaza al driver de la tarjeta de imagen que proporciona el proveedor de la tarjeta de imágenes, de tal manera que el objetivo inicial se centró en interpretar dicho programa y obtener la imagen como una matriz.

En caso de tiempo diferido se lee el archivo ya almacenadas las imágenes para poder trabajar con ellas. En caso de tiempo real se captura la imagen de una cámara webcam. Para realizar estos procesos debemos utilizar la clase Imagen y su método buscarImagen.

Pre-procesamiento

Inicialmente la imagen puede estar degradada para nuestros objetivos, debido a causas aleatorias en los procesos de captación, transmisión y digitalización de la imagen, así como de iluminación. Por ello, en esta fase se pretende eliminar estas distorsiones o mejorar las características de la imagen original para su posterior tratamiento.

Para la elaboración de este proyecto se analizó una serie de imágenes digitales, a las cuales debían aplicarse diversas técnicas de procesamiento de imagen y ver cuáles de ellas serían implementadas.

A continuación se muestran los pasos realizados en la etapa del preprocesamiento:

- Ecuilización del Histograma
- Incremento de contraste

Segmentación de Imagen

En este proceso se divide la imagen leída o capturada en objetos que será de nuestro interés; en este caso el objeto a identificar es el ojo, las curvaturas que presenta, para lo cual utilizaremos la clase Segmentación y el método de segmentación de detección de borde.

Determinación del objeto de área máxima

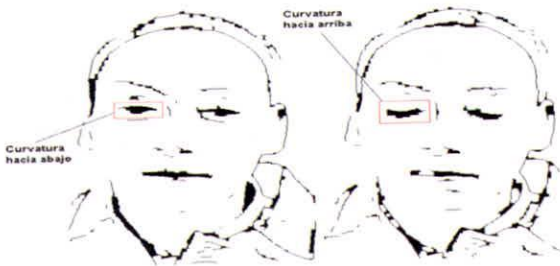
Una vez segmentadas todos los objetos de una imagen, utilizaremos la clase Descripción para calcular la circunferencia de área máxima y su centro.

Es necesario obtener el objeto de área máxima y desechar el resto ya que esto nos permitirá clasificar entre ojos abiertos y cerrados. Si se da la condición de ojos abiertos, dicha área será mayor que el caso de ojos cerrados.

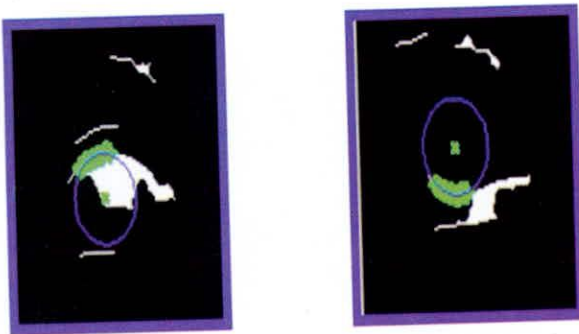
Algoritmo de clasificación entre ojos abiertos y cerrados

(Dirección de curvatura del objeto de área máxima)

Analizando las imágenes (Figura 2) podemos notar que en ojos abiertos, las curvaturas de los ojos están hacia abajo (el centroide estaría dentro o bajo del ojo), mientras que en ojos cerrados las curvaturas de los ojos están hacia arriba (el centroide estaría arriba del ojo). En las imágenes de la figura 2 se ha hecho una detección de bordes y utilizando los parámetros adecuados en las funciones analizadas anteriormente encontramos que es muy evidente y claro que cuando se tiene ojos cerrados la curvatura está hacia arriba y con ojos abiertos ésta se encuentra hacia abajo. Por tanto hemos decidido trabajar con este criterio para determinar si los ojos están abiertos o cerrados.



a) Imagen de todo el rostro con ojos abiertos y cerrados.



b) Imagen de ojos abiertos c) Imagen de Ojos cerrados

Figura 2. Dirección de curvatura del objeto de área máxima

Toma de decisión entre ojos abiertos o cerrados

Una vez obtenido el círculo es necesario determinar si esta imagen se encuentra con ojos abiertos o cerrados. Para eso utilizaremos el criterio mencionado anteriormente: si la curvatura es hacia abajo los ojos se encuentran abiertos; caso contrario si ésta se encuentra hacia arriba los ojos están cerrados.

Funcionamiento de los algoritmos Pantalla principal del sistema de visión artificial

En la figura 3 encontramos la pantalla principal, la cual nos dará las alternativas que tiene el programa, como son las de adquisición, pre procesamiento, segmentación, representación y descripción y el reconocimiento donde están las opciones de detección de ojos cerrados o abiertos en tiempo diferido y en tiempo real.



