LA LIBRERÍA DE VISIÓN ARTIFICIAL OPENCV APLICACIÓN A LA DOCENCIA E INVESTIGACIÓN[1]

V. M. Arévalo, J. González, G. Ambrosio
Dpto. De Ingeniería de Sistemas y Automática, Universidad de Málaga. España
varevalo@ctima.uma.es

En este artículo describimos las características fundamentales de la librería de visión artificial y código abierto *The Open Computer Vision Library* (OpenCV a partir de ahora). La librería OpenCV proporciona un marco de trabajo de alto nivel para el desarrollo de aplicaciones de visión por computador en tiempo real: estructuras de datos, procesamiento y análisis de imágenes, análisis estructural, etc. Este marco de trabajo facilita en gran manera el aprendizaje e implementación de distintas técnicas de visión por computador, tanto a nivel docente como investigador, aislando al desarrollador de las peculiaridades de los distintos sistemas de visión. De igual forma, presentamos algunos ejemplos de la utilización de dicha librería en docencia e investigación por parte del Dpto. de Ingeniería de Sistemas y Automática de la Universidad de Málaga.

Palabras clave: *Software libre*, visión por computador, procesamiento de imágenes, docencia, investigación, etc.

1. Introducción

La necesidad de disponer de un paquete de visión por computador y procesamiento de imágenes lo suficientemente flexible como para proporcionar herramientas de alto nivel para el desarrollo de las prácticas docentes que imparte el Dpto. de Ingeniería de Sistemas y Automática de la Universidad de Málaga; y la potencia, escalabilidad y disponibilidad del código fuente para el desarrollo de los distintos proyectos fin de carrera y tesis doctorales que dirige dicho departamento, propició una búsqueda incesante de software que cumpliera dichas características y sobre todo que fuese gratuito, estuviese soportado/revisado por personal cualificado y además se le intuyese una larga vida.

Son muchos los paquetes de procesamiento de imágenes comerciales y *software libre* disponibles actualmente, y muchas las ventajas e inconvenientes de cada uno de ellos.

Entre los distintos paquetes comerciales disponibles actualmente destacan por su potencia *The Matrox Image Library* (MIL) [1], Khoros, eVision, HIPS, Exbem, Aphelion, etc. sin embargo, el principal inconveniente es su elevado precio y su ciclo de actualización, muchas veces, relativamente largo. Algunos de ellos carecen de un entorno de desarrollo de alto nivel (i.e. HIPS), otros disponen de éste, pero están ligados a la plataforma de desarrollo (i.e. Khoros - Unix, Linux; Exbem - MacOS; eVision, Aphelion - Windows) o al propio hardware de captura (i.e. MIL). Todos ellos proporcionan funciones de procesamiento y análisis de imágenes, reconocimiento de patrones, estadísticas, calibración de la cámara, etc. a través del propio entorno o a través de librerías de funciones, desarrollados en la mayoría de las ocasiones en C/C++. Sin embargo, tan sólo HIPS pone a disposición del cliente su código fuente, y en la mayoría de los casos hablamos de librerías monolíticas, muy pesadas y no demasiado rápidas.

Por otro lado, son muchos los paquetes no comerciales (con y sin licencia *Software Libre*) disponibles en el mercado, entre ellos destacan OpenCV, Gandalf, TargetJr, VXL (basado en TargetJr), CVIPTools, ImageLib, ImLib3D (Linux), LookingGlass, NeatVision (Java), TINA y XMegaWave (Unix/Linux). Todos ellos disponen de herramientas para el procesamiento de imágenes, pero a excepción de OpenCV y Gandalf ninguno proporciona un marco de trabajo completo para el desarrollo de aplicaciones relacionadas con la visión por computador. Esta capacidad de desarrollo contempla, no sólo el procesamiento de imágenes, sino tareas mucho más complejas como el reconocimiento de gestos, estimación del movimiento y la posición de un objeto, *morphing*, estimadores (filtros de Kalman, etc.), etc. Sin embargo, sólo OpenCV proporciona bibliotecas de tipos de datos estáticos y dinámicos (matrices, grafos, árboles, etc.), herramientas con la posibilidad de trabajar con la mayoría de las capturadoras/cámaras del mercado, entornos de desarrollo fáciles e intuitivos y todo ello, corriendo en dos de los sistemas operativos más utilizados del mundo Microsoft® Windows y Linux.

[1] Este trabajo ha sido financiado por el proyecto CICYT DPI-2002-01319.

