Universidad Industrial de Santander



Facultad de Ingenierías Físico-Mecánicas Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones





Entrenamiento Día MATLAB® 2008: Taller de adquisición y procesamiento de imágenes y video mediante MATLAB®

MSc Leandro F. Ariza Jiménez Grupo en Conectividad y Procesado de Señal (CPS)

> Universidad Industrial de Santander



Agenda

- Introducción
- Objetivo propuesto
- Tópicos propuestos
- Desarrollo metodológico
 - Manipulación matricial en MATLAB®
 - Acondicionamiento y ejecución del proceso de adquisición de video e imágenes
 - Procesamiento de imágenes y video utilizando MATLAB®



Introducción

- Matrices y vectores
- MATLAB® >> Matrix Laboratory
- Imágenes >> Arreglo matricial (matrix array)
- Videos >> Secuencia matricial (frames)
- Toolboxes:
 - Image Acquisition Toolbox
 - Image Processing Toolbox



Objetivo propuesto

Acercar a los participantes a través de una sesión de cuatro horas a las capacidades de adquisición y procesamiento de imágenes y videos soportadas por los correspondientes toolboxes de MATLAB®.



Viernes, I de agosto de 2008



Tópicos propuestos

- Fundamentos de manipulación matricial en MATLAB®.
- Control del equipo de registro mediante MATLAB®.
- Adquisición de video e imágenes mediante MATLAB®.
- Técnicas básicas de procesamiento de imágenes y video utilizando MATLAB®.



Desarrollo metodológico

FASE DE TRABAJO	DURACIÓN		
Manipulación matricial en MATLAB®	60 minutos		
Acondicionamiento y ejecución del proceso de adquisición de video e imágenes	90 minutos		
Procesamiento de imágenes y video utilizando MATLAB®	90 minutos		







Vector fila

$$I(1,N) = \begin{bmatrix} I(1,1) & I(1,2) & I(1,3) & \dots & I(1,N) \end{bmatrix}_{1\times N}$$

$$I = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

$$I = \begin{bmatrix} 1, & 2, & 3, & 4, & 5 \end{bmatrix}$$

$$I = \begin{bmatrix} -2, & -1, & 0, & 1, & 2 \end{bmatrix}$$

CONSTRUIMOS FUTURO

Vector columna

$$I(M,1) = \begin{bmatrix} I(1,1) \\ I(2,1) \\ I(3,1) \\ \vdots \\ I(M,1) \end{bmatrix}_{Mx1}$$

$$I = [1; 2; 3; 4; 5]$$

$$I = [-2; -1; 0; 1; 2]$$

Universidad Industrial de Santander

Matriz

$$I(M,N) = \begin{bmatrix} I(1,1) & I(1,2) & I(1,3) & \dots & I(1,N) \\ I(2,1) & I(2,2) & I(2,3) & \dots & I(2,N) \\ I(3,1) & I(3,2) & I(3,3) & \dots & I(3,N) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I(M,1) & I(M,2) & I(M,3) & \dots & I(M,N) \end{bmatrix}_{MxN}$$

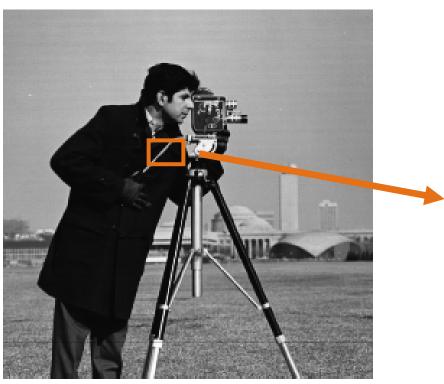
$$I = [1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 7 \ 8 \ 9]$$

$$I = [1, 2, 3; \ 4, 5, 6; \ 7, 8, 9]$$

Viernes, I de agosto de 2008



• Matriz: Escala de grises (grayscale)



8	9	11	16	10	13	26	142	173	65
10	11	9	17	13	17	118	178	73	13
13	11	11	11	16	106	178	68	16	13
12	11	12	11	69	181	62	15	13	16
12	11	12	82	168	60	14	13	15	16
9	10	69	182	67	14	12	14	15	15

