



## **Travaux pratiques de vision industrielle**

<b>TP 1 : Acquisition et traitement d'images .....</b>	<b>3</b>
<b>TP 2 : Analyse d'images, dénombrement et classification d'objets.....</b>	<b>8</b>
<b>TP 3 : Contrôle dimensionnel.....</b>	<b>11</b>
<b>Annexe TP 2 : Commandes de BasicScript.....</b>	<b>12</b>

Guillaume LAURENT



# TP 1 : Acquisition et traitement d'images

Cette séance de travaux pratiques vise à illustrer, dans un premier temps, la formation des images numériques puis, dans un second temps, à découvrir quelques fonctions de traitement d'images.

## Partie 1 : Présentation du système de vision

Le poste de travail complet comporte :

- un éclairage diascopique et un éclairage épiscopique,
- un objectif de 25 mm muni,
- une caméra matricielle numérique IEEE1394,
- un PC muni d'une carte d'acquisition FireWire IEEE1394,
- l'environnement de traitement et d'analyse d'images *Aphelion*.

Le processus pratique d'acquisition d'images sur Aphelion comprend dans l'ordre :

- la visualisation en temps réel de l'image (Image Acquisition->Grab/Snap->ImgGrab),
- le réglage des différents paramètres de formation d'images : ouverture du diaphragme, mise au point,
- l'arrêt de l'acquisition (fermeture de la fenêtre image).

*1.1.a. Acquérir l'image d'une scène (contenant des objets divers) avec un réglage permettant d'obtenir le maximum d'informations (zones claires, grises et noires) en utilisant l'éclairage épiscopique.*

## Partie 2 : Acquisition des images

Au niveau de la formation des images, les problèmes d'éclairage, d'optique et de numérisation seront mis en évidence.

### 2.1. Eclairage épiscopique

On définit la profondeur de champ par la plage (en mm) dans laquelle les objets sont nets. Les objets situés en dehors de la profondeur de champ seront flous sur l'image.

Allumer l'éclairage épiscopique et placer une feuille opaque sur la plaque de verre.

*2.1.a. Trouver une procédure pour mesurer expérimentalement la profondeur de champ. Mesurer la profondeur de champ avec et sans éclairage. Qu'observe-t-on et pourquoi ?*

*2.1.b. Enoncer un autre avantage d'un éclairage puissant dans le contexte de la vision industrielle.*

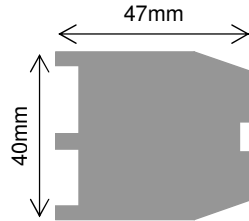
### 2.2. Eclairage diascopique

Choisir une pièce épaisse (>5mm) percée d'un alésage.

*2.2.a. Comparer les éclairages épiscopique et diascopique. Quels sont les avantages et les inconvénients des deux types d'éclairage ?*

### 2.3. Encombrement

Dans une société de fabrication de matériels électriques, on souhaite contrôler les dimensions et la présence ou non de bavures sur les boîtiers de connecteurs qui sortent du moulage sur un tapis roulant.



2.3.a. Le responsable qualité veut un contrôle par vision au 1/100 de millimètre, est-ce possible avec une caméra standard (640x480 pixels) ? Que peut-on lui proposer au mieux ?

2.3.b. Calculer l'encombrement du poste de vision pour des focales de 50mm et 25mm. Quelle solution faut-il privilégier ? Vérifier expérimentalement.

Rappel : l'encombrement correspond à la distance entre l'objet et l'objectif plus le tirage (distance entre l'objectif et le capteur CCD).

La notice d'une caméra standard donne les informations suivantes sur les dimensions du capteur CCD :

Chip size 6.33x4.75mm

### 2.4. Rehaussement du contraste

#### Notion d'histogramme

2.4.a. Ouvrir l'image test.tif et calculer son histogramme (Analysis->Global->ImgHistogramme). Interprétation ?

#### Extension de dynamique (recadrage)

2.4.b. Ouvrir l'image aquitain.tif et calculer son histogramme (Analysis->Global->ImgHistogramme). Quel est le problème ?

