

## EL ENTORNO: VISION - E LABORATORY

JAVIER MOLINA\* Y ALBERTO YÚFERA\*,\*\*

\* *Departamento de Tecnología Electrónica, Universidad de Sevilla, Av. Reina Mercedes s/n, Sevilla 41012, España. tf: 95 4556472, fax: 95 4552764.*

*email: {jfmolina,yuferaj}@dte.us.es*

\*\* *Instituto de Microelectrónica de Sevilla, Centro Nacional de Microelectrónica (IMSE-CNM)*

*Av. Américo Vespucio, sn, Sevilla, 41092*

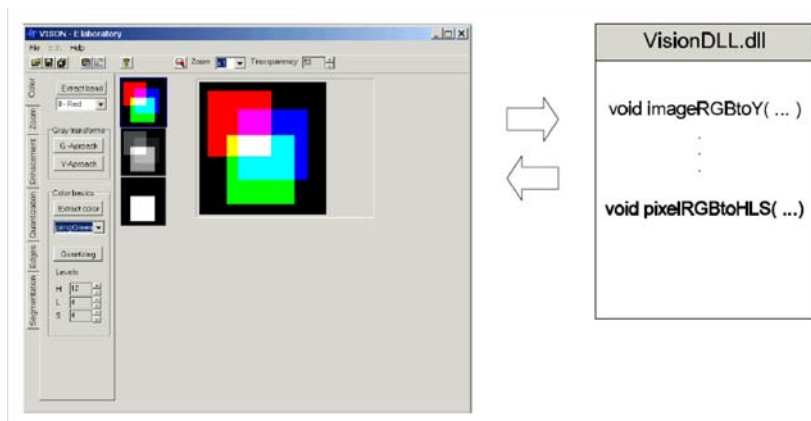
*España. tf: 954 466666*

*Se presenta un entorno de programación para el estudio y desarrollo de algoritmos de tratamiento de imágenes y visión por computador. La herramienta permite la programación en cualquier lenguaje, con la única restricción de que se desarrolle bajo entornos windows. Se ha concebido como instrumento de ayuda a la realización de prácticas de laboratorio de alumnos en las asignaturas de tratamiento digital de imágenes de ingeniería informática.*

*Palabras clave: Software educativo, tratamiento de imágenes digitales.*

### 1. Introducción

El procesado de imágenes digitales se encuentra presente en gran variedad planes de estudio y programas docentes de titulaciones pertenecientes a áreas científico-técnicas. A menudo, la formación de un ingeniero (telecomunicaciones, industrial, informático, electrónico) incluye tópicos sobre análisis y procesamiento de imagen, como instrumento para la resolución de diversos problemas prácticos (detección, diagnóstico, regulación, reconocimiento, etc). El aprendizaje y entrenamiento de tales destrezas se obtiene, en muchas ocasiones, a través de herramientas de computación comerciales (p.e. MATLAB) que han sido concebidas para un campo de aplicaciones más amplio, y que generalmente utilizan una sintaxis estricta y específica. El propósito de desarrollar una herramienta docente para el análisis y procesamiento de imágenes digitales, El Entorno Vision-E, ha sido facilitar la programación de algoritmos para el tratamiento de imágenes en diferentes lenguajes, bajo el SO Windows, haciendo el interfaz de programación lo más transparente y sencillo posible al usuario. El objetivo ha sido doble. Primero, tratar de que el entorno sea lo más amigable posible para el alumno, de manera que el tiempo de aprendizaje de su uso sea reducido al mínimo. Segundo, cubrir las necesidades de recursos de computación que surgen como consecuencia del desarrollo de los contenidos de las asignaturas de procesamiento digital de imagen. En la Figura 1 se muestra el panel frontal del E-visión lab. Los componentes fundamentales de Vision-E son un programa ejecutable (Vision.exe) y de una librería de enlace dinámico VisionDLL.dll.