256x256

Viernes, I de agosto de 2008



Matriz multidimensional

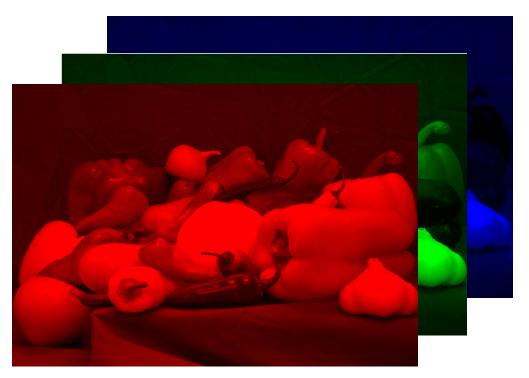
Viernes, I de agosto de 2008



Matriz multidimensional: Color verdadero







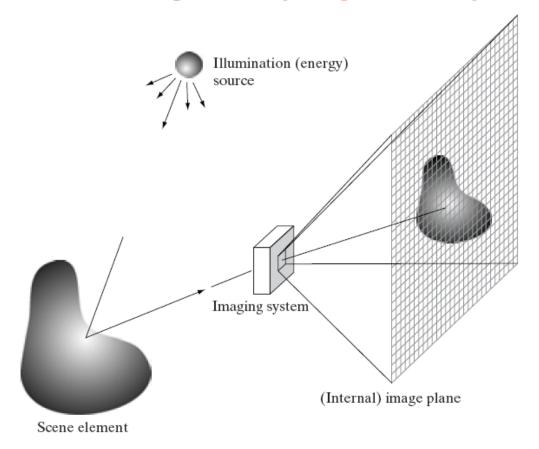
Viernes, I de agosto de 2008



- Direccionamiento e indexación
- Operaciones aritméticas entre matrices
- Operaciones lógicas sobre matrices
- Otras



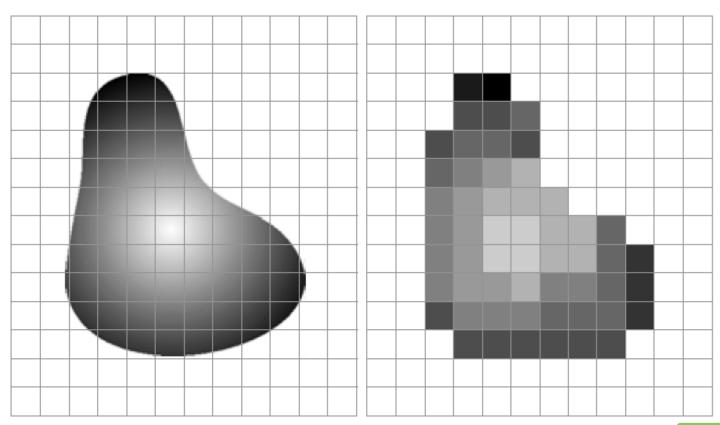
• Formación de imágenes (adquisición)



Viernes, I de agosto de 2008



• Formación de imágenes (muestreo y cuantiz.)



Viernes, I de agosto de 2008

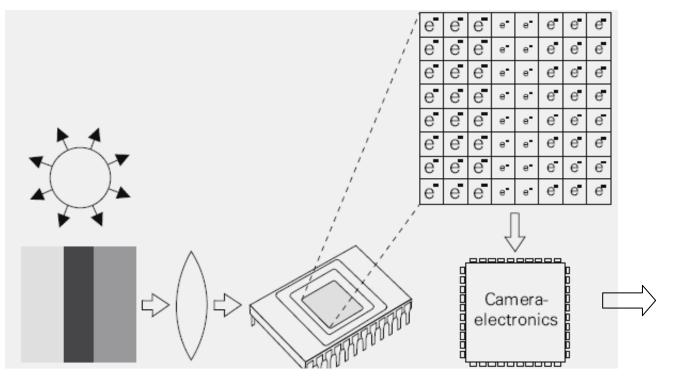
Taller de adquisición y procesamiento de imágenes y video mediante MATLAB





CONSTRUIMOS FUTURO

• Formación de imágenes (muestreo y cuantiz.)



220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130
220	220	220	50	50	130	130	130

Digital original image

Universida Industrial d Santande



Viernes, I de agosto de 2008

Taller de adquisición y procesamiento de imágenes y video mediante MATLAB

CONSTRUIMOS FUTURO



Acondicionamiento y ejecución del proceso de adquisición de video e imágenes



Image Adquisition Toolbox

- Capacita a MATLAB® para realizar un amplio conjunto de operaciones relacionadas con:
 - Adquisición de imágenes a través de dispositivos de registro (webcams USB >> frame grabbers).
 - Pre-visualizar un streaming de video "en vivo".
 - Triggering acquisitions.
 - Ejecutar callbacks ante ciertos eventos de la adquisición.
 - Exportar datos de imagen y video al workspace de MATLAB®



Pasos básicos de la adquisición a través de MATLAB®

- Paso 0: Instalación previa del dispositivo de adquisición.
- Paso I: Identificación de las propiedades del dispositivo.
- Paso 2: Creación de un objeto de adquisición.
- Paso 3: Previsualización del streaming de video.
- Paso 4: Configuración (adecuación) de las propiedades del objeto de adquisición.
- Paso 5: Adquisición de datos de imagen o video.
- Paso 6: Clausura de la adquisición.



