

# Relevé manuel de points sur une image ou une vidéo

---

## Ouverture de l'interface

Le module CV comporte les fonctions d'interface pour openCV plus quelques utilitaires. Pour charger le module et ouvrir l'interface :

```
In[24]:= Needs["CV`"];  
CVOpen[];
```

---

## Lecture d'une image et relevé des points

La fonction suivante permet de lire une image à partir d'un fichier (JPG, PNG, etc). Le chemin complet du fichier doit être indiqué (on peut aussi choisir le répertoire courant auparavant avec **SetDirectory**).

```
In[28]:=  
img = CVLoadImage[$BaseDirectory <> "/Applications/CV/pic5.png", CVLoadImageUnchanged];
```

On commence par créer un éditeur d'image. Le deuxième argument est True pour que l'éditeur soit une fenêtre modale.

```
In[29]:= edit = CVJImageEditor["formes", True];
```

Puis on fournit l'image à l'éditeur :

```
In[31]:= CVJShowImage[edit, img];
```

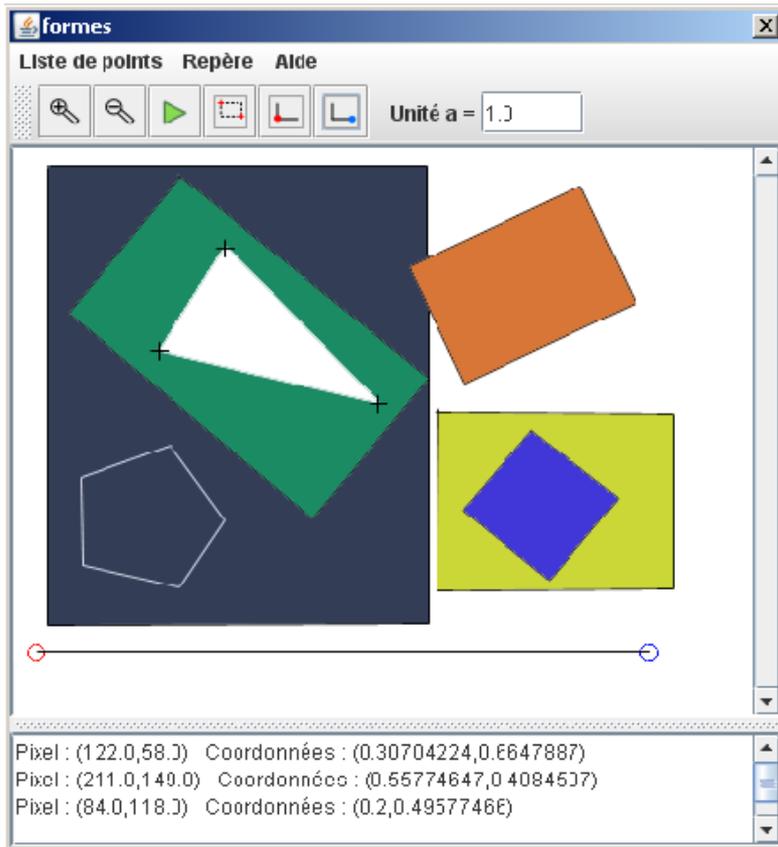
Si la fenêtre est modale, l'exécution du programme *Mathematica* est bloquée jusqu'à la fermeture de la fenêtre. Sinon, les instructions suivantes du bloc sont exécutées tout de suite.

L'éditeur permet de sélectionner des points dans l'image. Les coordonnées sont de deux types :

- Coordonnées en unité de pixels, avec l'origine au coin supérieur gauche.
- Coordonnées par rapport à un repère orthogonal direct défini par l'utilisateur.

Le repère est défini par son origine (0,0) et le point de coordonnées (a,0), où a est une unité modifiable. Il est aussi possible de définir le point (0,a), ce qui revient à définir l'axe Oy (option Définir l'axe Y dans le menu Repère).

Une fois les points saisis, on peut revenir au notebook sans fermer la fenêtre. Une autre image peut être envoyée à l'éditeur sans que le repère ne soit modifié, ni même la liste de points.



Sur cette image, un repère a été défini en bas de l'image par les points (0,0) et (a,0). Trois points ont été sélectionnés aux sommets du triangle.

