Aplicación turística para dispositivos móviles basada en técnicas de visión computacional

Pablo A. Sosa†, Enrique M. Albornoz†‡ y César E. Martínez†§

† Centro de Inv. en Señales, Sistemas e Inteligencia Computacional (SINC(i))
Facultad de Ingeniería y Ciencias Hídricas - Universidad Nacional del Litoral
CC217, Ciudad Universitaria, Paraje El Pozo, S3000, Santa Fe, Argentina
‡ CONICET, Argentina
§Laboratorio de Cibernética, Universidad Nacional de Entre Ríos
{psosa@quboo.com.ar,emalbornoz@fich.unl.edu.ar,cmartinez@fich.unl.edu.ar}

Resumen En la actualidad, los dispositivos móviles se han convertido en una herramienta útil que brinda al usuario una amplia variedad de funcionalidades. Además, la evolución tecnológica de éstos ha permitido que puedan ejecutarse algoritmos muy exigentes como aquellos utilizados en procesamiento de imágenes, reconocimiento de patrones, etc. Este contexto ha impulsado un nuevo campo de investigación y desarrollo, mientras que genera un nicho de mercado que apenas ha comenzado a explotarse a nivel regional. En este trabajo se presenta el desarrollo de una aplicación de información turística para dispositivos móviles que incorpora técnicas de visión computacional. Se trabajó sobre el circuito turístico del "Camino de la Constitución" de la ciudad de Santa Fe. El sistema realiza el reconocimiento automático de imágenes de los diferentes mojones informativos y provee al usuario una vasta cantidad de información.

Palabras claves: Turismo, OpenCV, Android, OCR.

1. Introducción

En los últimos años, se ha vuelto masiva la utilización de dispositivos móviles inteligentes. Su evolución tecnológica es tal que su capacidad de cómputo prácticamente iguala a las computadoras de escritorio, permitiendo ejecutar algoritmos cada vez más exigentes como los utilizados en procesamiento de imágenes, visión computacional, reconocimiento de patrones, entre otros [1]. La comunidad científica ha acompañado esta evolución generando un nuevo campo de investigación y desarrollo, migrando sus aplicaciones desde las computadoras de escritorio e inclusive diseñandolas para ser utilizadas dispositivos móviles como Tablets o Smartphones [2]. No obstante, el avance a nivel académico de este área no se corresponde con el mismo a nivel comercial, donde las aplicaciones usualmente son más básicas y no se ha logrado explotar el consumo masivo por parte de la población. Existen aplicaciones comerciales que realizan procesamiento de imágenes y a partir de la identificación de objetos de interés retornan información de utilidad. Entre las más conocidas se pueden citar *Google Goggles*, *Wikitude* y *Layar*.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

Sin embargo, las dos últimas realizan este trabajo mediante la utilización del *sistema de posicionamiento global* (GPS: del inglés Global Positioning System.) del dispositivo móvil [3]. Google Goggles es un servicio de Google disponible para Android e iPhone que permite reconocer objetos mediante fotos capturadas con un dispositivo móvil y devolver resultados de búsqueda e información relacionada. Este sistema reconoce lugares, monumentos y textos, entre otras cosas. Su funcionamiento consiste en apuntar con la cámara del dispositivo móvil a un lugar conocido, un producto, un código de barras o un código de respuesta rápida (QR: del inglés quick response code). Luego, si Google lo encuentra dentro de su base de datos, ofrece información relacionada [4]. Wikitude es un software de realidad aumentada (RA)[1] para dispositivos móviles que fue desarrollado por la compañía austríaca *GmbH* y publicado en octubre de 2008 como software gratuito. Para obtener la localización de los objetos en la RA se utiliza la posición del usuario a través del GPS y la dirección en la que el usuario mira mediante el uso de la brújula y el acelerómetro. Esta aplicación depende de la precisión del GPS, que a veces no es lo suficientemente exacto [5]. Layar es un software que utiliza el GPS y la brújula de los dispositivos Android para ubicar la posición del usuario y su orientación. La cámara del dispositivo captura el entorno y reproduce la imagen en la pantalla, mientras que el software adiciona información de lugares tales como cafeterías, restaurantes, cines, etc. [6].

