

---

# Poursuite de cible.

---

## 1 Distance / corrélation.

Si on a deux images de même taille, il est possible de mesurer à quel point les deux images se ressemblent via une mesure de distance ou une mesure de corrélation.



Image 1.



Image 2.

Plusieurs outils sont à notre disposition pour réaliser cette comparaison dont on peut se servir pour suivre un motif dans une séquence d'images. C'est ce qu'on appelle la poursuite de cible.

### 1.1 Distances.

Une distance est une mesure positive et symétrique qui passe par un minimum quand les deux arguments sont identiques et qui respecte l'inégalité triangulaire.

Soient  $I_1$  et  $I_2$  deux images à comparer et  $D$  une distance :

- $D(I_1, I_1) = 0$ ,
- $D(I_1, I_2) = D(I_2, I_1)$ ,
- $D(I_1, I_2) \geq 0$ ,
- $D(I_1, I_3) \leq D(I_1, I_2) + D(I_2, I_3)$ .

En imagerie, on utilise souvent les distance absolues et quadratiques (dite euclidienne), et leurs variantes centrées et normalisées (voir cours).

$$\text{Absolue : } D_A(I_1, I_2) = \frac{1}{N} \sum_1 \sum_c |I_1(l,c) - I_2(l,c)|,$$

$$\text{Quadratique : } D_Q(I_1, I_2) = \frac{1}{N} \sum_1 \sum_c |I_1(l,c) - I_2(l,c)|^2,$$

où  $N$  est le nombre de pixel de chaque image

Plus une distance est petite, plus elle indique une ressemblance entre deux images.

### 1.2 Corrélation

Le calcul de la corrélation ressemble beaucoup à celui d'une distance, à la différence près que cette mesure augmente avec la similarité de deux images et passe par un maximum qui est l'autocorrélation d'une image (sa corrélation avec elle-même).

La corrélation de deux images  $I_1$  et  $I_2$  s'écrit :

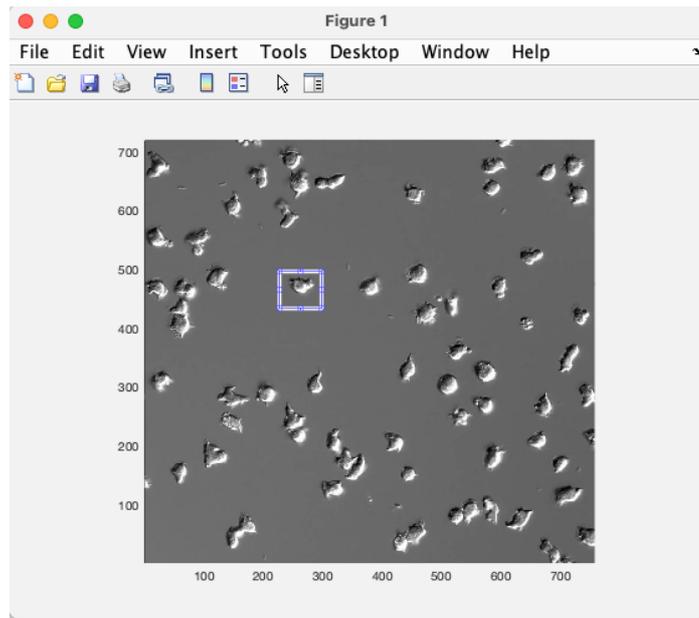
$$\text{Cor}(I_1, I_2) = \frac{1}{N} \sum_1 \sum_c I_1(l,c) \cdot I_2(l,c)$$

## 2 Poursuite de cible.

### 2.1 Principe.

La poursuite de cible, dans cette séance, va consister à implanter un algorithme de recherche locale d'un optimum (minimum pour une distance, maximum pour une corrélation) entre une sous-image de référence, prise au début de la séquence, et une sous-image extraite de l'image courante ayant la même dimension.

Sur le plan pratique, vous disposez d'une séquence de 60 images représentant des micro-organismes en mouvement. L'idée sera de sélectionner une des cellules et d'essayer de la suivre pendant son trajet dans la séquence d'images.



Sélection d'une cellule dans la première image.

On vous conseille naturellement de commencer par une cellule isolée pour assurer que l'algorithme reste sur la même cellule.

### 2.2 Méthode.

Le principe est le suivant :

1. on caractérise la position d'un motif (sous-image) par ses coordonnées dans l'image courante. Cela peut être le coin en haut à gauche, le centre de la sous-image, les limites en ligne et colonne de l'image, etc.
  2. dans l'image suivante on recherche, dans le voisinage de la position courante, la sous-image de même taille dont la ressemblance avec le motif initial est maximale (distance minimale, corrélation maximale),
  3. on remet à jour la position courante du motif en prenant la position du motif de ressemblance maximale,
  4. on revient à l'étape 2 en considérant l'image suivante,
- et cela bien sûr jusqu'à épuisement de la liste d'images.

Une fois que vous aurez programmé une méthode, essayez de varier les critères de ressemblance (distance, corrélation, autre ...) ainsi que le voisinage de recherche.

Dans votre programmation, attention aux bords de l'image ...