---

## Récupération de la liste des points

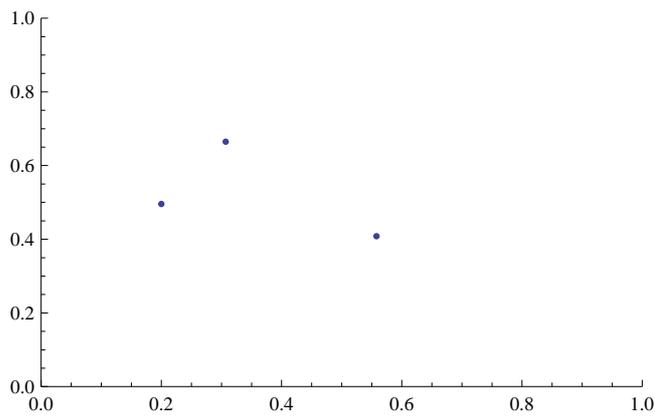
La liste des points en unité de pixels est obtenue par :

```
liste1 = CVJGetPixelPointList [edit]
{{122., 58.}, {211., 149.}, {84., 118.}}
```

La liste des points en coordonnées utilisateurs est obtenue par :

```
liste2 = CVJGetUserPointList [edit]
{{0.307042, 0.664789}, {0.557746, 0.408451}, {0.2, 0.495775}}
```

```
ListPlot[list2, PlotRange -> {{0, 1}, {0, 1}}]
```




---

## Enregistrement des listes sur fichier

Pour utiliser ces listes ultérieurement sans avoir à refaire la saisie sur l'image, il faut les sauvegarder sur fichier. Ceci est obtenu de la manière suivante :

```
Export["liste1.dat", liste1, "Table"]
```

```
liste1.dat
```

Pour récupérer la liste dans un autre notebook :

```
liste1 = Import["liste1.dat", "Table"]
```

```
{{122., 58.}, {211., 149.}, {84., 118.}}
```

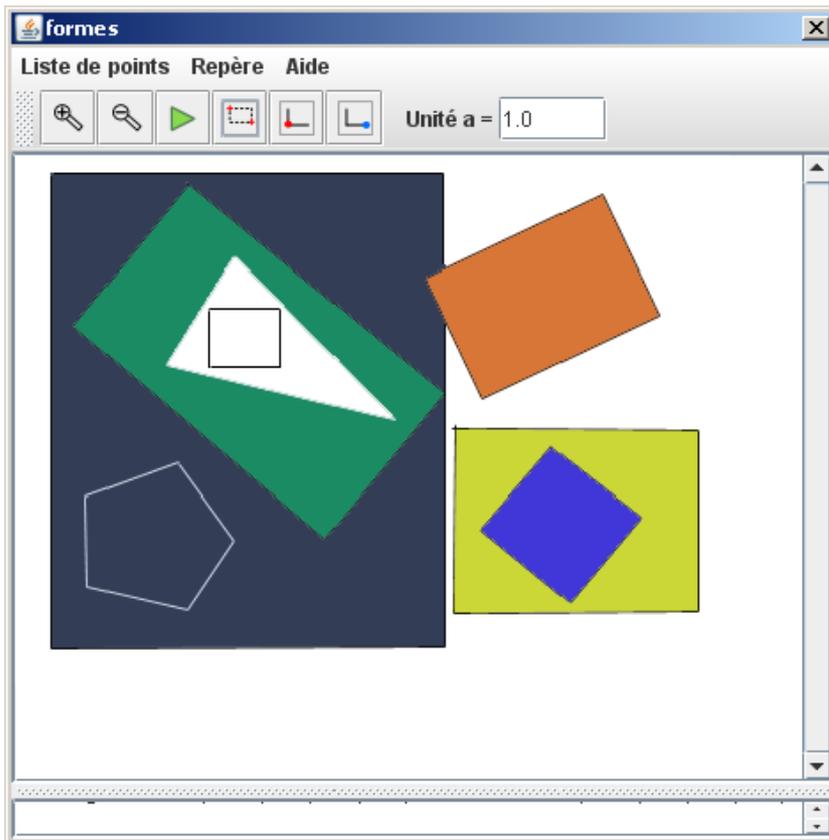
---

## Sélection de rectangles

L'éditeur d'image peut aussi servir à sélectionner des zones rectangulaires dans l'image :

```
edit = CVJOpenImageEditor["formes"];
```

```
CVJShowImage[edit, img];
```



La liste des rectangles sélectionnés (ici un seul) est récupérée par :

```
listeR = CVJGetPixelRectangleList [edit]
{{108., 86., 40., 33.}}
```

---

## Lecture d'une vidéo et relevé des points

La fonction suivante permet de lire une vidéo à partir d'un fichier (le décodage est effectué par ffmpeg) :

```
In[32]:= video = CVCaptureFromFile[$BaseDirectory <> "/Applications/CV/toupie.mpg"];
```

Pour obtenir une copie de la 10 ième image de la vidéo :

```
img10 = CVGetFrame [video, 10];
```

Cette opération a pour effet de placer le flux vidéo sur la 10ième image. Pour revenir au début de la vidéo, il faut la charger à nouveau avec la fonction **CVCaptureFromFile**.

