

Adquisición y filtrado de video digital en tiempo real usando GUIs de Matlab®

Ing. Juan M. Vilardy, Ing. Jorge E. Calderon, Ph.D. Cesar O. Torres, M.Sc. Lorenzo. Mattos.

Laboratorio de Óptica e Informática, Universidad Popular Del Cesar, Sede Balneario Hurtado, Valledupar, Colombia.
Teléfono: (+57) 5-5734089, torres.cesar@caramail.com, vilardy.juan@unicesar.edu.co

Resumen

Se desarrolló una interfaz grafica de usuario interactiva para capturar, almacenar y filtrar video digital en tiempo real, bajo la plataforma Matlab. El sistema desarrollado soporta diferentes tipos de dispositivos de captura de video, como: Matrox Inc, Coreco Imaging Inc, Data Translation Inc, DCAM (IEEE 1394, FireWire), Windows Driver Model (WDM) or Video for Windows (VFW) drivers (USB e IEEE 1394 Web Cameras, Digital Video Camcorders y TV tuner card), los frames se tratan matricialmente para su visualización y filtrado en tiempo real, mediante la convolución espacial con los diferentes filtros implementados. Esta solución para adquirir video puede ser utilizada para tratar el video digital de una manera eficiente, debido a que los frames se encuentran en forma matricial y así posibilita la aplicación de muchas técnicas de tratamiento de imágenes y video digital.

Palabras clave: Adquisición de video, Convolución Espacial, Kernel, GUI, MATLAB.

1. Introducción

Para el tratamiento de la señal de video digital en determinadas aplicaciones se hace necesario efectuar operaciones lo mas rápido posible, para tomar decisiones al instante analizando las secuencias de imágenes tratadas, en estas aplicaciones se hace necesario adquirir la señal de video digital y llevarla al entorno donde se va a tratar.

En el Laboratorio de Óptica e Informática se desarrolló un sistema de adquisición y filtrado de video digital en tiempo real, que utiliza cámaras comerciales, la plataforma de programación técnica Matlab® e interfaz grafica de usuario (GUI, Graphical User Interface) de Matlab®, cabe aclarar que en este trabajo, tiempo real esta definido como el tiempo necesario para adquirir, procesar y visualizar la señal de video digital de acuerdo a la persistencia del ojo humano para observar imágenes en movimiento.

2. Filtrado espacial

El filtrado espacial es la operación que se aplica a los cuadros de video raster (mapa de bits) para mejorar o suprimir detalles espaciales con el fin de mejorar la interpretación visual. Ejemplos comunes incluyen aplicar filtros para mejorar los detalles de bordes en los frames, o para reducir o eliminar patrones de ruido. El filtrado espacial es una operación “local” en el sentido que

modifica el valor de cada píxel de acuerdo con los valores de los píxeles cercanos y el kernel de convolución usado (por lo general este kernel es una mascara matricial de tamaño impar cuadrada), utilizando la operación de Convolución Discreta, así:

$$I_{m,n} = \sum_{i=-r}^r \sum_{j=-r}^r H_{i,j} G_{m-i,n-j} = \sum_{i=-r}^r \sum_{j=-r}^r H_{-i,-j} G_{m+i,n+j} \quad (1)$$

De la ecuación anterior tenemos: $G_{m,n}$ como el frame a filtrar de m filas y n columnas, $H_{i,j}$ es kernel de la convolución o la mascara, el cual es asumido de tamaño impar (esto con el fin de ubicar el píxel a cambiar en el centro de la mascara, y realizar el proceso de filtrado como se menciono anteriormente) con $(2r+1)(2r+1)$ coeficientes, es decir $2r+1$ filas y $2r+1$ columnas e $I_{m,n}$ es el resultado de la operación de filtrado.

Los filtros empleados en este artículo dan como resultado el suavizado y la extracción de bordes de los cuadros de video por medio de la operación efectuada en la ecuación (1). A continuación se muestran las diferentes características de los filtros usados y su correspondiente kernel.