El artículo está estructurado de la siguiente forma. En la sección 2 describimos las características de *The Open Computer Vision Library*. En la sección 3 detallamos algunas de las aplicaciones de dicha librería a la docencia e investigación. Finalmente, en la sección 4, presentamos las conclusiones de este trabajo.

2. The Open Computer Vision Library

El 13 de Junio del 2000, Intel® Corporation anunció que estaba trabajando con un grupo de reconocidos investigadores[2] en visión por computador para realizar una nueva librería de estructuras/funciones en lenguaje C. Esta librería proporcionaría un marco de trabajo de nivel medio-alto que ayudaría al personal docente e investigador a desarrollar nuevas formas de interactuar con los ordenadores. Este anuncio tuvo lugar en la apertura del IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). Había nacido *The Open Computer Vision Library* [5] y lo hacía bajo licencia BSD (*Software Libre*).

La librería OpenCV es una API de aproximadamente 300 funciones escritas en lenguaje C que se caracterizan por lo siguiente:

- Su uso es libre tanto para su uso comercial como no comercial (ver licencias en [2], [3] y [4] para más información).
- No utiliza librerías numéricas externas, aunque puede hacer uso de alguna de ellas, si están disponibles, en tiempo de ejecución.
- Es compatible con *The Intel® Processing Library* (IPL) y utiliza *The Intel® Integrated Performance Primitives* (IPP) para mejorar su rendimiento, si están disponibles en el sistema.
- Dispone de interfaces para algunos otros lenguajes y entornos: EiC - intérprete ANSI C escrito por Ed Breen. Hawk y CvEnv son entornos interactivos (escritos en MFC y TCL, respectivamente) que utilizan el intérprete EiC; Ch - intérprete ANSI C/C++ creado y soportado por la compañía SoftIntegration; Matlab® - gran entorno para el cálculo numérico y simbólico creado por Mathworks; y muchos más[3].

2.1 Estructura y características de la librería OpenCV

La librería OpenCV esta dirigida fundamentalmente a la visión por computador en tiempo real. Entre sus muchas áreas de aplicación destacarían: interacción hombre-máquina (HCI[4]); segmentación y reconocimiento de objetos; reconocimiento de gestos; seguimiento del movimiento; estructura del movimiento (SFM[5]); y robots móviles.

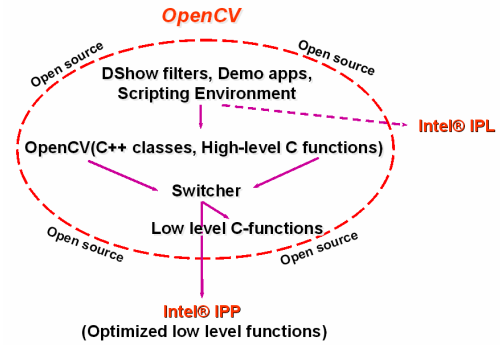

Figura 1: Estructura de la librería OpenCV.

[2] Intel facilitó el desarrollo y mantenimiento de la librería OpenCV aceptando y manteniendo las propuestas de código fuente revisadas por un grupo de expertos de la comunidad investigadora. Revisores que incluían representantes de los más importantes laboratorios de visión incluyendo: Jitendra Malik, University of California Berkeley; Takeo Kanade, Carnegie Mellon University; Pietro Perona, NSF Engineering Research Center, California Institute of Technology; Irfan Essa, Georgia Institute of Technology; Carlo Tomasi, Stanford University; Trevor Darrell, Massachusetts Institute of Technology; Stan Sclaroff, Boston University.

[3] Son muchos los proyectos *Software Libre* en avanzado estado de gestación que hacen uso de la librería OpenCV como marco de trabajo, entre ellos podemos destacar: Foresight; OpenCV User Contribution Library y S2iLib (visitar www.sourceforge.net para más información).

[4] Del término inglés *Human-Computer Interaction.*

[5] Del término inglés *Structure From Motion.*

Como podemos ver en la figura 1, la librería OpenCV proporciona numerosos elementos de alto nivel, como veremos en secciones posteriores, que facilitan sobre manera el trabajo al usuario (tanto al docente como al investigador). Por ejemplo, proporciona filtros Microsoft® DirectShow para realizar tareas tales como: calibración de la cámara, seguidores de objetos (*Kalman tracker* y *ConDensation tracker*), etc. Todos ellos se pueden utilizar en Microsoft® GraphEdit para ilustrar de forma bastante sencilla numerosas aplicaciones de visión. En la figura 2 y figura 3 podemos observar un ejemplo de ellos.

