

Procesamiento de imágenes en tiempo real utilizando tecnología embebida. Aplicación en rehabilitación visual.

Marcelo Raponi^{1,2}, Ezequiel Pawelko³, Leandro Arcusin⁴, Claudio Delrieux⁵

¹ Escuela de Ciencia y Tecnología, ECyT-UNSAM, Buenos Aires, Argentina

² Asociación Argentina de Baja Visión, Mar del Plata, Buenos Aires, Argentina

³ Instituto Universitario de la Policía Federal Argentina, IUPFA, CABA, Argentina

⁴ Facultad Regional Buenos Aires, UTN, CABA, Argentina

⁵ Departamento de Ingeniería Eléctrica y Computadoras, UNS, IIIE (CONICET), Bahía Blanca, Buenos Aires, Argentina

raponimarcelo@gmail.com

Abstract. Se presenta un dispositivo de rehabilitación visual, capaz de mejorar la calidad de vida de pacientes con disfunciones visuales severas (que padecen visión subnormal o baja visión). El mismo permite adquirir y procesar imágenes en tiempo real, efectuar un realce selectivo de la información visual y traducir dicha información a un patrón de estimulación apropiado para cada paciente. El software fue enteramente desarrollado empleando librerías open-source (OpenCV y Qt) y está lo suficientemente modularizado como para permitir una rápida adaptación a nuevos dispositivos embebidos, y un efectivo rediseño ajustando la plataforma a la evolución de la patología del paciente. El sistema está diseñado para correr sobre plataformas embebidas, como la placa de desarrollo Beagleboard, lo cual lo hace fácilmente configurable, portátil y de bajo costo. Se describen las herramientas utilizadas para implementar los diferentes procesamientos y se presentan los resultados obtenidos al aplicar los mismos sobre señales reales.

Keywords: Baja visión, rehabilitación visual, sistemas embebidos, OpenCV.

1 Introducción

El sentido de la visión es fundamental para llevar a cabo las distintas actividades que cotidianamente realizamos los seres humanos. Tareas tan habituales como cruzar una calle, leer un libro, ver televisión, entre otras tantas, se ven seriamente afectadas por traumatismos y patologías oculares que producen ceguera parcial o completa en millones de seres humanos. A nivel mundial, un gran número de personas padecen disfunciones visuales severas sin llegar a ser completamente ciegos (condición conocida con el nombre de baja visión o visión subnormal). La OMS define baja visión como: "pérdida de agudeza visual (AV) y/o campo visual (CV), que incapacita para la realización de tareas de la vida diaria, incluso tras un tratamiento y/o corrección refractiva convencional". Pacientes con reducción severa del CV, presentan una movilidad disminuida (no logran evitar obstáculos o identificar defectos del terreno de manera efectiva). Si la visión periférica está intacta pero la AV

se encuentra comprometida, la realización de actividades que necesitan alta agudeza visual (como leer, escribir, coser, etc.) se ven seriamente afectadas. Mediante el empleo de elementos ópticos y sistemas electrónicos (magnificadores de imágenes, circuito cerrado de TV, etc.), es posible brindar un cierto grado rehabilitación visual a estos pacientes. El acceso a dichos dispositivos es generalmente restrictivo debido a su alto costo, y no siempre brindan un beneficio sustancial al paciente. En general, sólo compensan una de las deficiencias y no se adaptan a la evolución temporal de la patología.

En este trabajo se presenta el desarrollo de un dispositivo de rehabilitación visual diseñado para mejorar la calidad de vida de personas con disfunciones visuales severas. El sistema se caracteriza por ser reconfigurable, portátil y de bajo costo. El mismo permite adquirir y procesar imágenes en tiempo real, efectuar un realce selectivo de la información visual y mapear dicha información en un patrón de estimulación apropiado a la situación de cada paciente. El software se basa en plataformas libres (open-source) y está lo suficientemente modularizado como para permitir una rápida adaptación a nuevos dispositivos embebidos o librerías, y un efectivo rediseño ajustando la plataforma a la evolución de la patología del paciente.