En este trabajo se presenta el diseño y desarrollo de una aplicación para dispositivos móviles con sistema operativo Android [7] basada en técnicas de visión computacional. La aplicación pretende ser una herramienta útil para los visitantes de la ciudad de Santa Fe, que visiten el circuito turístico del "Camino de la Constitución". Este es un recorrido museológico que integra 18 sitios y edificios de valor simbólico y arquitectónico, con el objetivo de reconstruir y narrar la historia de la relación de la ciudad de Santa Fe con la Constitución Nacional. La propuesta integra diversas funciones: histórica, cultural, educativa y turística, y pone de relieve nuestra vida política pasada y presente relacionada con los diferentes procesos y acontecimientos vinculados con la Carta Magna[2].

Una de las consignas fundamentales de este trabajo fue la de utilizar los mojones tal y como son actualmente, sin modificar su estética, emplazamiento, iluminación, etc. Entonces, en un primer paso se definió y realizó un importante relevamiento fotográfico utilizando diversos dispositivos móviles y condiciones de iluminación naturales, para generar una base de datos de imágenes de los mojones informativos. Luego se exploraron técnicas de preprocesamiento, análisis y clasificación de imágenes, orientadas a obtener el mejor desempeño manteniendo la simplicidad y una carga computacional baja, a fin de lograr una aplicación veloz. Finalmente, se diseñó una interfaz gráfica que permite al usuario tomar una fotografía del mojón (Figura 1) y obtener información turística relacionada.

[1] RA es el término que se usa para definir una visión directa o indirecta del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta en tiempo real.

[2] Más información en http://santafeciudad.gov.ar/cunadelaconstitucion/

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

Figura 1: Mojón informativo. (Imagen tomada de http://santafeciudad.gov.ar)

El resto del trabajo se organiza como se detalla a continuación. En la Sección 2 se describen los materiales y métodos utilizados. Mientras que la Sección 3 presenta el método propuesto considerando diferentes técnicas. Finalmente, en la Sección 4 se encuentran los resultados y conclusiones.

2. Materiales y tecnologías empleadas

El primer paso fue diseñar un protocolo para tomar las fotografías y generar una base de datos de imágenes. Se tomó un gran número de fotografías de los 18 mojones del Camino de la Constitución de la ciudad de Santa Fe, considerando siempre la mayor naturalidad posible en la toma para poder lograr el sistema más robusto posible. Se consideraron las siguientes condiciones:

1. Rotación: las fotos se capturaron con diferentes rotación hacia la izquierda o derecha.
2. Orientación del dispositivo: el dispositivo estuvo ubicado en forma vertical u horizontal, indistintamente.
3. Independencia del dispositivo: las capturas se realizaron para distintos dispositivos móviles:
   a) Smartphone Samsung Galaxy i5550, sistema operativo Android 2.3.
   b) Tablet ASUS Transformer TF101.CPU NVIDIA Tegra 2 1.0GHz. Android 3.2 Honeycomb O.S.
4. Múltiples resoluciones: las fotografías fueron tomadas con 1.3, 2.3 y 5 Megapíxeles.
5. Condiciones de iluminación: se consideraron condiciones naturales sin iluminación artificial y se tomaron de mañana, de tarde y de noche.

La base de datos conformada consiste en un conjunto de 693 fotografías. Éstas se utilizaron tanto para el preprocesamiento como para la clasificación de las imágenes.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

La aplicación se desarrolló en Android para la versión 2.3 o superior, que posee compatibilidad con la biblioteca OpenCV[3]. Más específicamente, se utilizó Java [8] para desarrollar la lógica de la aplicación y XML para la interfaz de usuario, utilizando el entorno Eclipse [9]. Con respecto al desarrollo de rutinas para el procesamiento de imágenes se decidió utilizar la biblioteca OpenCV (versión 2.4.2) que es Open Source y multiplataforma [10]. Ésta provee un conjunto de funciones para procesamiento y análisis de imágenes, permite trabajar con video y tiene una versión disponible para Android.

3. Método propuesto

En esta sección se describe primeramente el preprocesamiento de las imágenes y la extracción de características, y luego se presenta la evaluación de varias alternativas para la clasificación de las imágenes. Finalmente, se presenta el desarrollo de la interfaz gráfica con la que el usuario interactúa.

3.1. Preprocesamiento de la imagen

En esta etapa podemos definir dos fases, en la primera se busca normalizar las imágenes tomadas desde el dispositivo móvil y en la segunda se realiza la extracción de la información distintiva del mojón.