Figura 3. Interface principal del sistema de visión artificial

En condiciones normales de luz se muestra una persona tanto con ojos abiertos como con ojos cerrados, así como los resultados obtenidos por los algoritmos implementados.

Detección de Ojos abiertos o cerrados en tiempo diferido

En las figuras 3 y 4 mostramos las pantallas de detección de ojos abiertos y cerrados en tiempo diferido, lo cual nos permitirá seleccionar y trabajar solo con archivos almacenados en la computadora. Este programa consta de fotos almacenadas previamente, las cuales podemos seleccionar para su respectivo análisis.



Figura 4. prueba del sistema con ojos abiertos



Figura 5. Prueba del sistema con ojos cerrados

Pantalla de detección de ojos abiertos o cerrados en tiempo real

En la figura 5 tenemos la pantalla para la detección de ojos abiertos o cerrados en tiempo real.



Figura 6. Interface del sistema para procesamiento en tiempo real.

DISCUSION

Los puntos que analizaremos son los siguientes:

Influencia del pre procesamiento

La idea principal del pre procesamiento es mejorar las imágenes para poder realizar un mejor análisis de las mismas.

Las ideas principales que ponemos a discusión es si debemos utilizar filtros y ecualización en las mismas.

En nuestro proyecto no utilizamos filtros, ya que el hecho de usarlos no provocaba gran diferencia en las imágenes, y a su vez no producía grandes cambios al momento de procesarlas. La idea de usar o no filtros

continuará en discusión si se desea ir mejorando más detalladamente las imágenes y a su vez obtener resultados más específicos, pero en nuestro caso no fue necesario utilizarlos.

La ecualización fue reemplazada por el incremento de contraste, ya que este producía mejores resultados en las imágenes haciéndolas mucho más claras y convenientes para nuestro respectivo procesamiento. Claro que esta opinión puede variar dependiendo del tipo de intensidad de luz que tengamos en los diferentes lugares que se realicen las pruebas. Pero para nuestro caso no fue necesaria la ecualización en las imágenes, porque en algunos casos era de gran ayuda, mientras que en otros perjudicaba el proceso de análisis.

Limitaciones

Se han encontrado dos limitaciones principales en nuestro proyecto y son de gran interés como se muestran a continuación:

- **Uso de lentes:** En las limitaciones de nuestro proyecto encontramos que no se puede utilizar para personas que utilicen lentes o gafas de sol, ya que esto produce errores en el momento de la clasificación.
- **Zona de corte fija:** Una de las principales limitaciones que se tiene en este proyecto es el recorte previo que hay que hacer. Las coordenadas de la zona recortada son fijas y por tanto, para obtener una clasificación correcta, el usuario deberá ubicarse de modo que uno de sus ojos quede dentro del área recortada pues de lo contrario, el algoritmo fallará. Este inconveniente se tendría hasta encontrar la posición exacta de donde se encuentra el sector del ojo. Podemos también concluir que esto no tendría, demasiadas dificultades para un conductor ya que este no tiene movimientos bruscos desde el asiento de conducción.

CONCLUSIONES

1. Para determinar si los ojos están abiertos o cerrados, es válido usar el criterio de la curvatura del ojo.
2. El algoritmo desarrollado se basa en que si en dos o más imágenes consecutivas del sujeto se verifica que el conductor está con ojos cerrados, y si esto se produce a intervalos cada vez más frecuentes.

3. Una de las limitaciones que se tiene en el sistema es el hecho de que el análisis se hace en una zona determinada de la imagen, ya no funciona si la cabeza del conductor sale de la zona.
4. En ambientes con poca luz, se hace necesaria una cámara de video basada en infrarrojos.

AGRADECIMIENTO

A la Universidad Nacional Santiago Antúnez de Mayolo; por el financiamiento del proyecto de investigación.

REFERENCIAS BIBLIOGRÁFICAS

A. Branzan, B. Widsten, T. Wang, J. Lan, and J. Mah. 2008. *A computer vision-based system for real-time detection of sleep onset in fatigued drivers*. *IEEE*,

Intelligent Vehicles Symposium. 25–30.

Q. Ji and X. Yang. 2002. *Real-time eye, gaze and face pose tracking for monitoring driver vigilance*. *Elsevier Science Ltd., Real Time Imaging*. 1(8) 357–377.

T. Hamada, T. Ito, K. Adachi, T. Nakano and S. Yamamoto. 2003. *Detecting method for drivers' drowsiness applicable to individual features*, in *Proc. Intelligent Transportation Systems, IEEE*. 2(2)1405–1410.

Correspondencia:

Jesús Espinola Gonzales
espinolj@teleline.es