Paso I - Propiedades del dispositivo

- Propiedades claves para acceder al dispositivo:
 - Nombre del adaptador
 - Interfaz software de conexión
 - Drivers
 - ID Identificador del dispositivo
 - Formato de video del streaming
 - Resolución
 - Composición del video
- Función asociada: >> imaqhwinfo



Paso I - Propiedades del dispositivo

Determinar el nombre del adaptador:

```
>> imaqhwinfo
ans =
    InstalledAdaptors: {'coreco' 'winvideo'}
        MATLABVersion: '7.5 (R2007b)'
        ToolboxName: 'Image Acquisition Toolbox'
        ToolboxVersion: '3.0 (R2007b)'
```

Determinar el ID del dispositivo:

```
>> imaqhwinfo('winvideo')
ans =
    AdaptorDllName: 'C:\Program Files\...\mwwinvideoimaq.dll'
AdaptorDllVersion: '3.0 (R2007b)'
    AdaptorName: 'winvideo'
    DeviceIDs: {[1]}
    DeviceInfo: [1x1 struct]
```

Viernes, I de agosto de 2008



Paso I - Propiedades del dispositivo

Determinar el formato de video del streaming:

Universidad Industrial de Santander

CONSTRUIMOS FUTURO

Paso 2 - Creación del objeto de adquisición

Objeto de adquisición

Objeto de video

Fuente de video

Fuente de video

Fuente de video

...

- Objeto de video >> Representa la conexión entre MATLAB® y el dispositivo de adquisición.
- Fuente de video >> Un mismo objeto de video puede contener múltiples fuentes de video.

Viernes, I de agosto de 2008



Paso 2 - Creación del objeto de adquisición

Creación de un objeto de video:

```
>> vid = videoinput('winvideo',1,'RGB24_320x240');
```

Obtener información del objeto de video:

```
>> imaqhwinfo(vid)
ans =

AdaptorName: 'winvideo'

DeviceName: 'Laptop Integrated Webcam'

MaxHeight: 240

MaxWidth: 320

NativeDataType: 'uint8'

TotalSources: 1

VendorDriverDescription: 'Windows WDM Compatible Driver'

VendorDriverVersion: 'DirectX 9.0'
```

Viernes, I de agosto de 2008

Taller de adquisición y procesamiento de imágenes y video mediante MATLAB



CONSTRUIMOS FUTURO

Paso 2 - Creación del objeto de adquisición

• >> vid

Summary of Video Input Object Using 'Laptop Integrated Webcam'.

Acquisition Source(s): input1 is available.

Acquisition Parameters: 'input1' is the current selected source.

10 frames per trigger using the selected source.

'RGB24_320x240' video data to be logged upon START.

Grabbing first of every 1 frame(s). Log data to 'memory' on trigger.

Trigger Parameters: 1 'immediate' trigger(s) on START.

Status: Waiting for START.

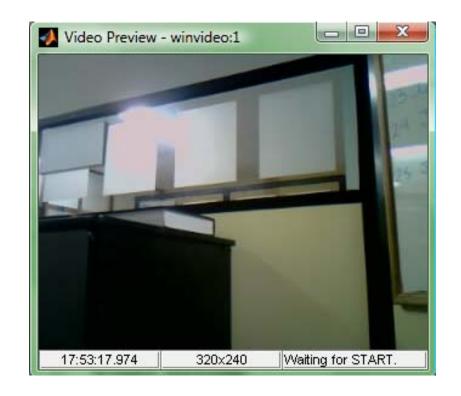
10 frames acquired since starting.
10 frames available for GETDATA.

Viernes, I de agosto de 2008



Paso 3 - Previsualización del streaming de video

- >> preview(vid)
- >> stoppreview(vid)
- >> closepreview(vid)



Viernes, I de agosto de 2008



- Adecuación propiedades: Controlar las características del video y como se adquiere.
- Identificar propiedades del objeto de video:
 - >> get(vid)
- Identificar propiedades de la fuente de video:
 - >> get(getselectedsource(vid))



• Identificar el valor de una propiedad específica:

```
>> get(vid,'Previewing')
>> get(getselectedsource(vid),'Gamma')
```

Obtener información de una propiedad específica:

```
>> propinfo(vid, 'VideoResolution')
>> propinfo(getselectedsource(vid),'Hue')
```

Obtener ayuda sobre una propiedad específica:

```
>> imaqhelp(vid, 'BayerSensorAlignment')
>> imaqhelp(getselectedsource(vid),'Contrast')
```

