

# ACQUISITION D'IMAGES ET DE VIDEOS SUR MATLAB

De nos jours, la plupart des caméras sont disponibles avec une interface USB. Une fois que vous installez le pilote du capteur d'images, l'ordinateur détecte le périphérique chaque fois que vous le connectez. Alternativement, si vous avez un caméscope ou une caméra CCD vidéo numérique connecté à une carte d'acquisition et interfacé avec un ordinateur, Windows OS détecte automatiquement le périphérique. Pour exécuter les instructions ci-dessous, vous aurez besoin d'une webcam USB fonctionnelle, connectée à votre PC.

## 1. Collecte d'informations sur l'environnement d'acquisition:

Dans MATLAB, vous pouvez vérifier si le support est disponible pour votre caméra. MATLAB a des adaptateurs intégrés pour accéder à ces dispositifs. Un adaptateur est un logiciel que MATLAB utilise pour communiquer avec un capteur d'images.

Toutes ces informations sont obtenues en tapant la commande suivante:

```
>> imaqhwinfo
ans =
    InstalledAdaptors: {'gentl' 'gige' 'matrox' 'winvideo'}
    MATLABVersion: '8.1 (R2013a)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '4.5 (R2013a)'
```

Pour obtenir plus d'informations sur le capteur, taper:

```
>> dev_info = imaqhwinfo('winvideo',1)
dev_info =
    DefaultFormat: 'YUY2_160x120'
    DeviceFileSupported: 0
    DeviceName: 'Sirius USB2.0 Camera'
    DeviceID: 1
    VideoInputConstructor: 'videoinput('winvideo', 1)'
    VideoDeviceConstructor: 'imaq.VideoDevice('winvideo', 1)'
    SupportedFormats: {1x5 cell}
```

## 2. Prévisualisation de la vidéo:

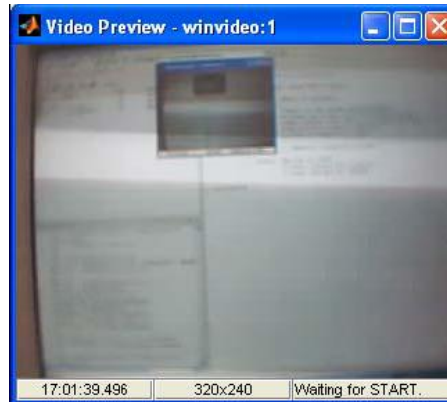
Vous pouvez visionner la vidéo capturée en définissant un objet et l'associer au capteur. Vous verrez les détails des paramètres d'acquisition, comme le montre la commande suivante:

```
>> vid=videoinput('winvideo',1, 'YUY2_160x120')
Summary of Video Input Object Using 'Sirius USB2.0 Camera'.
Acquisition Source(s): input1 is available.
Acquisition Parameters: 'input1' is the current selected source.
10 frames per trigger using the selected source.
'YUY2_160x120' 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.
0 frames acquired since starting.
0 frames available for GETDATA.
```

Maintenant, pour voir la vidéo capturée par la caméra, utilisez la commande "preview"

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

Vous devriez voir une fenêtre qui affiche ce que votre caméra capture:



Le capteur d'images peut prendre en charge plusieurs formats vidéo. Pour voir par vous-même tous les formats pris en charge, tapez:

```
>> dev_info = imaqhwinfo('winvideo',1);  
>> celldisp(dev_info.SupportedFormats); %displays list of supported formats
```

Vérifiez par vous-même l'affichage d'autres formats, en remplaçant 'YUY2\_160x120' par d'autres formats:

```
ans{1} =  
    YUY2_160x120  
ans{2} =  
    YUY2_176x144  
ans{3} =  
    YUY2_320x240  
ans{4} =  
    YUY2_352x288  
ans{5} =  
    YUY2_640x480
```

### **3. Capture et sauvegarde des images**

#### **3.1 Capture d'une image**

Maintenant, pour capturer une image dans la vidéo, définir l'objet "vid" comme décrit avant et utiliser getdata pour capturer une image de la vidéo.

```
>>start(vid);           % Cette commande démarre la capture des images et les sauvegarde en mémoire  
>>im=getdata(vid,1);  
>>figure,imshow(im);
```

#### **3.2 Sauvegarde de l'image**

Vous pouvez stocker l'image capturée comme un fichier « \*.jpeg » ou « \*.gif » en utilisant la fonction imwrite.

```
>>imwrite(im,'testimage.gif');
```

L'image sera enregistrée, par défaut, dans le dossier : "MATLAB\ work".

#### 4. Plus de maîtrise de l'acquisition d'images

Chaque fois que vous voulez capturer une image instantanée, vous devez arrêter la vidéo, commencer à nouveau, puis utiliser la fonction « getdata » ou « peekdata ».

Pour éviter ces actions répétitives, la boîte à outils Image Acquisition fournit une option de déclenchement de l'objet vidéo en cas de besoin et de capturer une trame instantanée.