2 Materiales y Métodos

El dispositivo desarrollado para asistir visualmente a personas con problemas visuales, consiste en un sistema reconfigurable basado en tecnología embebida y algoritmos de procesamiento de imágenes digitales. La plataforma consta de tres componentes: un módulo de adquisición de señales de video, un módulo de procesamiento del tipo ARM (Advanced RISC Machines) [1] y un módulo de visualización. La adquisición se efectúa mediante una mini-cámara (Logitech Pro-9000) de 1600x1200 pixeles de resolución espacial y foco ajustable, que se conecta a la unidad de control mediante un puerto USB 2.0. El dispositivo de salida consiste en unos videolentes (video eyewear EVG920V) que poseen dos displays LCD TFT gráficos de 640x480 pixeles (VGA), mediante los cuales se generan imágenes virtuales de aproximadamente 80" a 2 m de distancia. Ellos aceptan señales de video compuesto (NTSC/PAL/SECAM), consumen menos de 1.1 W y su batería dura unas 4 horas de uso. El corazón del sistema es una placa de desarrollo Beagleboard [2] que posee un núcleo de procesamiento ARM® Cortex™ A8 con frecuencia de trabajo de 600 MHz, 256 Mb de RAM, un DSP C64x+ y un acelerador gráfico. La placa tiene conectividad con periféricos (teclado y mouse USB), webcam, LCD, memorias SD, etc. Las pruebas preliminares del sistema demuestran que es posible realizar adquisición, procesamiento y visualización en tiempo real, dando resultados muy satisfactorios [3]. Para el diseño del dispositivo de rehabilitación visual se tuvo en cuenta una serie de requisitos, entre los cuales mencionaremos:

Tamaño físico: fue necesario reducir al mínimo el tamaño del equipo para mejorar su portabilidad y adaptarlo a un abanico más amplio de situaciones cotidianas. Para ello se optó por un *system on chip* que reúne múltiples funciones en una sola plataforma. En la Fig. 1 se pueden observar las dimensiones de la unidad de control (localizada en

el interior de una caja de acrílico especialmente diseñada) en relación con el tamaño de un mouse y un celular.



Fig. 1. Dimensiones de la unidad de control y procesamiento. La placa Beagleboard se encuentra localizada en el interior de una caja de acrílico diseñada a medida.

Consumo de energía: las dimensiones del dispositivo influyen a su vez en el tamaño de la batería, y por consiguiente, en el tiempo de uso sin necesidad de recarga. Los procesadores de tipo ARM son una buena opción debido a su gran capacidad de cálculo y su bajo consumo de energía.

Calidad de la imagen de entrada: las webcam estándares poseen sensores de baja resolución y alto blurring (difuminado, borronado) entre cuadros de la señal de video (frames). Para que el sistema opere de manera autónoma se requiere el diseño de un mecanismo de autoenfoco y una mini-cámara digital de resolución espacial aceptable (por lo menos 640x480).

Capacidad de procesamiento: las operaciones del pipeline gráfico, especialmente las de dewarping (para remover la distorsión esférica) y reconocimiento óptico de caracteres (OCR, Optical Character Recognition), requieren gran capacidad de cálculo. La placa de desarrollo posee una gran potencia de cálculo gracias a su DSP (Digital Signal Processing), su microprocesador de 32 bits y su aceleradora gráfica.

Latencia y usabilidad de interfaz gráfica: uno de los requerimientos del sistema es que su respuesta sea en tiempo real, por lo cual fue necesario minimizar la latencia, incluyendo en el diseño de las clases mecanismos de reducción de bloqueos de interfaz de usuario, implementando threads para las operaciones concurrentes.