Universidad Industrial de Santander

• Lista de propiedades que pueden modificarse (settable):

```
>> set(vid)
>> set(getselectedsource(vid))
```

Modificación de una propiedad específica:

```
>> set(vid,'LoggingMode','disk&memory')
>> set(getselectedsource(vid),'Sharpness', 10)
```



Ejemplo aplicado de adecuación de las propiedades de la fuente de video:

>> gui_video_consola(vid)

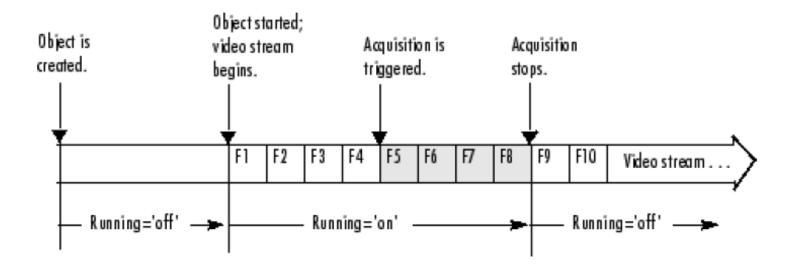


Viernes, I de agosto de 2008



• Iniciar el objeto de video: Prepara el objeto para adquirir; algunas propiedades se tornan se "sólo lectura".

>> start(vid)

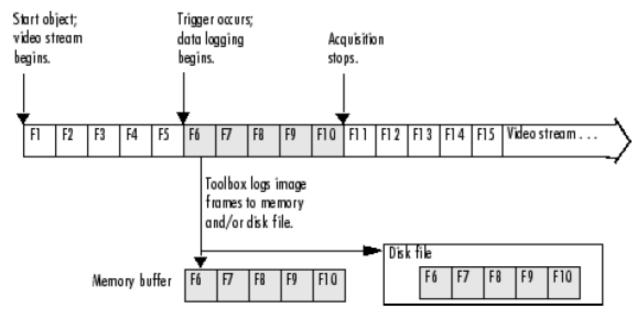


Viernes, I de agosto de 2008



Logging:

```
>> set(vid,'LoggingMode','disk')
>> set(vid,'LoggingMode','memory')
>> set(vid,'LoggingMode','disk&memory')
```

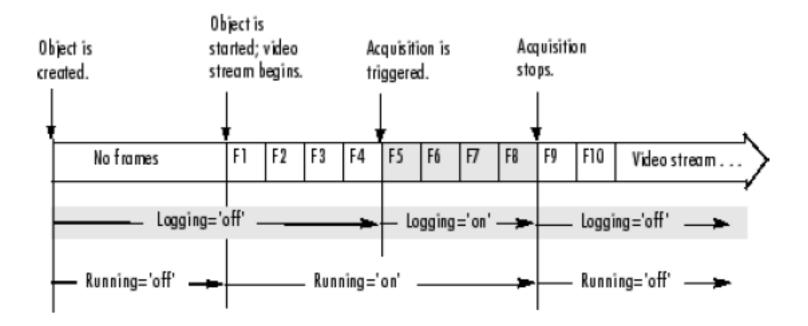


Viernes, I de agosto de 2008



• Trigger: Evento (inmediato o manual) que inicia la adquisición propiamente (logging) de frames.