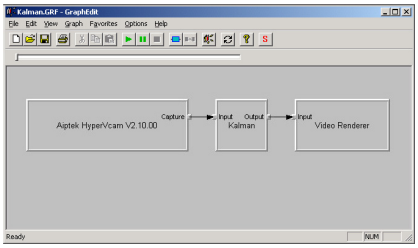

Figura 2: Seguidor de caras realizado con un filtro de *Kalman*.

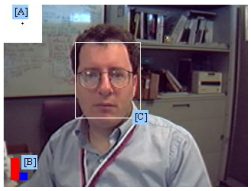

Figura 3: Salida del seguidor de caras realizado con un filtro de *Kalman*.

De igual forma, la librería OpenCV proporciona numerosas aplicaciones de ejemplo que ilustran como emplear las distintas funciones de la librería. En la figura 4 podemos ver una implementación de los HMM (*Hidden Markov Models*) para el reconocimiento de caras.

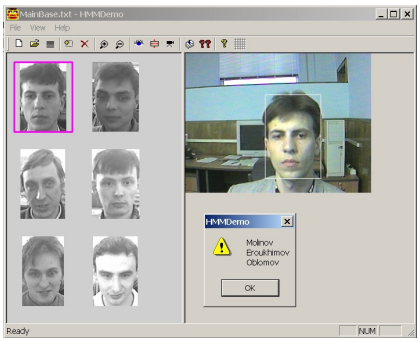

Figura 4: Detector de caras realizado con HMM (*Hidden Markov Models*).

También los entornos de *scripting* hacen uso de estas funciones para implementar su funcionalidad. La librería OpenCV proporciona una gran diversidad de entornos. En la sección 2.2 detallamos de forma pormenorizada algunos de los más utilizados.

Todas estas herramientas de alto nivel hacen uso de un paquete de clases C++ y funciones C de alto nivel que utilizan a su vez funciones muy eficientes escritas en C. Concretamente, el conjunto de funciones suministradas por la librería OpenCV se agrupan en los siguientes bloques:

- Estructuras[6] y operaciones básicas: matrices, grafos, árboles, etc.
- Procesamiento y análisis de imágenes: filtros, momentos, histogramas, etc.
- Análisis estructural: geometría, procesamiento del contorno, etc.
- Análisis del movimiento y seguimiento de objetos: plantillas de movimiento, seguidores (i.e. Lucas-Kanade), flujo óptico, etc.
- Reconocimiento de objetos: objetos propios (*eigen objects*), modelos HMM, etc.
- Calibración de la cámara: *morphing*, geometría epipolar, estimación de la pose (i.e. POSIT), etc.
- Reconstrucción tridimensional (funcionalidad experimental): detección de objetos, seguimiento de objetos tridimensionales, etc.
- Interfaces gráficos de usuarios y adquisición de video.

En la figura 1 también podemos observar como la librería OpenCV puede hacer uso de librerías propietarias como son en este caso *The Intel® Image Processing Library* (IPL) y *The Intel® Integrated Performance Primitives* (IPP) en caso de disponer de ellas. Estas aplicaciones tratan de mejorar el

[6] Todas estas estructuras disponen de funciones para su gestión: creación, inicialización, destrucción, conversión, estadísticas, etc. Por ejemplo, la librería OpenCV dispone de todo un conjunto de funciones para realizar operaciones con matrices, álgebra lineal (SVD, Mahalanobis, etc.) y funciones matemáticas de propósito general.

rendimiento de la librería con primitivas optimizadas para procesadores Intel®. Sin embargo, la licencia por las que se rigen estas aplicaciones no es *software libre* y por tanto carece de interés para los autores.

2.2 Interfaces gráficos y herramientas de la librería OpenCV

La librería OpenCV proporciona varios paquetes de alto nivel para el desarrollo de aplicaciones de visión. Todos ellos se pueden agrupar en librerías de C/C++ dirigidas a usuarios avanzados y en herramientas de *scripting* dirigidas, en este caso, a usuarios de nivel medio (ideal para practicar con las distintas técnicas de procesamiento de imágenes y visión). Al primer grupo pertenecen HighGUI y CvCam, mientras que al segundo pertenecen Hawk y OpenCV Toolbox para Matlab®.

HighGUI permite la escritura/lectura de imágenes en numerosos formatos (BMP, JPEG, TIFF, PxM, Sun Raster, etc.) y la captura de *stream* de video de capturadoras Matrox® y cámaras/capturadoras con drivers VFW/WDM (la mayoría del mercado); la creación de ventanas para visualizar imágenes en ellas, las ventanas HighGUI recuerdan su contenido (no es necesario implementar *callbacks* de repintado); y además, nos proporciona mecanismos muy fáciles de interaccionar con ellas: *trackbars*, capturando la entrada del teclado y el ratón.

