Image Acquisition and Analysis in MATLAB

Presenter: Claire Chuang
TeraSoft Inc.
Outline

- Hardware Connections
- Examining Hardware Resources
- Create a Video Input Object
- Configuring Properties
- Preview the Video Stream
- Acquiring Image Data
- Deleting and Clearing Toolbox Objects

Image Acquisition Application

This section shows you an example of:

1. Accessing the webcam’s video input.
2. Acquiring an image and store as a MATLAB variable.
3. Bring the acquired image into MATLAB, process, and visualize.
Hardware Connections

- Connect the image acquisition device to your PC (In our case, a USB port connected webcam).
- Check the hardware with the application software provided by the manufacturer.

![Webcam Connection](image)


Examining Hardware Resources

- You should examine the image acquisition hardware resources visible to the toolbox with the `imaqhwinfo` function.

  - General toolbox information
    ```matlab
    >> a = imaqhwinfo;
    >> a.InstalledAdaptors
    ans =
    'coreco' 'winvideo'
    ```

  - Adaptor-specific information
    ```matlab
    >> b = imaqhwinfo('winvideo');
    ```
Creating a Video Input Object – Basics

- Create a video input object with the `videoinput` constructor.
  
  ```matlab
  >> vid = videoinput('winvideo',1);
  ```

  where:
  - `vid` is called a video input object.
  - `winvideo` is the adaptor name for image acquisition hardware.
  - `1` is the device ID (given by `imaqhwinfo`)

- `vid` is a custom MATLAB class.
  
  ```matlab
  >> whos vid
  ```

Configuring Properties – Basics

- Use the `set` function to display all configurable device object properties.
  
  ```matlab
  >> set(vid)
  ```

- Use the `get` function to return the current device object property values.
  
  ```matlab
  >> get(vid)
  ```

- Use the image acquisition property editor to view and edit the device object's properties.
  
  ```matlab
  >> inspect(vid)
  ```
Configuring Properties – Summary

The Image Acquisition Toolbox provides the following functions to return property information.

<table>
<thead>
<tr>
<th>Function</th>
<th>Property information provided</th>
</tr>
</thead>
<tbody>
<tr>
<td>set</td>
<td>All configurable properties</td>
</tr>
<tr>
<td></td>
<td>Possible value strings, with the default contained by braces</td>
</tr>
<tr>
<td>get</td>
<td>All properties and their current values</td>
</tr>
<tr>
<td>propinfo</td>
<td>Runtime property characteristics</td>
</tr>
<tr>
<td></td>
<td>Property value data type</td>
</tr>
<tr>
<td></td>
<td>Default property values</td>
</tr>
<tr>
<td></td>
<td>Constraints (min and max values)</td>
</tr>
<tr>
<td>imaqhwinfo</td>
<td>Hardware-related property values</td>
</tr>
<tr>
<td></td>
<td>- Maximum height and width</td>
</tr>
<tr>
<td></td>
<td>- Vendor driver version</td>
</tr>
<tr>
<td>inspect</td>
<td>Image acquisition object property editor</td>
</tr>
</tbody>
</table>

Preview Video Stream

The data extraction process is shown below.

```
>> data = peekdata(vid,5);
>> montage(data);
```

```
>> preview(vid)
```

**Note** peekdata is an non-blocking function that immediately returns data and execution control to MATLAB.
Acquiring Image Data - Extracting (Continued)

The data extraction process is shown below.

Extract data from engine

Log data to engine

4-D data array

m-by-1 time array

>> [data,time] = getdata(ai);

Note: getdata blocks the command line until the requested number of samples is returned or a timeout occurs.

Use the getdata function to extract acquired data from the engine.

- Extract the specified number of frames.
  
  ```
  >> data = getdata(vid,5);
  ```

- Extract all the data associated with one trigger.
  
  ```
  >> data = getdata(vid);
  >> data = getdata(vid,get(vid,'FramesPerTrigger'));
  ```

- Extract all the data available in the engine.
  
  ```
  >> data = getdata(ai,get(vid,'FramesAvailable'));
  ```
Deleting and Clearing Toolbox Objects

- Remove variables from the MATLAB workspace.
  
  
  `>> delete(vid)`
  `>> clear vid`

- To quickly delete all device objects and reset the hardware to its initial state
  
  `>> imaqreset`

---

The Video Input Session

```
>> vid = videoinput('winvideo',1);
>> set(vid,'ReturnedColorSpace','rgb');
>> triggerconfig(vid,'immediate');
>> set(vid,'FramesPerTrigger',10);
>> preview(vid);
>> start(vid);
>> data = getdata(vid);
>> imshow(data(:,:,end,:));
>> stop(vid);
>> delete(vid)
>> clear vid
```
Image Processing Toolbox

Perform image processing, analysis, visualization, and algorithm development

- Graphical tools
- Image pre- and postprocessing
- Image analysis
- Spatial transformations
- Image registration
- Color image processing

Copyright © 2007 by TeraSoft, Inc.

Image Pre- and Postprocessing

- Contrast enhancement
- Noise removal
- Deblurring
- Region-based processing
- Linear and nonlinear filtering

Original image courtesy of MIT. This version created by simulating motion blur.

Deblurred image using Wiener filter deconvolution.

Original Landsat image courtesy of Space Imaging, LLC.
Image Analysis

- Edge detection
- Segmentation
- Morphological operators
- Image statistics
- Boundary tracing
- Region properties
- Texture analysis
- Hough transform

Q & A
Thanks!!