Dentro de las principales características del software podemos mencionar: *reconfigurabilidad* (se adapta al paciente y al estadio de su enfermedad), *simplicidad*, *versatilidad* (multifunción), *amplia conectividad* (a video-lentes, monitor de PC,

televisor LCD), *zoom digital ajustable y autofocus*, *visión aumentada*, *OCR+TTS* (reconocimiento de caracteres y conversión de texto a speech), etc. La adquisición, el procesamiento y la visualización de las imágenes, involucra elementos del stack completo de software: un sistema operativo, drivers de kernel, librerías gráficas de bajo nivel, toolkits gráficos y herramientas del user space. Se optó por el sistema operativo embebido *Ångstrom* v2.6.32 [4] y la adquisición de imágenes se realizó a través de los drivers de *V4L2*, que son llamados desde la librería *highgui* de OpenCV. La librería OpenCV [5,6] se encarga de proveer las estructuras de datos y los elementos algorítmicos tanto iniciales como avanzados. En la Fig. 2 se presenta un esquema con los bloques que conforman la solución de software.

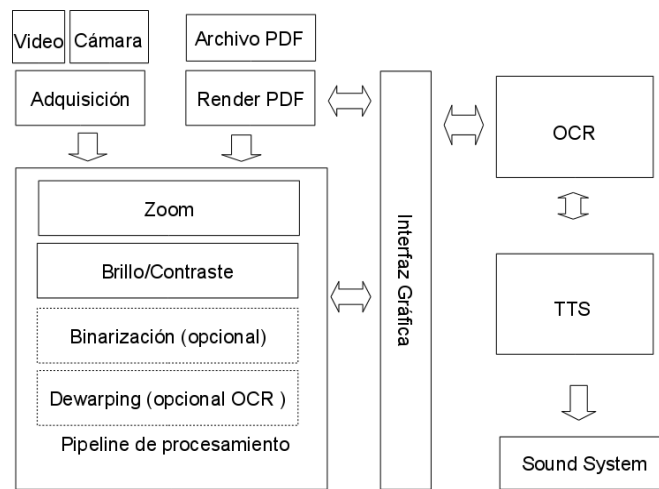


Fig. 2. Arquitectura del software desarrollado.

3 Resultados

Se implementaron una serie de algoritmos, algunos de los cuales describiremos a continuación. Se implementó un algoritmo para realizar *autofoco*, el cual consiste en determinar el operador F que da una idea del grado de enfoque actual de la cámara, debido a que el driver de Linux no proporciona uno. Para calcular F se emplea la ecuación 1:

$$F = \left(\frac{\sum |S(I)|}{n} \right) \quad (1)$$

donde S es un filtro Sobel de dimensión 3×3 que se aplica sobre cada cuadro de la señal de video (matriz I). Se calcula el valor promedio de la imagen filtrada, siendo n la cantidad de píxeles. El valor de F se relaciona con la presencia de altas frecuencias espaciales, cuanto más enfocada está la imagen filtrada, mayor es el valor de F . La

webcam seleccionada permite al usuario administrar su enfoque a través del envío de números entre 0 y 255. Con el fin de generar un autofocus, se ideó un algoritmo para encontrar el mejor enfoque en una cantidad iteraciones mínimas. A continuación describimos el algoritmo de cálculo secuencial del foco óptimo:

```
Setear variable max=0.0
Setear maxFoco=0
para (i=MinFocusValue;i<= MaxFocusValue;i=i+Step)
    { SetearFoco(i)
      capturar imagen
      derivada = Sobel (imagen)
      resultado = mean ( abs ( derivada ) )
      if (resultado > max) {
          max=current;
          maxFoco = i ; };
```

Pseudocódigo de método de cálculo de imagen mejor enfocada.

Una mejora al algoritmo de autofocus sería emplear una técnica de búsqueda en zig-zag, con el fin de evitar volver al punto inicial en cada iteración, pudiendo cambiar de esta manera el sentido de testeo del enfoque. En la Fig. 3 se puede apreciar el resultado de aplicar el algoritmo autofocus.