CvCam nos proporciona un único interfaz de captura y reproducción bajo Linux y Win32; *callbacks* para la gestión de *stream* de vídeo o ficheros AVI y un mecanismo fácil para implementar visión estéreo con dos cámaras USB o una estéreo-cámara.

Hawk es un entorno visual con el intérprete ANSI C EiC como núcleo; soporta *plugins*; proporciona soporte para OpenCV, IPL y HighGUI vía *plugin*; y soporte de video.

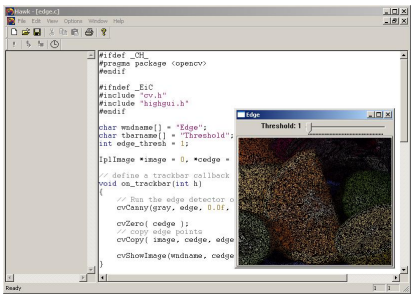

Figura 5: Hawk (EiC).

Por último, la librería OpenCV proporciona un *toolbox* para Matlab® que se caracteriza por lo siguiente:

- Utiliza tipos nativos de Matlab® (matrices, estructuras).
- Compatibilidad con la *Image Processing Toolbox*.

En la figura 6 podemos ver una implementación de un seguidor de caras denominado *CamShift* con el OpenCV Toolbox para Matlab®.

```
% Seguidor de caras Camshift, mejorado con filtro de ruido
function new_window = track_obj( img, obj_hist, window, thresh )
   probimg = cvcalcbackproject( img, obj_hist );
   % Eliminamos pequeños huecos a través de la operación morfológica 'close'
   probimg = cvclose( probimg, 3, 2 );
   probimg = cvthresh( probimg, thresh ); % Umbralizamos la imagen
   contours = cvfindcontours( probimg, 'external' ); % Localizamos los contornos
   mask_img = zeros(size(contours));
   for i = 1:length(contours)
      % Eliminamos los contornos pequeños;
      if  contous(i).rect(3)*contous(i).rect(4) < 30
         contours(i).pt = [];
      end
   end
   mask_img = cvfillcontours( mask_img, contours, 'w' ); % Obtenemos la mascara
   new_window = cvcamshift( mask_img, window ); % Aplicamos el seguidor CamShift a la mascara
return;
```

Figura 6: Seguidor de caras *CamShift* implementado con Matlab®.

3. Aplicación de la librería OpenCV en docencia e investigación

Como hemos detallado en secciones previas, la librería OpenCV nos proporciona el marco de trabajo ideal, tanto por el desarrollo de pequeñas aplicaciones prácticas, ideales como apoyo a la docencia en asignaturas relacionadas con la visión por computador, como para el desarrollo de la labor investigadora de cualquier grupo de visión.

En labores docentes es de gran interés disponer de herramientas que permitan al alumno experimentar las diferentes técnicas y procedimientos teóricos (metodológicos) explicados en clase.

Una de las herramientas más utilizadas a nivel académico es el paquete matemático Matlab®. Este paquete se utiliza en numerosas asignaturas como apoyo a la docencia y es ampliamente conocido por el mundo universitario. Matlab® dispone de un *toolbox* denominado *Image Processing Toolbox* que permite cargar imágenes en distintos formatos, visualizarlas y manejarlas como matrices numéricas. Sin embargo, carece de muchas de las características a las que hacíamos referencia en la sección 1, y que debería cumplir cualquier paquete de visión por computador que se precie. La librería OpenCV proporciona un *toolbox* que aprovecha muchos de los tipos de datos, estructuras y en general, la potencia que este paquete ofrece, para facilitar al usuario de nivel medio-bajo el acercamiento a muchas de las tareas habituales de la visión por computador. En la figura 6 podemos ver un ejemplo de código Matlab® de un seguidor de caras denominado *CamShift*, en la figura 7 podemos ver la salida de dicho algoritmo.

Figura 7: Salida del seguidor de caras *CamShift*.