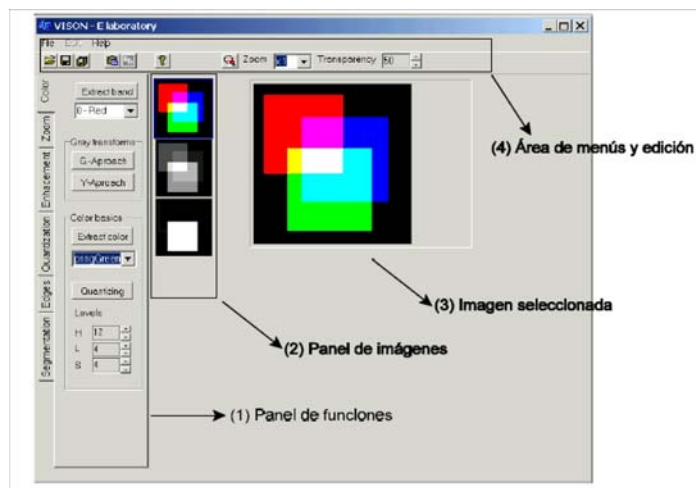
**Figura 1:** Estructura general de Vision-E lab.

La misión del programa *Vision.exe* es hacer de interfaz con el usuario, por un lado, y con los recursos del computador, por otro. Permite leer, visualizar y crear ficheros de imágenes en formato bitmap (BMP), y gestiona la memoria necesaria para manipular las matrices de las imágenes. De otra parte, *VisionDLL.dll* es un fichero que generará el programador. Debe contener las funciones que desarrollan los algoritmos de tratamiento de las imágenes seleccionadas en el entorno.

El programador deberá crear un proyecto de creación de una DLL empleando el compilador y el lenguaje que desee. Se recomienda emplear Visual C++ 6.0 o superior, para el cual ya hemos incluido un proyecto tipo. Todos los ficheros necesarios pueden descargarse de la Web de la asignatura: [www.dte.us.es/ing\\_inf/trat\\_voz/index.html](http://www.dte.us.es/ing_inf/trat_voz/index.html).

## 2. El entorno visión-E

El entorno de trabajo *Vision.exe* ofrece una interfaz gráfica para simplificar la adquisición, visualización y manipulación de imágenes, y una librería de funciones para facilitar el trabajo del programador. El entorno se divide en 4 zonas de trabajo como se indica en la figura 2.



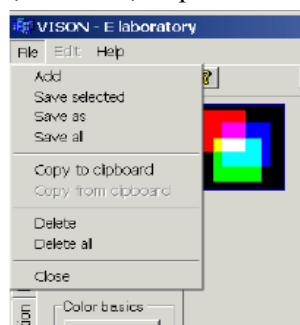
**Figura 2:** Descripción del Interfaz de Vision-E lab

### 2.1 Panel de imágenes

Contiene versiones en miniatura de las imágenes añadidas desde fichero y las resultantes de las operaciones de tratamiento. Al pulsar el ratón sobre alguna de ellas, automáticamente se selecciona y aparece en la ventana correspondiente. Todas las operaciones de tratamiento se ejecutan por defecto sobre la imagen seleccionada.

### 2.2 Área de menús y edición

Mediante menú o bien con los correspondientes botones pueden efectuarse las siguientes funciones: Añadir, Guardar, Eliminar, Copiar.

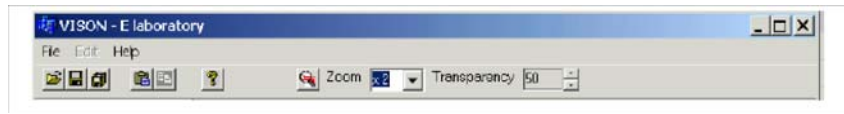


- **Añadir** una imagen de disco a las ya existentes
- **Guardar** una o todas las imágenes
- **Eliminar** la imagen seleccionada o todas
- **Copiar** la seleccionada sobre el portapapeles.