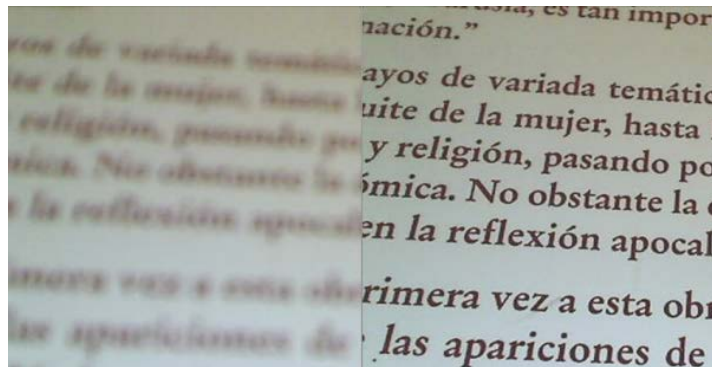


Fig. 3. Efecto de la aplicación del algoritmo de autofocus sobre la nitidez de texto.

Con el fin de mejorar la distinción de los rasgos sobresalientes (features) de las imágenes capturadas, se buscó una combinación de algoritmos que permita resaltar los bordes de los objetos presentes en la escena visual. El procesamiento se efectúa en tres etapas: primero se realiza un difuminado (blurring) para reducir los efectos de la compresión jpg que genera artefactos de blocking móviles, luego se aplica un umbralizado (thresholding) adaptativo y finalmente un operador morfológico (erosión), con el fin de acentuar las divisiones entre las características resaltadas (Fig. 4). Como operación extra, los valores extremos de luminancia (que superan el umbral) se igualan a un valor intermedio preservando la información específica de

color, y de esta manera evitar la generación de anillos (ringing) que se pueden esperar para zonas con excesiva diferencia de luminancia.

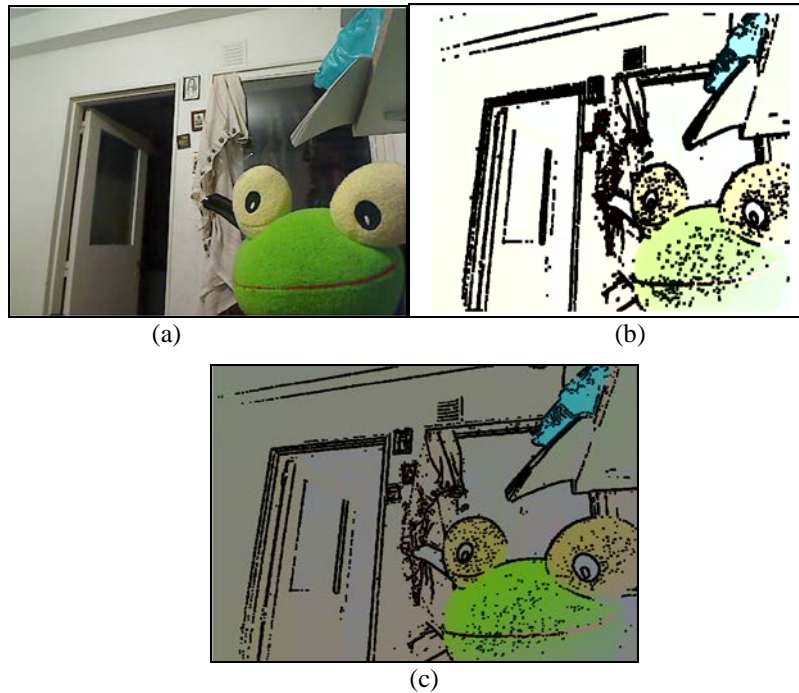


Fig. 4. a) Imagen original, b) Imagen con máxima luminancia y aplicación del operador morfológico erosión de dimensión 5x5, c) Imagen con luminancia media y erosión de 3x3.