Universidad Industrial de Santander

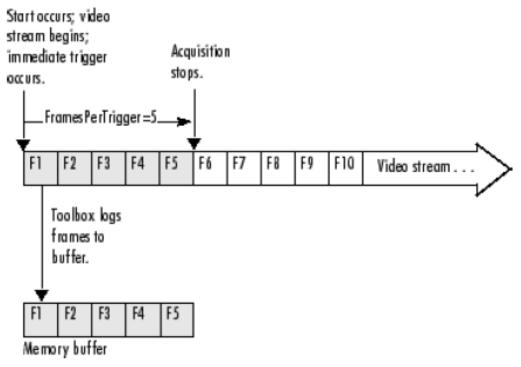


Viernes, I de agosto de 2008



Trigger inmediato:

```
>> start(vid)
>> get(vid,'FramesAcquired')
```



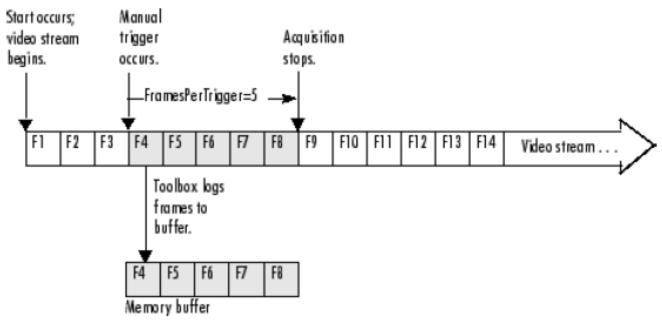
Viernes, I de agosto de 2008



Trigger manual:

```
>> triggerconfig(vid, 'manual')
```

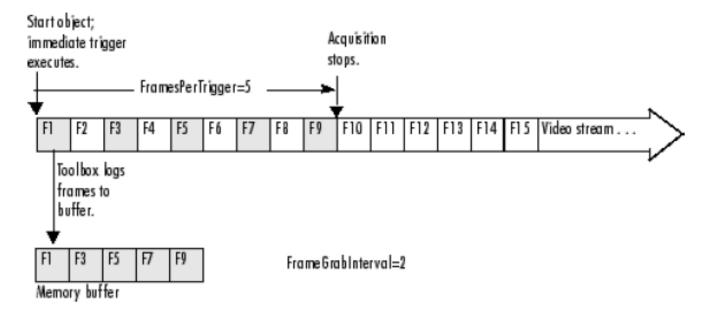
- >> start(vid)
- >> trigger(vid)
- >> get(vid,'FramesAcquired')



Viernes, I de agosto de 2008



- FramesPerTrigger, FrameGrabInterval:
 - >> set(vid,'FramesPerTrigger',5)
 - >> set(vid, `FrameGrabInterval',2)

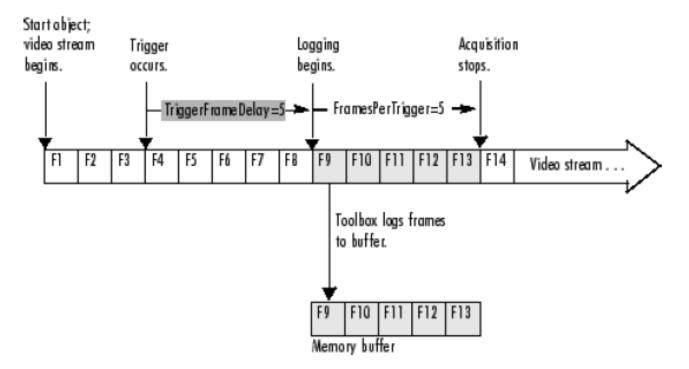


Viernes, I de agosto de 2008



TriggerFrameDelay:

>> set(vid, 'TriggerFrameDelay',5)

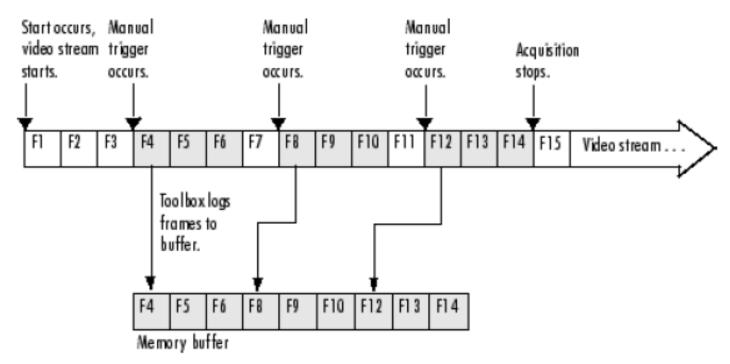


Viernes, I de agosto de 2008



• TriggerRepeat:

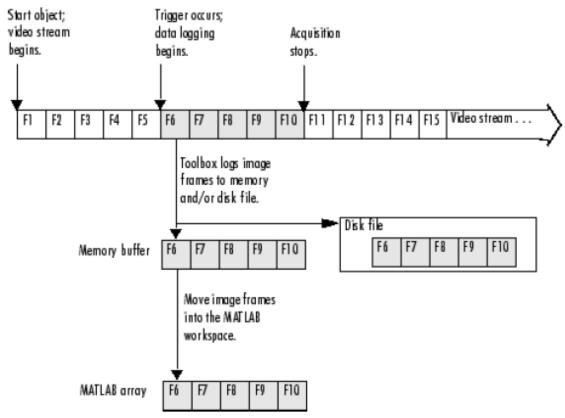
>> set(vid,'TriggerRepeat',2)