2.4.c. Corriger ce problème par une extension de dynamique (Image Utilities->LUT/Point->ImgLinearScale).

#### Egalisation d'histogramme

Cette transformation est construite de telle façon que le nouvel histogramme soit plat (distribution uniforme des niveaux de gris); cette opération est équivalente à maximiser l'entropie de l'image, on obtient alors théoriquement une image ayant une information maximale.

2.4.d. Tester l'égalisation d'histogramme sur l'image galaxy.tif (Image Utilities->LUT/Point->ImgEqualizeHistogram). Conclusion ?

## Partie 3 : Traitement d'images

Le traitement d'images regroupe toutes les techniques visant à transformer une image en vue de faciliter son analyse. La donnée de départ est une image et le résultat est également une image.

### 3.1. Détection des contours

3.1.a. Retrouver le noyau de convolution permettant de calculer l'image test1.tif à partir de test.tif.

Rappel : la convolution 2D d'une image I avec un noyau (ou filtre) N est donnée par :

$$\forall x, \forall y, (I \otimes N)(x, y) = IN(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} I(x-i, y-j)N(i, j)$$

Pour tester votre noyau, utiliser la fonction Filtering->Convolution->ImgConvolve.

Aide : commencer par visualiser le profil d'une même pièce dans les deux images (Analysis->Interactive->ImgProfil).

3.1.b. Retrouver le noyau de convolution permettant de calculer l'image test2.tif à partir de test.tif.

3.1.c. Retrouver le noyau de convolution permettant de calculer l'image test3.tif à partir de test.tif. Quel est son nom usuel ?

Cette convolution est un filtre parmi d'autres pour détecter les contours des objets.

3.1.d. Tester les résultats de quelques autres filtres classiques de détection de contours : sobel, prewitt, passage par zéro (dans Edge detection->Linear).

### 3.2. Renforcement des contours

3.2.a. En utilisant l'image obtenue par la convolution 3.2.c et l'image test.tif, trouver un moyen pour renforcer les contours de l'image de départ. En quoi cela améliore-t-il l'image ? Peut-on effectuer ce traitement à l'aide d'une seule convolution ?

Remarque : si l'image paraît sombre, corriger ce défaut en utilisant un recadrage d'histogramme (avec les valeurs 0-255 et 0-255 pour tronquer les niveaux de gris <0 et >255).

### 3.3. Réduction du bruit

On peut considérer une image comme une juxtaposition de plusieurs zones homogènes représentant les objets. Dans la réalité, des fluctuations des niveaux de gris sont présentes à cause du bruit. On cherche donc à diminuer l'amplitude de ces perturbations, sans toucher aux zones de transition.

#### Méthodes linéaires

L'idée est de réaliser une moyenne (pondérée ou non) des niveaux de gris dans le voisinage d'un pixel.

3.3.a. Trouver un noyau de convolution permettant de lisser (moyenner) une image. Le tester sur l'image bruitée mercury.tif. Le résultat est-il satisfaisant ? Que se passe-t-il si on effectue ce traitement plusieurs fois de suite ?

### Méthodes non linéaires

Le filtre médian calcule la valeur médiane des valeurs de niveaux de gris entourant un pixel.

3.3.b. Tester ce filtre sur mercury.tif (Filtering->Enhancement->ImgMedian5x5) et expliquer pourquoi il est bien adapté aux bruits de type « neige ».

Le filtre de Nagao calcule la variance des niveaux de gris sur un échantillon de masques différents autour du pixel. Le pixel reçoit alors la valeur moyenne des niveaux de gris du masque dont la variance est la plus faible. Par exemple :