Para resaltar los bordes de los objetos en las escenas visuales se implementó un filtro Canny. En la Fig. 5 se puede observar la interfaz del prototipo en modo de adquisición con webcam y el resultado de aplicar a dicha señal de video, en tiempo real, un filtro Canny para detectar bordes (otros filtros que se pueden aplicar son el filtro Laplaciano, el filtro Sobel).

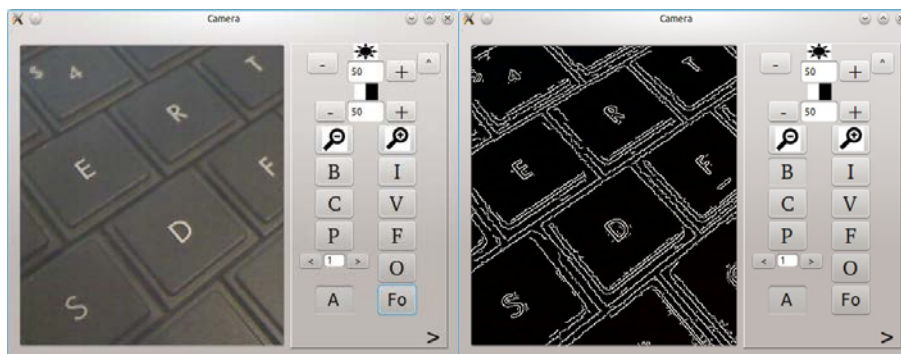


Fig. 5. Operación detección de bordes aplicada sobre la señal de video capturada por la webcam: (izq.) Imagen original y (der.) Imagen filtrada

Otra de las funciones desarrolladas es el reconocimiento de textos *OCR (Optic Character Recognition)* y su traducción a un archivo que será leído por medio de un engine de Text to Speech, *TTS* (Fig. 6). Para que el software de reconocimiento pueda cumplir su papel, es necesario que reciba las features gráficas sustancialmente diferenciadas del fondo y una imagen con un mínimo de transformaciones. Entre de los algoritmos de preprocesamiento destinados a la realización de OCR, el operación de binarizado cumple un papel fundamental debido a que es el encargado de separar del fondo del documento las rasgos característicos correspondientes a los caracteres.

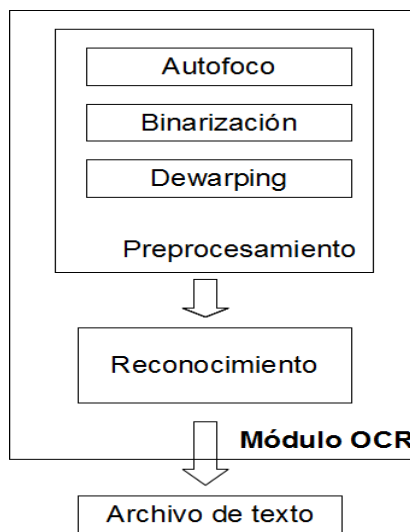


Fig. 6. Algoritmos de preprocesamiento de imágenes para OCR

Dentro de los esfuerzos realizados para la binarización de imágenes documentales, el algoritmo de Sauvola [7] tiene un rol preponderante por su sencillez y efectividad. Para ésta técnica, el umbral de binarización $t(x,y)$ es calculado utilizando la media $\mu(x,y)$ y la desviación standard $\sigma(x,y)$ de las intensidades de los pixeles en una ventana W_{xy} centrada alrededor del pixel con coordenadas x,y (ecuación 2):

$$t(x,y) = \mu(x,y) \left[1 + k \left(\frac{\sigma(x,y)}{R} \right) - 1 \right] \quad (2)$$

donde R es el valor máximo de la desviación standard ($R=128$ para un documento de escala de grises) y k es un parámetro que toma valores positivos. La implementación computacional actual de dicha binarización es una variación eficiente de la misma, implementada en el paquete OCRopus (<http://code.google.com/p/ocropus/>). La Fig. 7 muestra una comparación entre los algoritmos de umbral adaptativo - utilizado en la etapa 2 de la binarización tradicional - y el de Sauvola, sobre una imagen de un texto

comprimida con el formato jpg (lo cual es común en frames de webcam). Como se puede apreciar, el algoritmo de Sauvola es muy resistente a las perturbaciones locales, como los efectos de ringing y blocking propios de los archivos jpg.