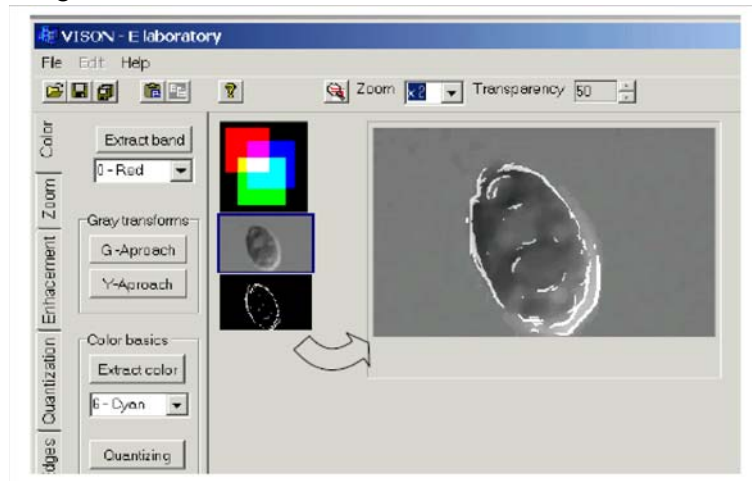
**Figura 3:** Descripción del Interfaz de Vision-E lab

2.3 Las funciones de edición son dos:

- **Zoom** sobre la imagen seleccionada.



• **Transparencia.** Esta función permite superponer sobre la ventana de selección, una segunda imagen con el grado de transparencia que se desee. Basta para ello arrastrar desde el panel de imágenes hacia la ventana de selección, la imagen que deseamos superponer. Para realizar esta operación es necesario que ambas imágenes tengan el mismo tamaño.



2.3 Panel de funciones:

Las funciones de tratamiento que se desarrollan en cada una de las prácticas se han clasificado en diferentes paneles tabulados con las siguientes etiquetas.



- **Color.** Contiene llamadas a diferentes algoritmos de manipulación de imágenes a color como:
  - Extraer banda
  - Transformaciones a gris
  - Extraer color
  - Cuantizar el color de una imagen
- **Zoom.** En este panel se realizan llamadas a operaciones básicas de zoom como:
  - Zoom de orden cero
  - Zoom de primer orden
  - Zoom de interpolación bilineal
- **Realce**
- **Cuantización**
- **Detección y seguimiento de flancos**
- **Segmentación de imágenes**

En cada panel se incluyen botones que lanzarán la llamada a las funciones que el programador situará en la librería de enlace dinámico. También aparecen elementos para la selección de parámetros necesarios para efectuar el algoritmo. P.e el color a extraer, el número de niveles de saturación, etc.

### 3. Librería de funciones de Vision-E

Vision-E contiene un conjunto de funciones en *runtime* o en código fuente, que pueden emplearse desde el programa de usuario, y que facilitan la creación, y manipulación de imágenes dentro del entorno de trabajo.

#### • Funciones para la manipulación de matrices en memoria:

- `newImage` Crea una matriz tipo byte
- `deleteImage` Libera la memoria de la matriz
- `newFloatImage` Crea una matriz tipo float
- `deleteFloatImage` Libera la memoria anterior

#### • Funciones que actúan sobre la interfaz gráfica

- `DisplayFile` Muestra un fichero de imagen
- `DisplayGray` Muestra una matriz de imagen
- `DisplayRGB` Muestra una imagen RGB
- `SaveGrayImage` Guarda una matriz como fichero
- `SaveGrayImageAs`
- `SaveRGBImage` Guarda una imagen RGB a disco
- `SaveRGBImageAs`
- `ImageSelect` Selecciona una imagen del panel
- `DeleteSelect` Elimina la imagen seleccionada
- `DeleteAll` Elimina todas las imagenes

#### • Funciones auxiliares

- `DisplayHistogram` Muestra un histograma
- `DisplayImageHistogram` Calcula y muestra el histograma de una imagen, y lo guarda en disco

#### • Funciones para la creación y eliminación de matrices

```
unsigned char ** newImage( int Rows, int Cols)
float ** newFloatImage( int Rows, int Cols)
```

Descripción: Crean una matriz de las dimensiones indicadas y elementos tipo byte y float respectivamente

```
void deleteImage(unsigned char **matrix, int Rows)
void deleteFloatImage(unsigned char **matrix, int Rows)
```

Descripción: Liberan la memoria reservada de la matriz

### 3.1 Funciones sobre la interfaz gráfica (Runtime de Vision-E)

```
void DisplayFile(char *filename)
void DisplayGray(unsigned char **matrix, int Rows, int Cols,
char *filename)
void DisplayRGB(unsigned char **R, unsigned char **G,
unsigned char **B,
int Rows, int Cols, , char *filename)
```