Viernes, I de agosto de 2008



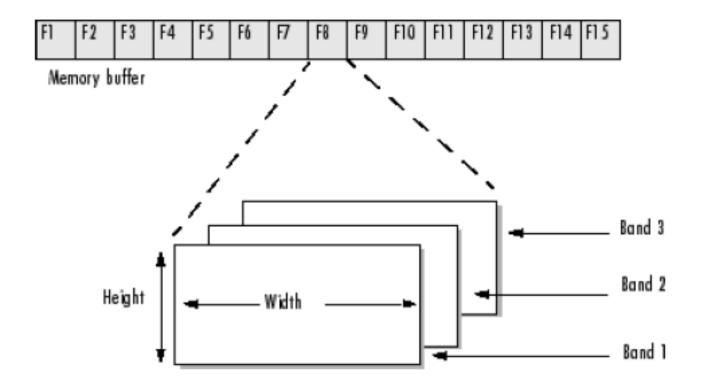
Exportación de frames



Viernes, I de agosto de 2008



• Exportación de frames



Viernes, I de agosto de 2008



- Exportar frames: Buffer de memoria >> workspace
 - Exportar múltiples frames con remoción

Viernes, I de agosto de 2008



- Exportar frames: Buffer de memoria >> workspace
 - Exportar múltiples frames sin remoción

Viernes, I de agosto de 2008



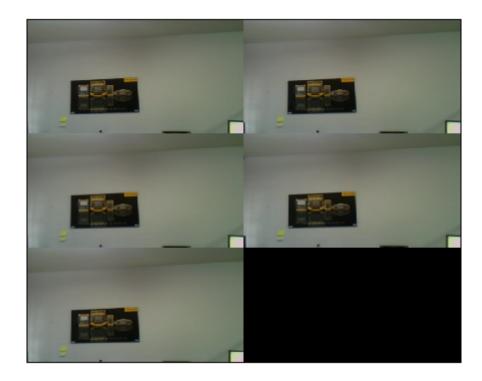
• Exportar frames: Buffer de memoria >> workspace

```
>> size(data)
ans =
     240     320     3     5
>> size(pdata)
ans =
     240     320     3     5
```

Viernes, I de agosto de 2008



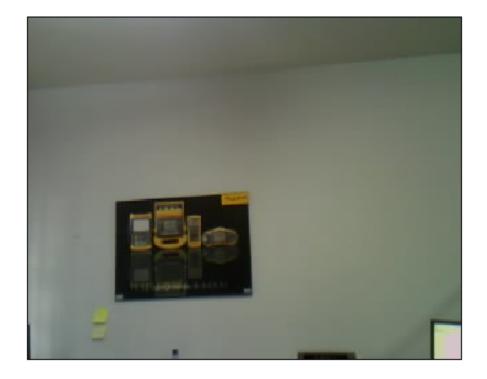
- Visualizar multiples frames:
- >> imaqmontage(data)



Viernes, I de agosto de 2008



- Visualizar un frame de una secuencia múltiple:
- >> imshow(data(:,:,:,1))

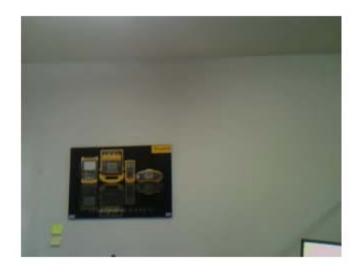


Viernes, I de agosto de 2008



Exportar un frame: Objeto de video >> workspace

```
>> vid = videoinput('winvideo',1,'RGB24_320x240');
>> frame = getsnapshot(vid);
>> imshow(frame)
```



Viernes, I de agosto de 2008



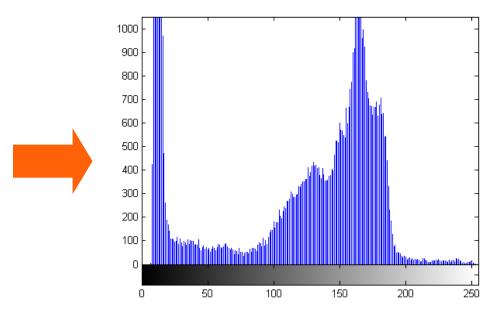


Procesamiento de imágenes y video utilizando MATLAB®



Histograma





- >> I = imread('cameraman.tif');
- >> figure, imshow(I); figure, imhist(I)
- Distribución de la frecuencia con que aparecen los niveles de gris de una imagen.
- No aporta información sobre la distribución espacial de los niveles de gris.