Otro de los lenguajes más utilizados en el ámbito universitario es el lenguaje C. La librería OpenCV también proporciona, como ya hemos visto en secciones anteriores, numerosas librerías de apoyo que nos permiten implementar de forma sencilla y rápida aplicaciones de visión, obviando aspectos tales como las características de la cámara, el formato de las imágenes, etc. En la figura 8 podemos observar el código de un programa suministrado al alumno para que se familiarice con el entorno Hawk y con el operador de *Canny* de extracción de bordes[7]. En la figura 9 podemos observar la imagen de entrada y dos imágenes de bordes obtenidas con distintos parámetros de procesamiento.

```
void on_trackbar(int h) {
   cvCanny(gray, edge, 0.0f, (float)edge_thresh*5, 3); // Detectamos la imagen de bordes
   cvCopy( image, cedge, edge ); // Copiamos la imagen de bordes
   cvShowImage(wndname, cedge);
}
int main( int argc, char** argv ) {
   char wndname[] = "Edge", tbarname[] = "Threshold", filename[] = "fruits.jpg";
   if( (image = cvLoadImage( filename, 1)) == 0 ) return -1;
   // Creamos la imagen de salida
   cedge = cvCreateImage(cvSize(image->width,image->height), IPL_DEPTH_8U, 3);
   // Convertimos a escala de grises
   gray = cvCreateImage(cvSize(image->width,image->height), IPL_DEPTH_8U, 1);
   edge = cvCreateImage(cvSize(image->width,image->height), IPL_DEPTH_8U, 1);
   cvCvtColor(image, gray, CV_BGR2GRAY);
   cvNamedWindow(wndname, 1); // Creamos una ventana y un trackbar
   cvCreateTrackbar(tbarname, wndname, &edge_thresh, 100, on_trackbar);
   on_trackbar(0); // Mostramos la imagen
   cvWaitKey(0); // Esperamos a que el usuario pulse una tecla
   cvReleaseImage(&image); cvReleaseImage(&gray); cvReleaseImage(&edge);
   cvDestroyWindow(wndname);
   return 0;
}
```

Figura 8: Operador de extracción de bordes *Canny* escrito con C.

[7] La extracción de bordes constituye una de las etapas del procesamiento de imágenes más habituales e importantes. La librería OpenCV nos proporciona una función de extracción de bordes que nos permite trabajar en tiempo real con un rendimiento aceptable.

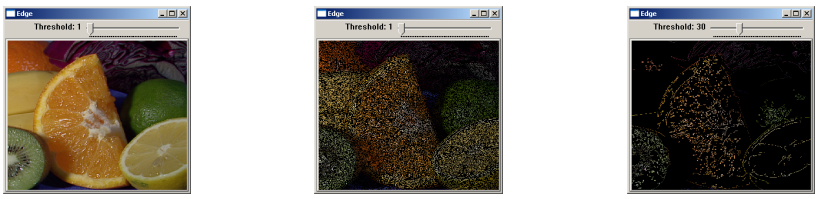

Imagen fuente Canny (t=1) Canny (t=30)

Figura 9: Salidas del operador de *Canny*.

Todo lo expuesto anteriormente es perfectamente extrapolable a cualquier investigador que desarrolle su labor en algún área de la visión por computador. Además, la disponibilidad del código de la librería permite añadir nuevas funcionalidades o simplemente modificar las existentes (respetando siempre los requisitos exigidos por la licencia BSD-*Software libre* [2] por la que se rige la misma) lo que proporciona una ventaja añadida. Así mismo, es importante señalar que existe una lista de distribución (OpenCV@yahoogroups.com) con cientos de usuarios en la que se discuten aplicaciones, *bugs*, etc. y un grupo de cualificados revisores encargados de examinar exhaustivamente todas las contribuciones de código antes de incorporarlas a la siguiente versión estable de la librería.

4. Conclusiones

Según lo expuesto a lo largo de este artículo, *The Open Computer Vision Library* constituye una de las apuestas más interesantes e importantes en visión por computador a lo largo de los últimos tiempos y además *Software Libre*. Su estabilidad, escalabilidad, su rápido crecimiento y desarrollo garantizan su continuidad y la hacen especialmente útil en el ámbito universitario, para la realización de tareas de docencia e investigación a corto, medio y largo plazo.

Referencias

[1] *Matrox Electronic Systems Ltd. Matrox Imaging Library (MIL).* www.matrox.com/imaging/products/mil/home.cfm, 2003.
[2] *Open Source Initiative. The BSD License.* www.opensource.org/licenses/bsd-license.php, 2003.
[3] *Open Source Initiative. The MIT License.* www.opensource.org/licenses/mit-license.php, 2003.
[4] *Open Source Initiative. The Open Source Definition.* www.opensource.org/docs/definition.php, 2003.
[5] *The Open Computer Vision Library.* www.intel.com/software/products/opensource/libraries/cvfl.htm, 2003.