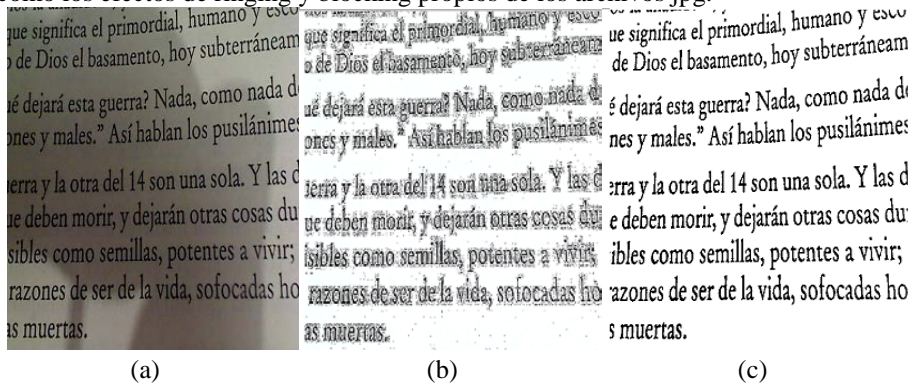


Fig. 7. Comparación de los algoritmos de binarización: (a) imagen Original, (b) umbral adaptativo y (c) algoritmo de Sauvola.

Además de los procesamientos mencionados previamente, la interfaz desarrollada permite manipular documentos del tipo pdf (formato de documento portátil). En la Fig. 8 se puede observar una partitura en formato pdf, sobre la cual se pueden aplicar diferentes procesamientos, como por ejemplo, inversión de colores, detección de bordes, etc. Cabe destacar que el usuario puede desplazarse por las diferentes páginas del documento, empleando un simple control localizado en la barra lateral desplegable.

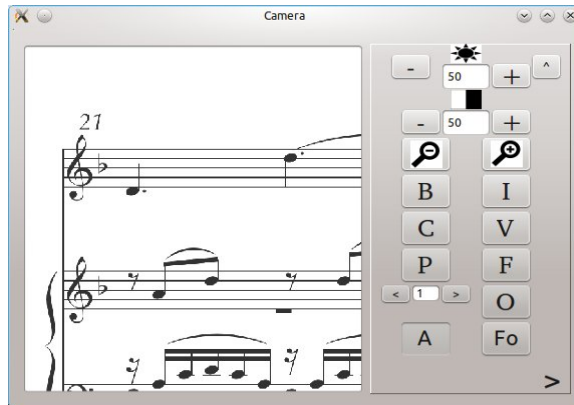


Fig. 8. Visualización de un archivo en formato PDF (partitura musical)

Un importante algoritmo desarrollado es el de *visión aumentada*, el cual proporciona a los pacientes con visión túnel, información necesaria acerca de objetos ubicados en su periferia (fuera del campo visual del paciente, el cual varía según el estadio de la enfermedad), permitiendo mejorar la movilidad de los mismos, sin comprometer la visión central residual. Para conformar las imágenes a ser visualizadas en los display

portátiles, los cuadros adquiridos de la señal de video son filtrados utilizando un detector de bordes. En un mismo frame se presenta la imagen original ampliada (zoom) y los bordes de los objetos (en color blanco, con posibilidad de seleccionar otra tonalidad, según gusto del usuario) de la escena visual original (que presenta el mayor campo de visión posible) (Fig. 9). El objetivo de esta herramienta computacional es la de brindarle al paciente información sobre los objetos dentro de un campo visual amplio (con menor resolución o detalle, pero suficiente para la navegación), y a la vez, seguir disfrutando de la alta resolución de su visión central residual.