Viernes, I de agosto de 2008

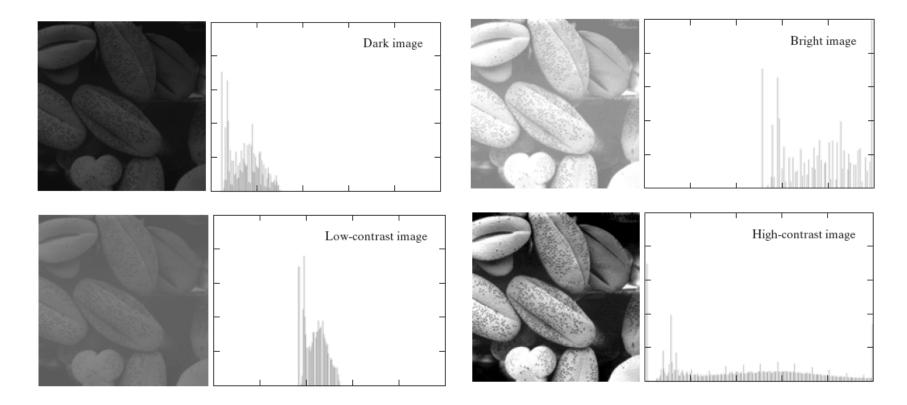
Taller de adquisición y procesamiento de imágenes y video mediante MATLAB





CONSTRUIMOS FUTURO

Contraste

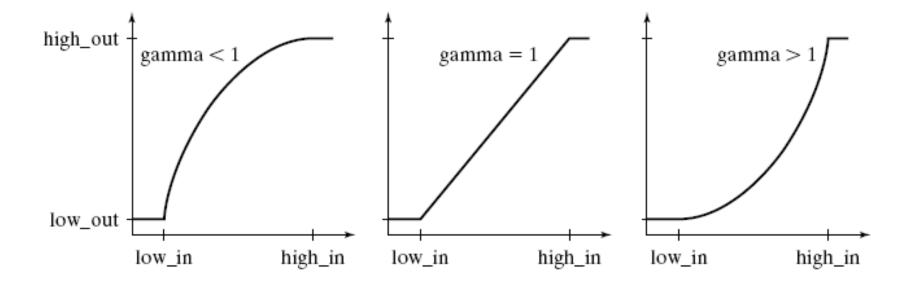


- Mide el rango dinámico de los niveles de gris en una imagen.
- Puede mejorarse por medio de una transformación lineal.

Viernes, I de agosto de 2008



Transformaciones de intensidad



J = imadjust(I, [low_in high_in], [low_out high_out], gamma)

Viernes, I de agosto de 2008

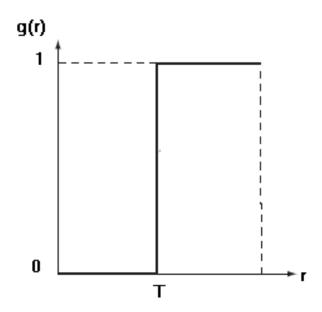


Segmentación

- Subdivide una imagen en sus regiones constitutivas u objetos.
- Algunos algoritmos se basan en propiedades básicas los niveles de intensidad:
 - Discontinuidad
 - Similitud



Segmentación basada en umbralización



$$g(x,y) = \begin{cases} 1 & si \quad f(x,y) \ge T \\ 0 & si \quad f(x,y) < T \end{cases}$$

- Asigna un valor lógico de I o 0 a los píxeles de una imagen según un umbral de intensidad, T.
- Separa el objeto de interés del fondo de la escena.
- Requiere condiciones controladas de registro, fondo uniforme, sin texturas, etc.

Viernes, I de agosto de 2008

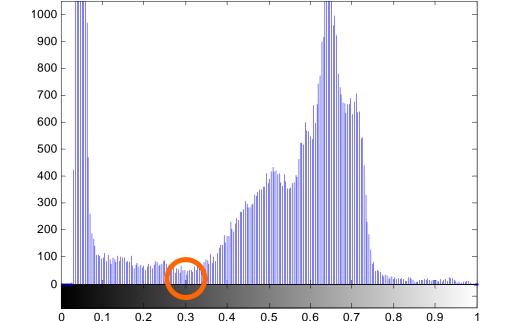


Segmentación basada en umbralización









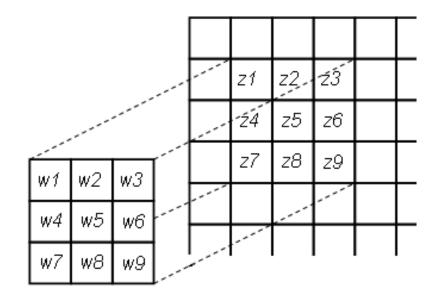
>> K = im2bw(I,0.3);

Viernes, I de agosto de 2008



Filtrado espacial

- Operaciones realizadas directamente sobre los píxeles de la imagen.
- Se convoluciona
 empleando una máscara
 de m x n con coeficientes.
- Se utilizan técnicas de rellenado (padding).