2.1. Filtro de Promedio

Es un filtro que se aplica directamente en el dominio espacial, obteniendo un cuadro de video suavizado, cuya intensidad para cada píxel se obtiene promediando los valores de intensidad de los píxeles incluidos en el entorno de vecindad predefinido por el kernel para cada píxel. Este filtro además de suavizar el frame, suprime ruidos que pueden aparecer como causa del muestreo, cuantización y transmisión, o bien por perturbaciones en el sistema tales como partículas de polvo en el sistema óptico, obteniendo homogeneidad en el frame filtrado. El kernel de este filtro es:

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2)$$

2.2. Filtro Sobel en ambas direcciones

Este tipo de filtro enfatiza los bordes que rodean a un objeto en un frame en todas las direcciones (tanto horizontal como vertical) utilizando el operador gradiente. Este filtro también es conocido como el

operador Sobel y además de enfatizar los bordes este tiene la propiedad añadida de suavizar el frame filtrado, eliminando parte del ruido y por consiguiente minimiza la aparición de falsos bordes debido al efecto de magnificación del ruido por parte de los operadores derivada. Para el operador Sobel en ambas direcciones el kernel esta dado por [1]:

$$H = \frac{1}{8} \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad (3)$$

2.3. Filtro Laplaciano de Gaussianas

El filtro Laplaciano de Gaussianas es un operador de segunda derivada, isótropo capaz de detectar bordes en todas las orientaciones posibles, reduciendo el efecto del ruido al suavizar el frame por el efecto Gaussiano. Este operador de segunda derivada como su nombre lo indica esta constituido por una función Gaussiana bidimensional a la cual se le aplica el operador Laplaciano, su mascara o kernel esta definido por [2]:

$$H = \begin{bmatrix} 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0564 & 0.7146 & -4.9048 & 0.7146 & 0.0564 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \end{bmatrix} \quad (4)$$

2.4. Filtro Sharpen

A diferencia de los dos filtros anteriores de extracción de bordes (Sobel en ambas direcciones y Laplaciano de Gaussianas) este filtro resalta los bordes no creando un fondo oscuro y sin suprimir los colores. Esta clase de filtro es una variación de los filtros paso altos discretos (los cuales mejoran los detalles finos de una imagen), ya que estos logran dar un mejor contraste y nitidez visual al frame procesado. El kernel del filtro Sharpen esta representado por [3]:

$$H = \begin{bmatrix} -0.1667 & -0.6667 & -0.1667 \\ -0.6667 & 4.3333 & -0.6667 \\ -0.1667 & -0.6667 & -0.1667 \end{bmatrix} \quad (5)$$

3. Desarrollo del sistema de adquisición y filtrado de video digital en tiempo real

La interfaz grafica de usuario implementada se constituye de tres ventanas, las cuales son: Ventana de presentación de la aplicación, la cual contiene tres botones llamados Adquisición, Filtrar y Salir, los dos primeros botones conducen a las otras dos ventanas restantes, es decir si el usuario solo quiere adquirir y no filtrar debe presionar el botón Adquisición y se despliega la ventana respectiva a la adquisición de video, si se desea adquirir y filtrar a la vez se debe presionar el botón Filtrar para que se abra la ventana que realiza las operaciones mencionadas. Para realizar la aplicación de adquisición y/o filtrado de video digital, se utilizó

Matlab® versión 7.1, con el Image Acquisition Toolbox (IMAQ), GUIDE (Graphical User Interface Development Environment) y el Image Processing Toolbox, una tarjeta receptora de T.V (T.V Tuner Card) para importar el video y llevarlo a Matlab® (al WorkSpace), cámaras análogas (Handycam Vision CCD-TRV108 NTSC, SONY) o USB, y un PC IBM con un procesador de 2.26 GHz y 256 MBytes de memoria RAM. En la figura 1 se muestra la GUI de presentación de la aplicación.



Figura 1. GUI de presentación de la aplicación.