Descripción: Añaden una imagen al final del panel de imágenes, seleccionándola automáticamente. El origen de la imagen puede ser disco, una imagen monocromática (una matriz), o una imagen RGB (tres matrices)

```
void SaveGrayImage(unsigned char **matrix,int Rows,int Cols, char *filename)
void SaveGrayImageAs(unsigned char **matrix, int Rows, int Cols)
void SaveRGBImage( unsigned char **R, unsigned char **G, unsigned char **B,
int Rows, int Cols, char *filename)
void SaveRGBImageAs(unsigned char **R, unsigned char **G, unsigned char** B,
int Rows, int Cols)
```

**Descripción** Guardan o copian una imagen sobre el disco con el nombre indicado.

```
int ImageSelect( int number)
void DeleteSelected(void)
void DeleteAll()
```

Respectivamente:

1. Selecciona una imagen de la lista (devuelve 1 si cierto)
2. Elimina la seleccionada

3. Elimina todas la imágenes

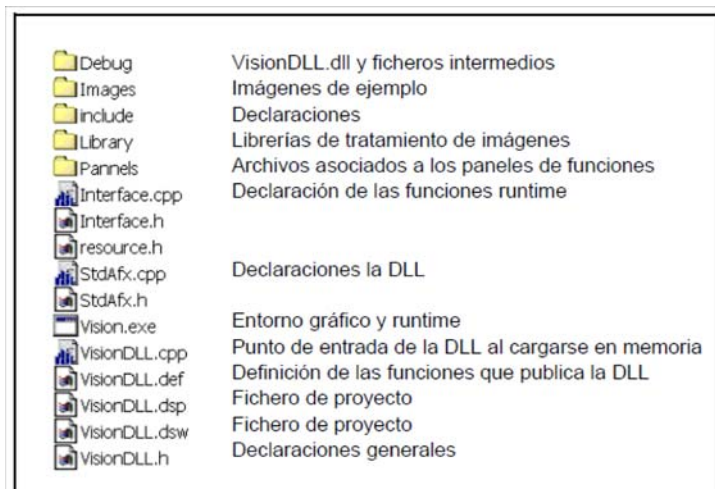
### 3.3 Funciones auxiliares (Runtime de Vision-E)

```
void DisplayHistogram( float *histo, char *filename)
void DisplayImageHistogram( unsigned char **matrix,
int rows, int cols, char *filename)
```

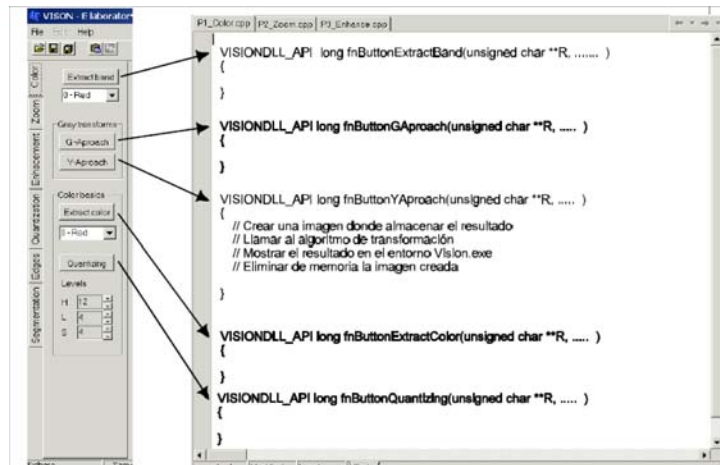
Descripción: Muestran el histograma como una imagen añadida a la lista del panel de imágenes. El origen del mismo puede ser un vector con los datos del histograma o una imagen en memoria.

## 4. Estructura del proyecto VisionDLL.dll con Visual C++ 6.0

El fichero VisionE.zip contiene el entorno y todos los ficheros necesarios para generar la DLL con Visual C++ 6.0 ó superior. Los programas de tratamiento residen en dicha DLL.