Pour cela, créer un script Matlab avec la séquence suivante:

```
vid=videoinput('winvideo',1);
triggerconfig(vid,'manual');
set(vid,'FramesPerTrigger',1 );
set(vid,'TriggerRepeat', Inf);
start(vid);
for i=1 : 5
    trigger(vid);
    im= getdata(vid,1);
    figure,imshow(im);
end
stop(vid), delete(vid), clear vid;
```

Dans le code ci-dessus, l'objet " im " est écrasé à chaque exécution de la boucle FOR. Pour être en mesure de voir toutes les cinq images, remplacer "im" avec " im (:,:, i)".

Les instructions de la ligne 2 à la ligne 5 définissent les propriétés de l'objet "vid" :

- « Triggerconfig » définit l'objet avec un déclenchement manuel.  
Par défaut, le déclenchement est de type immédiat. Dans le déclenchement immédiat, la vidéo est capturée dès que vous lancez l'objet «vid». Les images capturées sont alors stockées dans la mémoire.
- La fonction « getdata » peut être utilisée pour accéder à ces images. Mais dans le déclenchement manuel, vous obtenez l'image que lorsque vous déclenchiez (trigger) la vidéo.
- « FramesPerTrigger » décide le nombre d'images que vous voulez capturer chaque fois que la commande « trigger » est exécutée.
- « TriggerRepeat » doit être soit égal au nombre d'images que vous souhaitez traiter dans votre programme ou il peut être réglé sur « Inf ». Si sa valeur est « Inf », vous pouvez utiliser «trigger» autant de fois que vous le souhaitez. Si la valeur est un entier positif, vous aurez à «démarrer» la capture vidéo à nouveau après que « trigger » est utilisé pour le nombre de fois.

En outre, une fois que vous avez terminé avec l'acquisition et le stockage d'images, vous pouvez arrêter la capture vidéo et effacer les images stockées dans la mémoire tampon, en utilisant les commandes suivantes:

```
>>stop(vid);
>>delete(vid);
>>clear vid;
```

#### **Note:**

"getsnapshot" retourne une trame d'image et est indépendante de la propriété FramesPerTrigger.

Donc, si vous voulez l'enregistrement continu d'images, vous pouvez écrire un script \*.m comme suit:

```
vid=videoinput('winvideo',1)
triggerconfig(vid,'manual');
set(vid,'FramesPerTrigger',1);
set(vid,'TriggerRepeat', Inf);
start(vid);
while(1)
```

## 5. Les fonctions de l'Image Acquisition Toolbox

### 5.1 Device Connection

<b>clear</b>	Clear image acquisition object from MATLAB workspace
<b>delete</b>	Remove image acquisition object from memory
<b>disp</b>	Display method for image acquisition objects
<b>imaqfind</b>	Find image acquisition objects
<b>imaqhwinfo</b>	Information about available image acquisition hardware
<b>imaqreset</b>	Disconnect and delete all image acquisition objects
<b>imaqtool</b>	Launch Image Acquisition Tool
<b>start</b>	Obtain exclusive use of image acquisition device
<b>stop</b>	Stop video input object
<b>videoinput</b>	Create video input object
<b>imaq.VideoDevice</b>	Acquire one frame at a time from video device

### 5.2 Image Preview and Device Configuration

<b>closepreview</b>	Close Video Preview window
<b>get</b>	Return image acquisition object properties
<b>getselectedsource</b>	Return currently selected video source object
<b>imaqhelp</b>	Image acquisition object function and property help
<b>imaqtool</b>	Launch Image Acquisition Tool
<b>preview</b>	Preview of live video data
<b>propinfo</b>	Property characteristics for image acquisition objects
<b>set</b>	Configure or display image acquisition object properties
<b>stoppreview</b>	Stop previewing video data

### 5.3 Image Data Acquisition

<b>imaqtool</b>	Launch Image Acquisition Tool
<b>videoinput</b>	Create video input object
<b>imaq.VideoDevice</b>	Acquire one frame at a time from video device
<b>getdata</b>	Acquired image frames to MATLAB workspace
<b>peekdata</b>	Most recently acquired image data
<b>getsnapshot</b>	Immediately return single image frame
<b>set</b>	Configure or display image acquisition object properties
<b>start</b>	Obtain exclusive use of image acquisition device
<b>islogging</b>	Determine whether video input object is logging
<b>isrunning</b>	Determine whether video input object is running
<b>isvalid</b>	Determine whether image acquisition object is associated with image acquisition device
<b>wait</b>	Wait until image acquisition object stops running or logging
<b>stop</b>	Stop video input object
<b>clear</b>	Clear image acquisition object from MATLAB workspace
<b>delete</b>	Remove image acquisition object from memory
<b>flushdata</b>	Remove data from memory buffer used to store acquired image frames

<b>obj2mfile</b>	Convert video input objects to MATLAB code
<b>imaqmontage</b>	Sequence of image frames as montage
<b>load</b>	Load image acquisition object into MATLAB workspace
<b>save</b>	Save image acquisition objects to MAT-file
<b>trigger</b>	Initiate data logging
<b>triggerconfig</b>	Configure video input object trigger properties
<b>triggerinfo</b>	Provide information about available trigger configurations

## 5.4 Acquisition Using the Image Acquisition Tool

<b>imaqtool</b>	Launch Image Acquisition Tool
-----------------	-------------------------------

### **Bibliographie :**

- <http://www.nex-robotics.com/workshop.html>
- [www.mathworks.com/support](http://www.mathworks.com/support)