3.1. Adquisición

En esta GUI se tienen los siguientes objetos:

TABLA I
TIPO DE DATOS Y OBJETOS DE LA GUI DE ADQUISICIÓN

Descripción	Tipo de Datos	Objeto de la GUI
Adaptador	Carácter	Static text
Nombre de Adaptadores	Celda de Caracteres	Popup Menu
Dispositivo	Carácter	Static text
Nombre de Dispositivos	Celda de Caracteres	Popup Menu
Reset	-----	Push button
Preview	-----	Push button
Adquisición	-----	Push button
Stop	-----	Push button
Guardar	-----	Push button
Salir	-----	Push button

La función de los Popup Menu es proveer información al usuario de los adaptadores y dispositivos conectados al sistema (computador), el adaptador es el software que el toolbox (Image Acquisition) usa para comunicarse con el dispositivo que adquiere el video por medio del driver de dicho dispositivo, entonces dependiendo del número de adaptadores que soporte el sistema, el usuario podrá establecer comunicación con un dispositivo soportado por algún adaptador. Al iniciar esta GUI se aplica un reset al IMAQ, se pide la información de los adaptadores existentes y soportados por Matlab®, con esta información se actualiza el popup menu que contiene el nombre de los adaptadores. Dependiendo del adaptador seleccionado y los dispositivos de captura de video

(cámaras) conectados, se actualiza el otro popup menu que contiene el nombre de las cámaras. Los push button ejecutan acciones relacionadas a su nombre:

1. Preview: Muestra el video capturado por la cámara seleccionada.
2. Adquisición: Crea un objeto de video en Matlab®, lo inicializa (start), adquiere los frames del video capturado por el dispositivo seleccionado en la forma especificada por los parámetros de adquisición del objeto de video [4], y por ultimo estos frames son escritos en un archivo del disco duro en formato AVI.
3. Stop: Botón de parada para detener la prevista (Preview) o la adquisición de video (Adquisición).
4. Guardar: Aquí se determina si se almacena o no el video que fue adquirido y llevado a un archivo del disco duro.
5. Reset: Devuelve al IMAQ a un estado conocido, borrando todos los objetos de video.
6. Cerrar: Determina si el usuario desea salir de la aplicación o no.

La GUI de Adquisición de video se muestra en la figura 2.

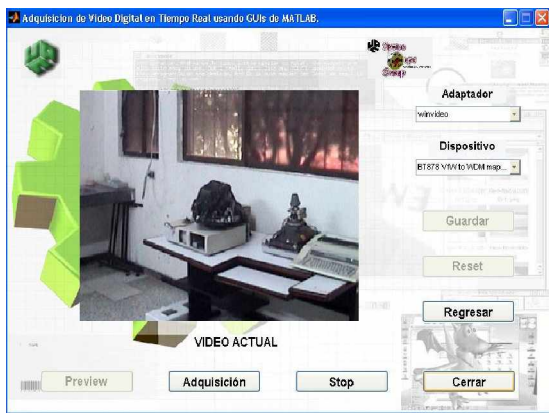


Figura 2. GUI de Adquisición de Video.

3.2. Adquisición y Filtrado

Para esta GUI se tiene los mismos objetos que la GUI anterior, pero aquí el botón de adquisición es ahora el botón de Filtrar y se adicionaron tres Radix button llamados: Sharpen, Sobel y Laplacian, los cuales son los nombres de los filtros a aplicar y están contenidos en un iobutton group [5], estos radix button son excluyentes, lo cual significa que si uno está seleccionado los demás no lo están; Dependiendo del radix button seleccionado, cuando se presione el botón Filtrar, se visualizará el efecto del filtro seleccionado al instante por medio de la programación de eventos realizada de la siguiente manera: Cuando el objeto de video es inicializado (start), este es inmediatamente disparado (trigger inmediato), por consiguiente las imágenes capturadas por la cámara seleccionada son almacenadas en un buffer de la memoria del PC, luego de inicializar y dar el trigger al objeto de video cuando se cumple el tiempo especificado por el TimerPeriod (este es el tiempo en segundos que especifica el tiempo entre eventos, siempre y cuando el

objeto de video este inicializado), se ejecuta la función especificada por el TimerFcn (esta es una propiedad del objeto de video [4]), en la función que se llama para ser ejecutada, los procedimientos realizados son:

1. Adquisición de un frame del buffer de la memoria del PC, para ser llevado al espacio de trabajo (WorkSpace) de Matlab®.
2. Borrado del frame adquirido del buffer de la memoria del PC.
3. Visualización del frame adquirido en la GUI.
4. Filtrado en el dominio espacial del frame adquirido, utilizando el kernel adecuado para el filtro seleccionado.
5. Visualización del frame filtrado en la misma GUI, donde fue visualizado el frame adquirido.
6. Se guardan los frames adquiridos y filtrados en dos archivos de video (en formato AVI) respectivamente.