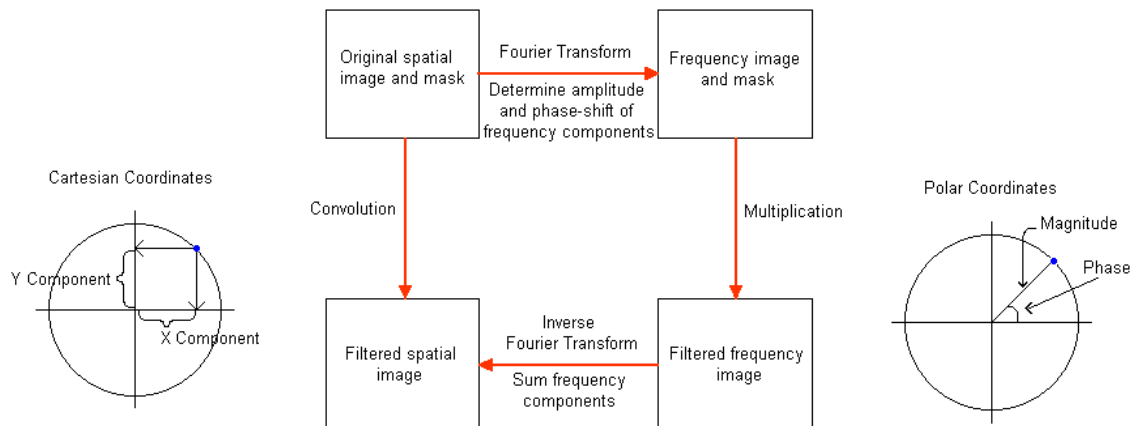
Image data					Average	Variance	Average	Variance			
10	12	40	16	19	10		7.1	104.89		11.1	388.89
14	22	52	10	55	41		9.2	227.32		11.3	360.02
10	14	51	21	14	10		9.55	314.20		10.3	292.01
32	22	9	9	19	14		11.2	553.56		7.7	109.37
41	18	9	22	27	11						
10	7	8	8	4	5						

New pixel value = 7.1

3.3.c. Comparer le filtre médian et le filtre de Nagao sur les images mercury.tif et squares.tif. Quel est l'intérêt du filtre de Nagao ?

### Méthodes de Fourier

La transformation de Fourier constitue une alternative pour faire du filtrage sur une image. Le processus est le suivant : on calcule la transformée de Fourier de l'image sur laquelle on effectue le filtrage puis on calcule la transformée inverse de cette image.



3.3.d. A l'aide de cette méthode, corriger le défaut des images noise.tif et clown.tif (utiliser les commandes de menu Frequency Domain->Fourier).

## **Partie 4 : Localisation par corrélation de pixels**

La corrélation peut être employée pour localiser ou contrôler des objets. La corrélation la plus simple correspond à la convolution d'une image avec un noyau décrivant exactement l'objet recherché (modèle). Les endroits où le résultat est élevé sont les endroits où l'image est semblable au modèle. On utilisera dans notre cas la convolution normalisée qui est indépendante de la luminosité.

On propose de tester cette méthode sur l'exemple classique d'alignement automatique de wafers. Ouvrir les images wafer.tif et mire.tif. Exécuter la fonction Analysis->Correlation->ImgToTemplate sur l'image mire.tif. Effectuer ensuite une corrélation normalisée sur l'image wafer.tif avec le modèle (template) nouvellement créé (Analysis->Correlation->ImgCorrelateWithTemplate).

*4.1.a. Que resterait-il à faire pour localiser la mire ? Quel est le principal défaut de cette méthode ?*

## TP 2 : Analyse d'images, dénombrement et classification d'objets

L'objectif de cette séance est d'étudier le processus complet d'analyse d'images (seuillage, segmentation, mesures, etc.). La première partie est guidée pas à pas et traite des méthodes de seuillage (binarisation). La seconde partie propose de travailler sur des exemples concrets d'analyse d'images. Ces exemples seront l'occasion d'apprendre à écrire des applications automatiques de vision (macros) à l'aide du langage BasicScript d'Aphelion (cf. annexe).

### Partie 1 : Seuillage

Le seuillage est une méthode permettant de ne garder que les pixels satisfaisant à un critère donné. Le seuillage produit une image binaire : 1 pour les pixels satisfaisant au critère, 0 sinon.