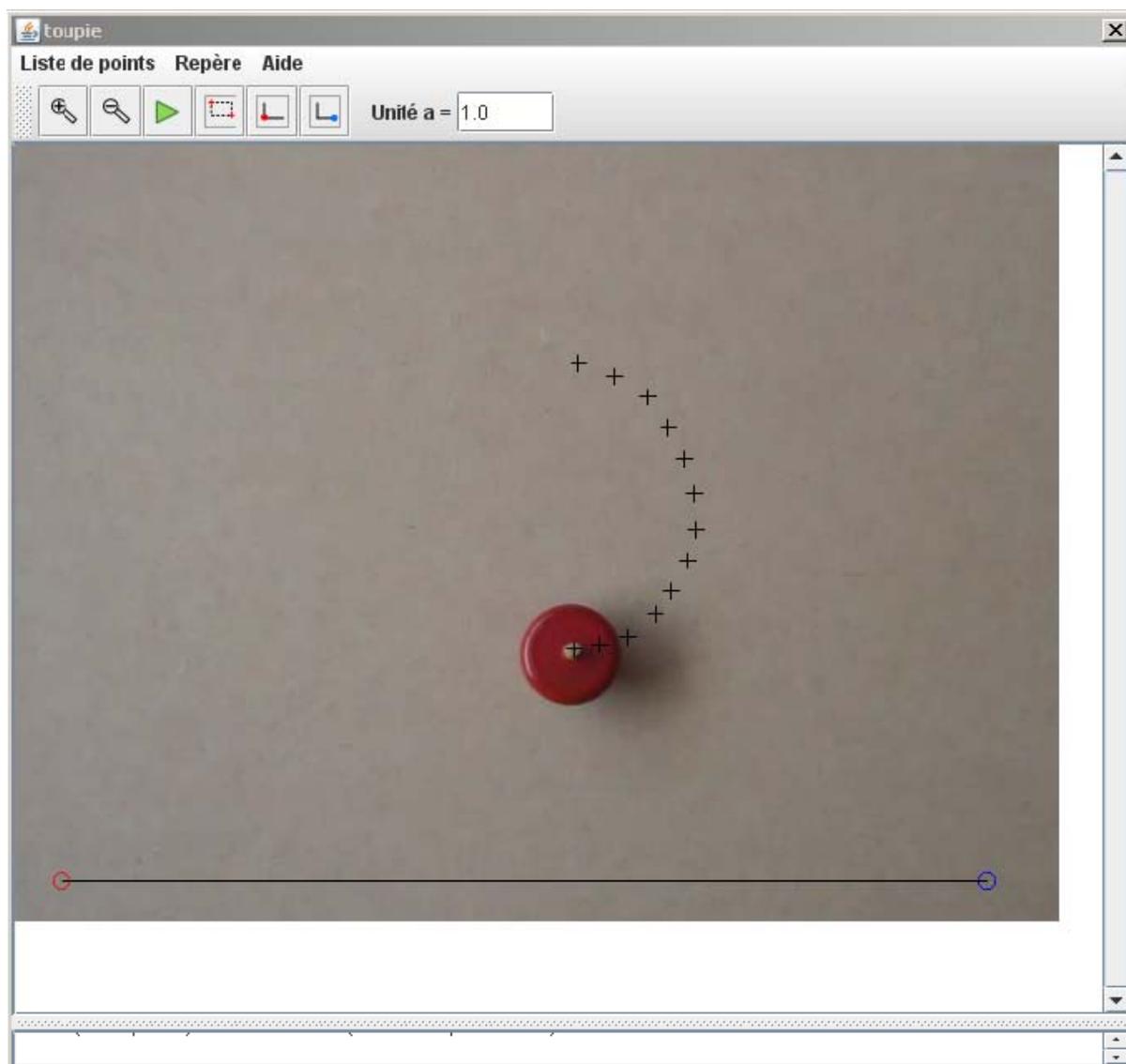
Pour relever des points sur plusieurs images d'une vidéo (par exemple pour relever une trajectoire), on ouvre l'éditeur d'image puis on lui fournit le flux vidéo à lire :

```
In[33]:= edit = CVJImageEditor ["toupie", True];
CVJShowVideoFrames [edit, video, 10];
```

Le dernier argument est le nombre d'images à sauter entre deux images affichées consécutives. Dans l'exemple présent, il s'agit d'une vidéo prise à 60 images par secondes. Les images affichées sont donc espacées de  $10/60 = 1/6$  ième de secondes. Le calcul du temps est à la charge de l'utilisateur car le nombre d'images par seconde (fps) dépend du matériel utilisé et généralement n'apparaît pas de manière correcte dans le fichier vidéo.