El proyecto de DLL se encuentra configurado en todas sus opciones en **VisionDLL.dsw**. Basta abrirlo con Visual C++ para comenzar a programarlo. Todas las prácticas están incluidas en él. En cada sesión deberá programarse la parte que requiera el problema. Para facilitar la localización de las funciones, en la carpeta Pannels se incluyen 3 ficheros: P1\_Color, P2\_Color y P3\_Color que corresponden con las funciones asociadas a la interfaz gráfica de los paneles 1, 2 y 3 respectivamente. P.e: VisionDLL.dll y ficheros intermedios



En general, dentro de las funciones asociadas a los botones, debe programarse el código necesario para:

- Crear, copiar o destruir matrices con imágenes que se usan de modo temporal.
- Mostrar en el entorno imágenes, resultados, histogramas, etc.
- En general, para emular cualquier acción que pueda hacerse con el ratón (abrir ficheros, seleccionar de la lista, eliminar, etc...).

- Realizar llamadas a funciones de tratamiento de imágenes. Estas funciones deberán programarse en ficheros de la librería específicos para cada problema.

## 5. Metodología de trabajo

Partiendo de estas condiciones de trabajo, el alumno (usuario) debe proceder a realizar la programación de las funciones, algoritmos necesarios par el problema de procesado que debe resolver. Para ello, deberá:

1. Analizar si la acción requiere una función de procesado o se resuelve simplemente con operaciones de manipulación de matrices.
2. En caso de necesitar algún algoritmo específico de tratamiento, abrir el fichero que corresponda a la librería que necesite. Como guía se le ofrece una definición de las funciones que habitualmente va a necesitar. Sin embargo, es libre de cambiarla o definir otras diferentes. Los ficheros de la librería son los siguientes:
  - Color.cpp Funciones propuestas para manipulación del color
  - Zoom.cpp Funciones para algoritmos de zoom e interpolación
  - Histogram.cpp Funciones relacionadas con el cálculo de histogramas
  - Enhance.cpp Funciones para el realce de imágenes
3. Programar la función asociada a la acción del ratón en el fichero del panel que corresponda. El diseño de estos paneles se ha realizado conforme a las especificaciones de cada práctica, de forma que el alumno solo tenga que programar el cuerpo de las funciones. Se han diseñado una serie de templates genéricos para dejar abierta la programación de funciones previamente no especificadas.

La experiencia obtenida después de varios cursos utilizando esta herramienta en las prácticas de la asignatura de Tratamiento de Voz e Imágen, de 5º curso de la titulación de Ingeniería Informática, es muy positiva. Por un lado, los tiempos de realización de la prácticas (dos horas por sesión) son suficientes en la mayoría de los casos. Los alumnos aprenden a utilizar la herramienta en cinco minutos, y a partir de ahí se dedican exclusivamente a programar los algoritmos de procesado de imagen. Por otro, dada la sencillez de uso de la herramienta, los alumnos tienen la oportunidad de experimentar y programar modificaciones sobre los algoritmos propuestos, motivando su participación.

## 6. Conclusiones

En este trabajo se ha descrito un entorno de programación que pretende facilitar el acceso rápido de los alumnos y de interesados en el tratamiento/procesado digital de imágenes a la programación de algoritmos. En los casos de prácticas dirigidas, incluye el diseño de los paneles de gestión y ejecución de funciones. Se han desarrollado gran cantidad de funciones de tratamiento de funciones básicas que ayudan al alumno/usuario a centrarse exclusivamente en la programación de los algoritmos de procesado de interés, reduciendo errores de programación y potenciando el uso de algoritmos avanzados. Desde hace varios años, la herramienta se utiliza con éxito en la asignatura de tratamiento de voz e imagen, quinto curso de ingeniero en informática.

## Agradecimientos

Este trabajo ha sido en parte financiado por el proyecto: *Auto-Calibración y Auto-Test de Circuitos Analógicos, Mixtos y de Radio Frecuencia*. P09-TIC5386. Proyecto de Excelencia de la Junta de Andalucía.

## Referencias

- [1] Web de la asignatura Tratamiento de Voz e Imagen, 5º Ingeniero en Informática, Universidad de Sevilla. [https://www.dte.us.es/ing\\_inf/trat\\_voz/](https://www.dte.us.es/ing_inf/trat_voz/)
- [2] Manual para Borland C++ Builder 6.0.
- [3] R. C. González; R. E. Woods; *Digital Image Processing* Prentice Hall; 2002.