#### 1.1. Seuillage manuel

*1.1.a. Disposer des pièces variées sous l'éclairage épiscopique et afficher le profil d'une des pièces à l'aide de la fonction Analysis->Interactive->ImgProfil. Proposer une méthode pour choisir un seuil permettant de garder les pixels appartenant à une pièce donnée.*

*1.1.b. Tester le seuillage manuel sur cette image (Segmentation->Threshold->ImgThreshold).*

#### 1.2. Seuillage global

Il existe de très nombreuses méthodes de seuillage automatique basées sur une mesure globale ou l'optimisation d'un critère global.

*1.2.a. Tester à titre d'exemple le seuillage ImgMomentThreshold. Quel est son principe ?*


#### 1.3. Seuillage à hystérésis

L'idée est d'utiliser deux critères, l'un plus strict que l'autre. Les pixels répondant au premier critère sont automatiquement gardés, les pixels répondant seulement au second sont gardés s'ils sont connexes avec des pixels du premier groupe.

*1.3.a. Sur l'image test.tif, essayer d'utiliser un seuillage manuel pour ne conserver que les deux grandes pièces et les grains noirs. Que se passe-t-il ? Tester ensuite un seuillage avec hystérésis. Proposer de bonnes valeurs.*

## Partie 2 : Analyse d'images binaires

### 2.1. Processus d'analyse

A partir d'une image binaire, il s'agit de caractériser chaque région de pixels connexes par des mesures géométriques ou autres. Pour cela, on convertit l'image binaire en régions : c'est la segmentation (fonction Segmentation->Utilities->ImgClustersObj, pour afficher le tableau des régions cliquer sur le bouton ). On calcule ensuite des caractéristiques géométriques des régions (par exemple la forme avec Analysis->Objects->RegionShape),



2.1.a. Tester le processus ci-dessus sur l'image test.tif seuillée manuellement avec les valeurs 0-83.

Remarque : pour visualiser les régions sur l'image de départ, utiliser la fonction Object Utilities->Graphics->ObjDraw.

## 2.2. Dénombrement d'objets séparés

Disposer quelques dizaines de lentilles ou autres grains (sans qu'ils se touchent) sur l'éclairage diascopique. L'objectif est de déterminer le nombre d'objets présents à l'aide du processus décrit ci-dessus (les objets qui « touchent » le bord de l'image ne seront pas comptabilisés)

2.2.a. Ecrire une macro permettant de calculer le nombre d'objets visible par la caméra. Tester avec différentes configurations de grains.

Remarques :

- pour acquérir une seule image caméra dans une macro utiliser la fonction Image acquisition->Grab/Snap->ImgSnap,
- pour les régions qui touchent le bord, utiliser la fonction Mathematical Morphology->Geodesy->ImgBorderKill.
- pour récupérer le nombre d'objets présents dans une liste d'objets consulter l'exemple donné en annexe.

## 2.3. Dénombrement d'objets mélangés

Disposer quelques objets de plus grande taille en plus des grains (sans qu'ils se touchent) sur l'éclairage diascopique. L'objectif est de déterminer le nombre d'objets de grande taille.

2.3.a. Quelle opération de morphologie mathématique (menu Mathematical Morphology) faut-il ajouter dans le processus d'analyse pour éliminer les grains de l'image binaire ? Que faut-il faire pour conserver la dimension des objets ? Comment s'appelle cette double opération. Ecrire une macro affichant le nombre d'objets de grande taille. Tester avec différentes configurations.

2.3.b. Modifier la macro en utilisant une reconstruction (Mathematical Morphology->Geodesy->ImgReconstruct) plutôt que l'opération morphologique précédente. Quel est l'intérêt de cette méthode ?

## 2.4. Classification de pièces mécaniques

Sur un banc de montage robotisé, on souhaite trier des pièces mécaniques avec un robot (pick-and-place). Le travail préalable est de savoir reconnaître les types de pièces.

Disposer sur l'éclairage diascopique deux catégories de pièces mécaniques ayant une surface visible similaire mais des formes différentes.

2.4.a. Ecrire une macro permettant d'afficher le nombre d'objets de chaque type. Tester avec différentes configurations.

Remarque : on peut trier les régions selon un ou plusieurs critères grâce à la fonction Object Utilities->Cut/Copy/Paste->ObjFilter.