$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$

$$R = \sum_{i=1}^{9} w_i z_i$$



Filtro pasa-bajo

Font size=5
Font size=4
Font size=3
Font size=2
Font size=2
Font size=1

Otras máscaras:

$$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{c|cccc}
 & 1 & 2 & 1 \\
 & 1 & 2 & 4 & 2 \\
 & 1 & 2 & 1 \\
 & 1 & 2 & 1
\end{array}$$

Viernes, I de agosto de 2008

Taller de adquisición y procesamiento de imágenes y video mediante MATLAB



CONSTRUIMOS FUTURO



Filtro paso-bajo (mediana)

- No lineal y de ordenamiento estadístico.
- Computacionalmente ineficiente.
- Degrada la imagen en menor cuantía que otros filtros paso bajo.
- Excelente reducción de ruido impulsivo (salt & pepper).



Filtro pasa-bajo







Imagen con ruido salt & pepper

Filtro promedio (máscara 3 x 3)

Filtro mediana (máscara 3 x 3)

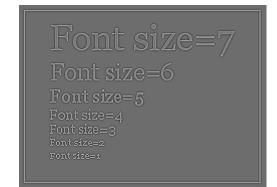
Viernes, I de agosto de 2008

Taller de adquisición y procesamiento de imágenes y video mediante MATLAB



Filtro pasa-alto

Font size=2
Font size=1



Otras máscaras:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

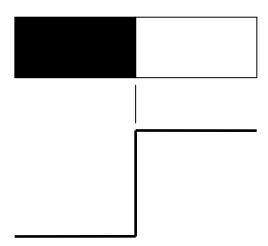
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Viernes, I de agosto de 2008

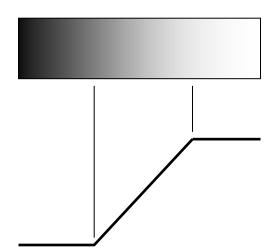




• Frontera entre dos regiones con nivel de gris relativamente diferentes.



Modelo ideal de un borde digital en rampa



Modelo en rampa de un borde digital

Viernes, I de agosto de 2008



 Detector de bordes >> Operador local de derivación.





$$\overline{\nabla}I = \begin{bmatrix} Gx \\ Gy \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$$



$$\nabla I = mag(\overline{\nabla}I) = \sqrt{Gx^2 + Gy^2}$$



$$\nabla I \approx |Gx| + |Gy|$$

$$\alpha(x, y) = \tan^{-1} \left(\frac{Gy}{Gx} \right)$$

Viernes, I de agosto de 2008

Taller de adquisición y procesamiento de imágenes y video mediante MATLAB





CONSTRUIMOS FUTURO

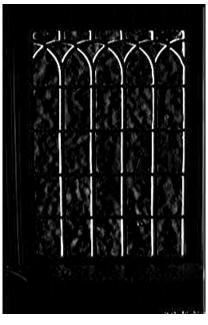
Operador \ Deriv. Parcial	Gx	Gy
Prewitt	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
Roberts	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$ \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} $
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$ \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} $

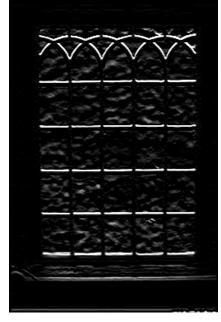
Viernes, I de agosto de 2008











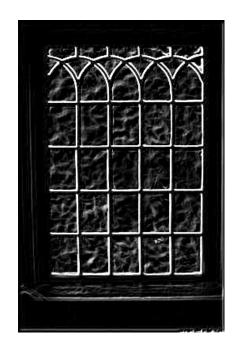


Imagen original

Gx - Sobel

Gy - Sobel

Gradiente - Sobel

Viernes, I de agosto de 2008



- Derivada de segundo orden >> Laplaciano.
- Sensible en exceso al ruido, produce bordes dobles y no permite determinar direcciones.

$$\nabla I^{2} = \frac{\partial^{2} I}{\partial x^{2}} + \frac{\partial^{2} I}{\partial y^{2}} \longrightarrow \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Viernes, I de agosto de 2008



Filtrado espacial mediante MATLAB®

Procedimiento básico:

```
% Filtros predefinidos
>> w = fspecial(tipo, parametro);
% Filtro definido
>> w = [a b c; d e f; g h i];
% Filtrado por convolución
>> J = imfilter(I, w, padding);
```

Viernes, I de agosto de 2008



Bibliografía

- GONZÁLEZ, Rafael C. y WOODS, Richard E. Digital Image Processing. 2da ed. New Jersey, Prentice Hall, 2002. 793 p.
- GONZÁLEZ, Rafael C., WOODS, Richard E. y EDDINS, Steven. Digital Image Processing Using MATLAB®. New Jersey, Prentice Hall, 2004. 782 p.
- THE MATHWORKS™, Image Adquisition Toolbox User's Guide.
- THE MATHWORKS[™], Image Processing Toolbox User's Guide.





Gracias por atención...





Viernes, I de agosto de 2008