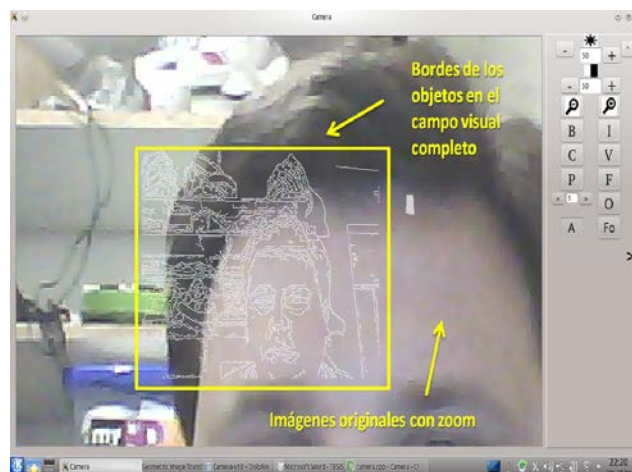


Fig. 9. Modo de visualización configurado para visión aumentada.

La captura de imágenes desde una cámara sin fijar, posee comúnmente deformaciones debidas a múltiples fuentes: perspectiva desde la cual se tome, curvatura de las lentes de webcams (diseñadas para tomar rostros desde una distancia de más de 30-40 cm), y los efectos de deformación debido a la encuadernación. Con el fin de resolver dichas perturbaciones, se están desarrollando dos procesamientos: una corrección de la relación de aspecto por medio de la estimación de un trapecio contenedor y sus 4 puntos de control asociados, y un algoritmo orientado a la corrección de curls (como el basado en coupled snakelets de Bukhari [8]).

Conclusiones

Se presentó el desarrollo de un dispositivo de rehabilitación visual destinado a mejorar la calidad de vida de pacientes con disfunciones visuales severas. El dispositivo está diseñado para poder adquirir y procesar imágenes en tiempo real, efectuar un realce selectivo de la información visual y traducir dicha información a un patrón de estimulación apropiado para cada paciente. Se mostraron los detalles de implementación del hardware y los algoritmos empleados. El desarrollo utiliza únicamente bibliotecas open source, y está diseñado para permitir una rápida

adaptación a nuevos dispositivos embebidos y a diferentes patologías. Se presentaron los resultados preliminares obtenidos en situaciones reales, los cuales muestran que el desarrollo es efectivamente aplicable como dispositivo de rehabilitación visual, y actualmente se encuentra en etapa de testeo con pacientes voluntarios. Un trabajo futuro que se está considerando es la utilización de implantes neuroestimuladores para poder utilizar este dispositivo en casos de pérdida completa de visión.

Referencias

1. ARM, descripción del procesador Cortex A8. (www.arm.com/products/processors/cortex-a/cortex-a8.php)
2. Coley, G.: BeagleBoard System Reference Manual Rev C4 (2009). (www.beagleboard.org/static/BBSRM_latest.pdf)
3. Raponi, M.M., Bonnin, R.O.: Dispositivo de rehabilitación visual basado en sistemas embebidos del tipo ARM. Congreso Argentino de Sistemas Embebidos, CASE. UTN-FRBA, Argentina. ISBN 978-987-9374-69-6, pp. 85--88 (2011)
4. The Ångström Distribution Embedded power (www.angstrom-distribution.org)
5. Bradski, G., Kaebler, A.: Learning OpenCV. Ed. O'Reilly Media Inc., ISBN 978-0-596-51613-0 (2008)
6. Laganière, R.: OpenCV 2 Computer Vision Application Programming Cookbook. Ed. Packt Publishing. ISBN 978-1-849513-24-1 (2011)
7. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recognition, 33 (2), pp. 225--236 (2000)
8. Bukhari, S.S., Shafait, F., Breuel, T.M.: Coupled Snakelet Model for Curled Textline Segmentation of Camera-Captured Document Images. Proc. 10th Int. Conf. on Document Analysis and Recognition, ICDAR'09, Barcelona, Spain (2009)