## Partie 3 : Autres opérations de morphologie mathématique

### 3.1. Détection de points caractéristiques

A l'aide de la méthode de transformation de voisinage « tout-ou-rien », il est possible de conserver les pixels d'une image binaire qui satisfont à une condition sur leur voisinage. Typiquement, la fonction tout-ou-rien permet d'extraire les pixels en limite de région, les pixels terminant une ligne, les pixels à l'intersection de deux lignes (points triples) ou les pixels solitaires.

*3.1.a. Tester cette fonction (Mathematical Morphology->Thinning/Thickening->ImgHitOrMiss) sur une image binaire pour extraire les contours des régions. Retrouver le voisinage qui produit ce résultat.*

### 3.2. Amincissement/épaississement

L'amincissement est une opération qui permet de retirer les points appartenant aux contours des régions et qui satisfont à une condition sur leur voisinage.

*3.2.a. Disposer un ruban métallique découpé sur l'éclairage diascopique et tester la fonction d'amincissement (Mathematical Morphology->Thinning/Thickening->ImgThin) sur l'image binaire obtenue. Quel est l'intérêt de l'amincissement par rapport à l'érosion. A quoi servent les différents voisinages proposés ?*

*3.2.b. Tester la fonction d'épaississement (Mathematical Morphology->Thinning/Thickening->ImgThicken) sur l'image binaire initiale. Que constatez-vous ?*

### 3.3. Squelettisation

Les opérations de squelettisation permettent de réduire à une « ligne » la forme d'une région d'une image binaire tout en conservant ses caractéristiques topologiques. On peut notamment obtenir le squelette d'une région par amincissements itératifs.

*3.3.a. Disposer des pièces de formes variées (trouées et non trouées) sur l'éclairage diascopique et tester différentes méthodes de squelettisation (Mathematical Morphology->Skeletons) sur l'image binaire obtenue.*

### 3.4. Séparation de régions

Il arrive couramment que deux régions représentant deux objets distincts soient connexes ce qui pose problème lors de l'analyse. Une méthode classique permettant de séparer ces régions est d'utiliser l'érosion et la squelettisation.

*3.4.a. Disposer des objets de forme **convexe** sur l'éclairage diascopique de telle manière que certains se touchent (des lentilles par exemple). Proposer un processus permettant de séparer les régions en utilisant l'érosion et la squelettisation.*

Une autre méthode consiste à utiliser les lignes de partages des eaux.

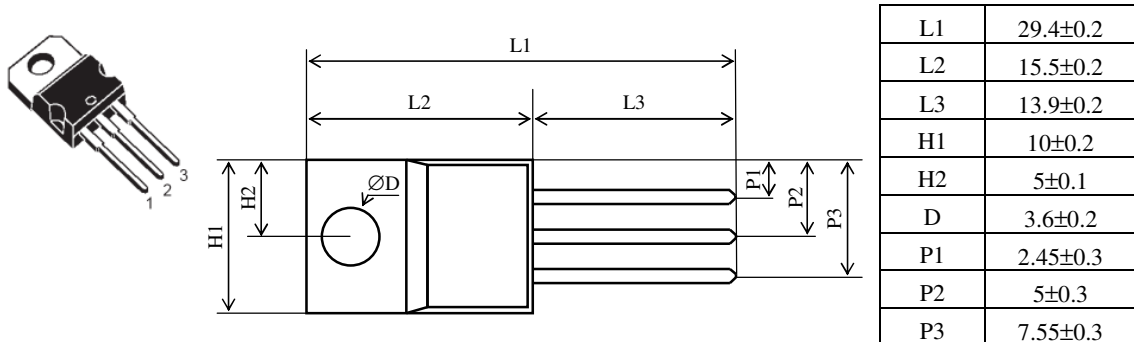
*3.4.b. Proposer un processus de séparation fondé sur les fonctions : Mathematical Morphology->Distance->ImgGraphDistance et Mathematical Morphology->Watershed->ImgWatershed.*

Ce processus étant fastidieux, Aphelion propose une fonction englobant les différentes étapes.