La sélection des points se fait comme sur une image fixe, à la différence qu'il est possible de faire défiler les images avec la flèche verte. Bien sûr, le repère utilisateur doit être défini avant la saisie des points, si possible à l'aide d'une règle apparaissant au bord du champ (ou simplement sur les premières images). L'éditeur permet de sélectionner autant de points que l'on souhaite dans chaque image.

Dans le cas présent, on sélectionne un point par image (le centre de la toupie) de manière à obtenir une trajectoire.



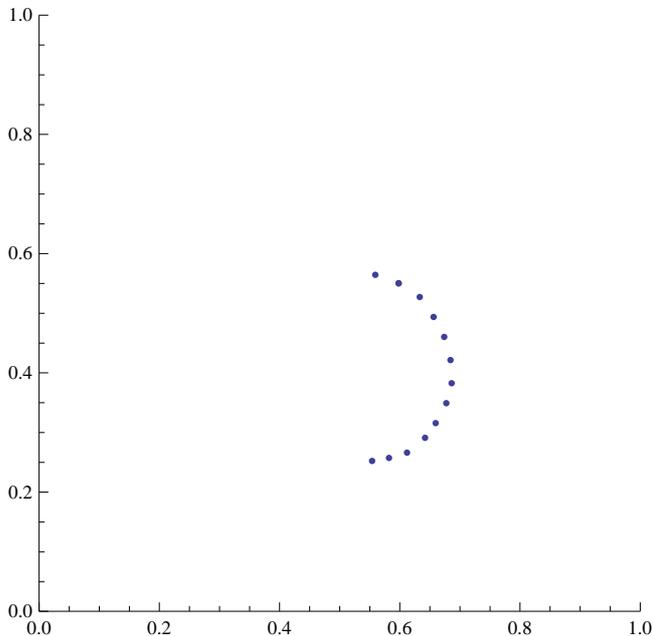
La liste des points est récupérée comme précédemment :

```

liste3 = CVJGetUserPointList[edit]
ListPlot[liste3, PlotRange -> {{0, 1}, {0, 1}}, AspectRatio -> 1]

```

{{0.559083, 0.564374}, {0.597884, 0.550265}, {0.597884, 0.550265},  
 {0.633157, 0.527337}, {0.656085, 0.493827}, {0.673721, 0.460317},  
 {0.684303, 0.421517}, {0.686067, 0.382716}, {0.677249, 0.349206}, {0.659612, 0.315697},  
 {0.641975, 0.291005}, {0.611993, 0.266314}, {0.582011, 0.257496}, {0.553792, 0.252205}}



En cas de sélection de points plus complexe, par exemple deux points par image, c'est à l'utilisateur de faire le tri dans la liste qu'il récupère. De même, la liste des temps correspondant doit être générée par l'utilisateur en fonction des caractéristiques de l'appareil de prise de vue (et du saut programmé dans **CVJShowVideoFrames**)

---

## Relevé de groupes de points dans une vidéo

La fonction **CVJGetVideoFramesPoints** permet de relever plusieurs points par image et de récupérer la liste des points. A la différence de la méthode précédente, les points sont regroupés par image. En revanche, les marqueurs des points sont effacés à chaque image.

```

In[37]:= edit = CVJImageEditor["toupie", True];
liste = CVJGetVideoFramesPoints[edit, video, 10]

```

Out[38]= {{{{414., 220.}, {449., 228.}, {496., 233.}}, {{418., 238.}, {456., 250.}, {494., 256.}},  
 {{408., 261.}, {454., 275.}, {494., 281.}}, {{403., 279.}, {431., 286.}, {464., 292.}}},  
 {{{0.68538, 0.421423}, {0.763764, 0.404791}, {0.868824, 0.395272}},  
 {{0.694934, 0.381394}, {0.780153, 0.35594}, {0.865163, 0.343875}},  
 {{0.67342, 0.329719}, {0.776561, 0.30008}, {0.866034, 0.288085}},  
 {{0.662889, 0.289376}, {0.725618, 0.274731}, {0.79947, 0.262492}}}

La liste obtenue contient deux éléments : la liste des points en unité de pixels et la liste des points en unité utilisateur. La liste des points en pixel est donc :

```

In[39]:= liste[[1]]

```

Out[39]= {{{{414., 220.}, {449., 228.}, {496., 233.}}, {{418., 238.}, {456., 250.}, {494., 256.}},  
 {{408., 261.}, {454., 275.}, {494., 281.}}, {{403., 279.}, {431., 286.}, {464., 292.}}}

Sur cet exemple, il y a 3 points relevés sur chaque image, et 4 images traitées.

---

## Fermeture de l'interface

```
In[40]:= CVC1ose []
```

Il est conseillé de quitter le Kernel avant d'exécuter à nouveau le notebook (Menu Evaluation->Quit Kernel->Local).