Normalización de imágenes

Luego de evaluar varias alternativas, se obtuvo una secuencia de operaciones que permite preprocesar cualquier imagen para normalizarla:

1. Escalar la imagen: considerando el compromiso entre la velocidad de cómputo y los detalles en la imagen, el tamaño elegido es 800x600 píxeles.
2. Convertir a escala de grises: se realiza porque los procesos posteriores no precisan información del color y así se manipula menos información.
3. Ecualizar el histograma: se realiza para mejorar el contraste. Redistribuye de la forma más uniforme posible los grises de la imagen original sobre el total de intensidades [11].
4. Binarizar: es útil para la posterior detección de bordes.
5. Detectar bordes: se obtienen los bordes de la imagen aplicando el método de Canny [12]. Con esta técnica se obtuvieron mejores resultados que con el detector de bordes de Sobel [11].
6. Encontrar líneas principales: se utiliza la *transformada de Hough* que considera las relaciones globales entre píxeles permitiendo encontrar ciertos patrones en la imagen como líneas y círculos [13]. Aquí se determina el ángulo de rotación de la imagen capturada utilizando características particulares (marcos de los mojones) presentes en todas las imágenes.
7. Rotar: se rota la imagen utilizando el ángulo obtenido previamente.

En la Fig. 2 se puede ver el resultado de aplicar los pasos de la normalización.

[3] http://opencv.org/platforms/android.html.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

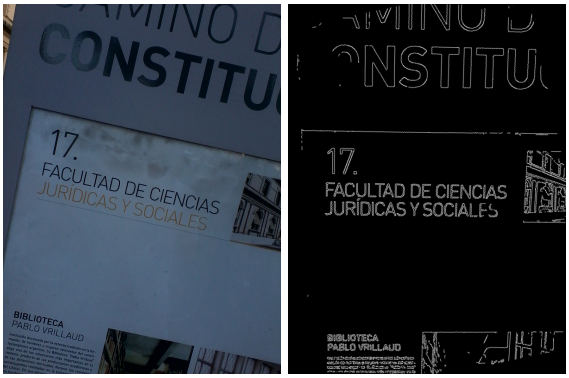


Figura 2: Imagen original y normalizada.

(a) Recorte de la zona de interés.

(b) Imagen de sólo números.

(c) Imágenes con los dígitos recortados.

Figura 3: Imágenes de la extracción del identificador.

Extracción del identificador único

Luego de realizar la normalización, se procede a extraer el número que fue el patrón escogido como identificador único para cada mojón. Para ésto se realiza el siguiente procedimiento:

1. Localizar: se extrae el cuadrante superior izquierdo de la imagen.
2. Dilatar: se realiza un proceso de dilatación con una máscara de 2x2, para que las líneas de los marcos de los mojones queden bien definidas, y así, facilitar el recorte del identificador.
3. Recortar zona de interés: se buscan las líneas horizontal y vertical correspondientes a los marcos de los mojones para hacer el recorte, ya que éstas son características que siempre están presentes en la imagen. En primer lugar se utilizan acumuladores de píxels no nulos a lo largo de las filas para encontrar la línea más extensa, luego se aplica el mismo proceso sobre las columnas. Es importante el orden de estas operaciones para salvar casos de mojones con perspectivas. El resultado de esta etapa se ve en la Figura 3a.
4. Erosionar: se realiza una erosión para que los dígitos esten bien separados.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

5. Extraer número: se utiliza el acumulador de píxeles no nulos sobre filas para obtener la imagen del número basado en los espacios anterior y posterior (Fig. 3b).
6. Recortar los dígitos: se realiza utilizando el método de componentes conectadas [11]. Se seleccionan las dos componentes más grandes y se evalua la relación de aspecto de cada una para descartar ruido. En la Figura 3c se observa un ejemplo del resultado final.

3.2. Extracción de características

En esta etapa se extraen características relevantes de las imágenes de los dígitos. A continuación se presentan las alternativas consideradas.

Momentos invariantes de Hu

Los momentos invariantes de Hu representan una serie de características de los objetos independientemente de su posición, escala o rotación [11]. Hu obtuvo sus invariantes a través de los momentos geométricos. Éstos se definen como :

$$u_{pq} = (x - \hat{x})^p (y - \hat{y})^q f(x, y) dx dy \tag{1}$$

donde u_{pq} es el momento geométrico de orden $(p + q)$, $f(x, y)$ es el valor del píxel en la posición (x, y) y $(\hat{x}, \hat{y})$ son las componentes del centroide. Partiendo de los momentos definidos anteriormente, se definen los coeficientes n_{pq} que son invariantes al escalamiento y su expresión viene dada por:

$$n_{pq} = \frac{u_{pq}}{u_{00}^{1+\frac{p+q}{2}}} \tag{2}$$

Utilizando n_{pq} se obtienen las expresiones matemáticas que dan lugar a los siete momentos invariantes de Hu.