*3.4.c. Tester la fonction Segmentation->Morphology->AphImgClustersSplitConvex. Que se passe-t-il si on effectue la séparation sur des régions concaves ?*

## TP 3 : Contrôle dimensionnel

On souhaite inspecter la conformité géométrique des composants électroniques présentés ci-dessous.



L'objectif est mettre au point une macro Aphélon permettant d'inspecter séparément chaque composant. Les cotes à vérifier sont indiquées en millimètres dans le tableau ci-dessus.

### 1.1. Etalonnage

Pour obtenir des mesures en vraies grandeurs (millimètres) et non en pixels, il est nécessaire d'étalonner le système. Pour ce faire, on utilise l'image d'un objet étalon ou mire dont on connaît les cotes avec précision.

Il est souhaitable de prendre une mire dont les cotes à tester sont parallèles aux axes x et y. Une fois le système étalonné, on peut acquérir les images de travail à la condition de ne pas modifier les conditions de prise de vues (mise au point, etc.).

1.1.a. Ecrire une macro permettant de calculer les coefficients de calibration en x et y à l'aide du composant étalon.

1.1.b. Quelle est la précision maximale que l'on peut atteindre ?

### 1.2. Contrôle dimensionnel

1.2.a. Ecrire une macro pour réaliser le contrôle des cotes L1, H1, H2 et D d'un composant. La macro devra afficher les cotes et un message indiquant si le composant est bon ou mauvais.

1.2.b. Compléter la macro pour réaliser le contrôle des autres côtes d'un composant. La macro devra afficher les cotes et un message indiquant si le composant est bon ou mauvais.

1.2.c. Utiliser cette macro pour tester chaque composant et présenter les résultats sous forme d'un tableau dans le compte-rendu de TP.

### 1.3. Conclusion

1.3.a. Dans le cadre d'une chaîne de production, quels seraient les points à améliorer ?

## Annexe TP 2 : Commandes de BasicScript

### Nettoyage de la mémoire d'Aphelion (à mettre en début de macro)

```
AphImgFreeAll  
AphObjFreeAll
```

### Messages

```
MsgBox "Nombre d'objets_1:" & nb1 & chr(13) + chr(10) &  
"Nombre d'objets_2:" & nb2 & chr(13) + chr(10)
```

### Remarques :

- & est l'opérateur de concaténation,
- chr(13)+chr(10) permet de passer à la ligne suivante à l'écran,
- \_ autorise le retour à la ligne au sein d'une instruction

### Ouverture de fichiers images

```
AphImgRead AphImgNew ("Riz"), "C:\Chemin Complet\image.tif"
```

### Déclaration d'un tableau d'entiers

```
Dim Index() As long
```

### Calcul du nombre d'objets dans une liste de région (nommée GRAINES)

```
dim index() as long  
AphObjGetIndexList AphObj("GRAINES"), index  
lo = LBound(index)  
hi = UBound(index)  
nb = hi-lo+1  
MsgBox "Nombre de graines : " & nb
```

### Lecture d'une mesure dans une liste de région

Lecture de la mesure "REGION.EXTENTS.LL.X" de la 3<sup>ème</sup> région :

```
dim index() as long  
AphObjGetIndexList AphObj("GRAINES"), index  
mesure = AphObjGetAttributed( AphObj("GRAINES"), index(2),  
"REGION.EXTENTS.LL.X")
```

### Structure Si-Alors-Sinon

```
If (condition) Then  
Traitement1  
End If
```

```
If (condition1) and (condition2) Then  
Traitement1  
Else  
Traitement2  
End If
```

### Structure Tant Que

```
While (condition)
  Traitement
Wend
```

### Structure Pour

```
For i=1 To N
  Traitement
Next i
```

### Ecriture dans un fichier texte

```
open "c:\ensmm\test.txt" for output as #1
write #1, "TP de vision"
write #1, 23
write #1,
N=2.65
write #1, N
close #1
```

Produit le fichier suivant :

```
"TP de vision"
23
2.65
```

### Formatage de l'affichage d'un réel

```
MsgBox "N = " & Format(2.65397, "0.00")
```

Produit un message « N=2.65 »