Una vez ejecutados estos pasos, el toolbox IMAQ mide el tiempo establecido por el TimerPeriod y una vez cumplido este, se repiten los procedimientos anteriores y así sucesivamente. El TimerPeriod establecido en este trabajo fue de 1/25 segundos, es decir que la aplicación implementada puede adquirir, visualizar y filtrar 25 frames por segundo (en la visualización se muestran tanto el frame adquirido como el frame filtrado). El equipo (computador) utilizado para realizar la adquisición, filtrado y visualización como ya fue mencionado, tiene un procesador de 2.26 GHz y una memoria RAM de 256 MBytes, ocupando el 100% del procesador y el 50% de la memoria RAM aproximadamente cuando la GUI de Adquisición y Filtrado se encuentra adquiriendo, procesando y visualizando el video digital, por lo cual si se desea guardar el video adquirido y el filtrado adicionalmente se necesita un procesador más rápido en comparación al utilizado en este trabajo de investigación (se necesita mínimo un procesador de 3.0 GHz), debido a que la operación de escribir el frame en un archivo del disco duro exige un buen porcentaje de uso del procesador, adicionando más tiempo de ejecución a la programación del evento realizada. La GUI de Adquisición y Filtrado de video se encuentra en la figura 3.

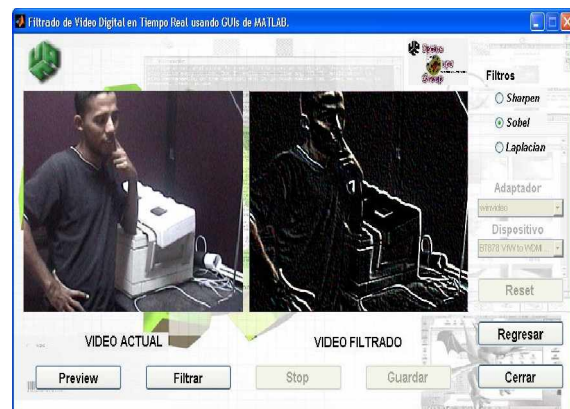


Figura 3. GUI de Adquisición y Filtrado de Video.

4. Conclusiones

En este trabajo se presentó la adquisición y filtrado de video digital en tiempo real por medio de la implementación de un sistema de video digital controlado por un programa codificado en la plataforma de programación técnica Matlab®, con ayuda del toolbox de adquisición de imágenes, y el ambiente de desarrollo de GUIs, mostrando como los frames son tratados matricialmente para su filtrado en el dominio espacial y visualización, siendo así atractivo adquirir video por medio de esta herramienta, ya que se puede modificar cualquier frame en la forma que se desee. Como lo que se esta tratando es video, la velocidad con la cual el toolbox IMAQ puede procesar los diferentes frames depende de la velocidad del procesador, la complejidad del algoritmo de procesamiento y la velocidad con que el dispositivo de captura adquiere los frames; por lo tanto dado un procesador bastante rápido, un simple algoritmo, y una velocidad de adquisición de frames ajustada a la velocidad de adquisición de frames del dispositivo de captura, el toolbox IMAQ puede procesar los frames muy rápidamente.

Referencias

- [1] B. Jahne, “Digital Image Processing”. Ed. Springer-Verlag. Alemania, 2003.
- [2] G. Pajares, J. De la Cruz, “Visión por Computadora”, Ed. Alfaomega, pp 65 – 179. Colombia, noviembre, 2004.
- [3] The MathWorks, *MATLAB Image Processing Toolbox help*.
http://www.mathworks.com/access/helpdesk/help/pdf_doc/imaq/imaq_print.pdf. September 2005.
- [4] The MathWorks, *MATLAB Image Acquisition Toolbox help*.
http://www.mathworks.com/access/helpdesk/help/pdf_doc/imaq/imaq_print.pdf. September 2005.
- [5] The MathWorks, *MATLAB Creating Graphical User Interfaces help*.
http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/buildgui.pdf. September 2005.