Vector de distancias al primer píxel no nulo de una imagen.

Este método utiliza la imagen binaria del dígito obtenida previamente. Se crea un vector columna de longintud igual al alto de la imagen cuyos valores, para cada fila, representan las distancias al primer píxel no nulo de la imagen, en el sentido izquierda a derecha. Luego, se crea otro vector considerando las distancias de derecha a izquierda. Esta información es representativa del contorno del número en la imagen[4].

[4] Para esta etapa, las imágenes están normalizadas en 125x65 píxeles.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

3.3. Métodos de Clasificación

A partir de imágenes limpias y manualmente ajustadas se calcularon los patrones de referencia para los momentos invariantes de Hu y vectores de distancias no nulos. El paso siguiente es comparar las características calculadas para una nueva captura con las características de los patrones de referencia. Para realizar ésto, y considerando la velocidad de cómputo, se decidió clasificar utilizando una medida de distancia basada en la *norma-2*, así el patrón será clasificado considerando la menor distancia euclídea:

$$\text{Clase}_{\text{Y}} = \min_{j} \left\{ \sqrt{\sum_{i=1}^{n} (\mathbf{X}_i^j - Y_i)^2} \right\} \tag{3}$$

donde $\mathbf{X}^j$ son los vectores de referencia, Y el patrón a clasificar y n la dimensión de las características.

Método de clasificación alternativo

Como alternativa de clasificación se consideró la utilización del reconocimiento óptico de caracteres (OCR: del inglés Optical Character Recognition). Se investigaron varias alternativas de OCR y finalmente se seleccionó Tesseract OCR [14]. Ésta es una biblioteca libre con licencia *Apache License 2.0* escrita en C++, es muy eficiente y ampliamente utilizada[5].

Aquí también se realiza el preprocesamiento completo para obtener los dígitos, sin embargo, se recortan los dígitos a color para ser utilizados en el OCR. El uso de las imágenes de sólo dígitos mejora el rendimiento del OCR.

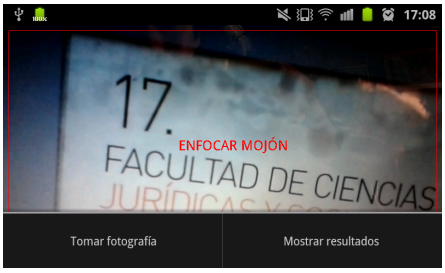
3.4. Interfaz de usuario

Una consideración general es que la información multimedia que provee la aplicación no se almacena en el dispositivo, a fin de utilizar la menor cantidad de recursos posibles, sino que se obtiene directamente desde la web. Por lo tanto, se requiere acceso a Internet mediante 3G o Wi-Fi para poder visualizar las imágenes, videos mediante streaming y enlaces a páginas web. El diseño de la interfaz de usuario es simple y permite rápidamente acceder a la información relacionada. La pantalla inicial permite acceder al sistema o encontrar ayuda acerca de cómo utilizar el sistema (Figura 4). El sistema solicita al usuario que tome una fotografía del mojón (Fig. 5a) e inmediatamente después de realizar el procesamiento y clasificación, arroja el resultado del mojón reconocido (Fig. 5b). En una etapa posterior el sistema brinda información en diversas alternativas multimedias (Fig. 6).

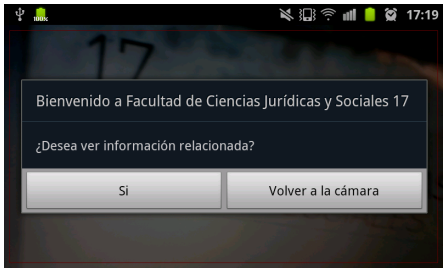
[5] Disponible en https://code.google.com/p/tesseract-ocr/.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

Figura 4: Pantalla inicial.



(a) Captura de imagen.



(b) Mojón reconocido.

Figura 5: Capturas del sistema.

Figura 6: Información multimedia relacionada.

4. Experimentos y resultados

A continuación se presentan algunos resultados preliminares sobre los diferentes métodos propuestos. Para este experimento se utilizaron 75 imágenes (aproximadamente 5 de cada mojón), capturadas con iluminación natural (de mañana y tarde) y con el dispositivo Samsung Galaxy en 1.3 mpx. Se obtuvo un $78,6\,\%$ de segmentaciones correctas hasta la etapa de recorte de los dígitos. Luego, se evaluó el tiempo total de los tres métodos de clasificación, ya que el

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.

Tabla 1: Resultados de clasificación.

Método de clasificación	Aciertos	Tiempo de ejecución
Vector de distancias a píxeles no nulos	100 %	5475 milisegundos
Momentos invariantes de Hu	66.1 %	1329 milisegundos
Tesseract OCR	100 %	298 milisegundos

tiempo de preprocesamiento (hasta el recorte de los dígitos inclusive) es el mismo para todos los casos. También se registró la tasa de aciertos en el reconocimiento de los diferentes mojones. En la Tabla 1 se pueden ver algunos resultados preliminares de los tres métodos de clasificación. Estos resultados son promedios de evaluar los 59 casos en que la segmentación fue exitosa.

Los momentos invariantes de Hu mostraron cierta inestabilidad, a pesar de ser invariantes a la rotación y escalado y los resultados obtenidos son inferiores a los esperados. El método basado en las distancias demostró ser muy bueno a pesar de su sencillez, aunque requiere mucho tiempo de cómputo. Finalmente, el OCR demostró ser el que obtuvo más aciertos y respecto de los tiempos de ejecución, también Tesseract OCR fue le de mejor rendimiento. Evidentemente, la opción de clasificación elegida para la aplicación final es Tesseract por su desempeño y velocidad de cómputo.

5. Conclusiones y trabajos futuros

En este trabajo se ha presentado una aplicación turística para sistemas Android basada en técnicas de visión computacional. La aplicación reconoce automáticamente las imágenes de los mojones del circuito turístico *Camino de la Constitución* de la ciudad de Santa Fe y devuelve información multimedial relacionada. Se ha realizado una base de datos de imágenes considerando diferentes condiciones de iluminación, resoluciones de la cámara y dispositivos de captura. Se han evaluado diferentes alternativas de procesamiento de imágenes para lograr un adecuado preprocesamiento de las fotografías obtenidas por el usuario y se evaluaron diferentes clasificadores. Se diseñó una interfaz simple e intuitiva para la aplicación. Finalmente, se ensambló el sistema considerando los mejores resultados obtenidos en la etapa de desarrollo-evaluación para lograr una aplicación simple, potente y veloz.

Como trabajo futuro se propone mejorar el método de preprocesamiento para lograr una mayor tasa de segmentaciones correctas. También se considera hacer una evaluación más profunda del comportamiento del sistema considerando otras condiciones en la captura la fotografía, por ejemplo con imágenes nocturnas.

Agradecimientos

Los autores desean agradecer a la *ANPCyT* y *Universidad Nacional de Litoral* (proyectos PAE 37122, PACT 2011 #58, CAI+D 2011 #58-511) y al *CONICET*, por su apoyo.

Referencias

1. Daniel Wagner, Gerhard Reitmayr, Alessandro Mulloni, Tom Drummond, and Dieter Schmalstieg. Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, pages 125–134, Washington, DC, USA, 2008. IEEE Computer Society.
2. Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001.
3. Zornitza Yovcheva, Dimitrios Buhalis, and Christos Gatzidis. Overview of smartphone augmented reality applications for tourism. *e-Review of Tourism Research*, 10(2):63–66, 2012.
4. Google Goggles. `http://www.google.com/mobile/goggles/`. Último acceso: Mayo-2013.
5. Wikitude. `http://www.wikitude.com/en/`. Último acceso: Mayo-2013.
6. Layar - Augmented Reality Browser. `http://www.layar.com/`. Último acceso: Mayo-2013.
7. Nisarg Gandhewar and Rahila Sheikh. Google Android: An emerging software platform for mobile devices. *International Journal on Computer Science and Engineering*, 1(1):12–17, 2010.
8. Javier Garca de Jaln de la Fuente. *Aprenda Java como si estuviera en primero*. Aprenda ..., como si estuviera en primero. Universidad de Navarra. Escuela Superior de Ingenieros Industriales, 1999.
9. Jeff McAffer, Jean-Michel Lemieux, and Chris Aniszczyk. *Eclipse Rich Client Platform*. Addison-Wesley Professional, 2nd edition, 2010.
10. Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.
11. Rafael Gonzalez and Richard Woods. *Digital Image Processing (2nd Edition)*. Prentice Hall, 2002.
12. John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.
13. Richard Duda and Peter Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
14. Ray Smith. An overview of the Tesseract OCR engine. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
P. Sosa, E. M. Albornoz & C. E. Martínez; "Aplicación turística para dispositivos móviles basada en técnicas de visión computacional"
XIX Congreso Argentino de Ciencias de la Computación (CACIC), oct, 2013.