### 2.3 Aide à la programmation.

Comme d'habitude, on vous donne ci-dessous un début de canevas pour votre programme.

```

NomDossier = 'SequenceCells' ; % le dossier dans lequel sont les images
ok = 1 ; % indicateur permettant de savoir si on a épuisé la liste des images
n = 0 ;
voisinage = 10 ; % voisinage d'exploration de la position du motif dans l'image
courante
TransPoss = -voisinage:voisinage ; % valeurs possible de la translation du motif
while ok
    % creation du nom de l'image courante
    if n<10
        Nom = sprintf('%s/Cells0%d.jpg',NomDossier,n) ;
    else
        Nom = sprintf('%s/Cells%d.jpg',NomDossier,n) ;
    end

    FID = fopen(Nom) ; % pour vérifier que l'image existe
    if(FID>0)
        fclose(FID) ;
        MonImage = imread(Nom) ; % si oui on lit l'image courante
        [Nlin, Ncol, Nplan] = size(MonImage) ; % on recupere sa dimension
        figure(1) ; clf ; hold on ; image(MonImage) ; colormap(gray(256)) ; axis
image ;
        MonImage = double(MonImage) ;
        if n<1 % si c'est la premiere image on selectionne le motif a poursuivre
            [Icropped, rectangle] = imcrop ;
            % affichage du motif
            figure(10) ; image(Icropped) ; colormap(gray(256)) ; axis image ;
            % recuperation du rectangle selectionne
            col_min = floor(rectangle(1)) ;
            col_max = ceil(rectangle(1)+rectangle(3)) ;
            lin_min = floor(rectangle(2)) ;
            lin_max = ceil(rectangle(2)+rectangle(4)) ;
            % selection du motif de reference
            ImageRef = double(MonImage(lin_min:lin_max,col_min:col_max)) ;
            figure(17) ; image(uint8(ImageRef)) ; colormap(gray(256)) ;
        end

        Critere = zeros(length(TransPoss)) ;
        % creation d'un tableau de visualisation de l'evolution du critere
        u_opt = 1 ; v_opt = 1 ; Cr_opt = 0 ; % attention a la valeur par default du
critere

        for u=1:length(TransPoss)
            lin_min_cur = lin_min + TransPoss(u) ;
            lin_max_cur = lin_max + TransPoss(u) ;
            for v=1:length(TransPoss)
                col_min_cur = col_min + TransPoss(v) ;
                col_max_cur = col_max + TransPoss(v) ;
                ImageCur =
MonImage(lin_min_cur:lin_max_cur,col_min_cur:col_max_cur) ;

                % calcul du critere et recherche de la valeur optimale
                % Critere(u,v) = ;
            end
        end

        % visualisation du critere
        figure(22) ; surf(TransPoss,TransPoss, Critere) ;
        shading interp ; drawnow ;
        % Mise a jour de la position du motif dans l'image courante
        ImageCur = MonImage(lin_min:lin_max,col_min:col_max) ;
        figure(18) ; image(uint8(ImageCur)) ; colormap(gray(256)) ;

    else ok=0 ; % si on ne peut pas lire l'image on sort du programme
    end
    n = n+1;
end
end

```

## 2.4 Pour aller plus loin...

Comme vous pouvez le constater, le motif à poursuivre change de forme et d'orientation. Vous pouvez réfléchir (on l'a vu en cours) à comment prendre en compte cette variation d'orientation. Vous pouvez aussi utiliser une distance qui n'est pas sensible à l'orientation, comme par exemple une distance entre les histogrammes des deux motifs à comparer.

Nous vous conseillons d'essayer la distance de Kolmogorov (ou sa variante dite de Cramér-von Mises), qui est très simple à calculer, ou encore l'information mutuelle (un peu plus difficile).

Pour la distance de Kolmogorov, on bâtit les histogrammes cumulés des deux sous-images :

```
Histo_cur = hist(ImageCur(:),0:255) ;  
Gamma_cur = Histo_cur ;  
for i=2:255 Gamma_cur(i) = Gamma_cur(i-1) + Histo_cur(i) ; end
```

La distance de Kolmogorov entre deux histogrammes est la valeur maximale de l'écart absolu entre les deux histogrammes cumulés.

La distance de Cramér-von Mises entre deux histogrammes est la distance quadratique les deux histogrammes cumulés.

Pour l'information mutuelle, on doit bâtir trois histogrammes : l'histogramme de chaque sous-image et l'histogramme mutuel (qui est un histogramme a deux dimensions).

L'histogramme mutuel se calcule ainsi :

```
HistoMutuel = zeros(256) ;  
for i=1:256  
    for j=1:256  
        Indicateur = ( ImageCur == (i-1) ) .* ( ImageRef == (j-1) ) ;  
        HistoMutuel(i,j) = sum(Indicateur(:)) ;  
    end  
end
```

L'information mutuelle est alors :

$$IM(I_1, I_2) = \sum_{i=0}^{255} \sum_{j=0}^{255} H_{12}(i, j) \cdot \log \left( \frac{H_{12}(i, j)}{H_1(i) \cdot H_2(j)} \right)$$

où  $H_{12}$  est l'histogramme mutuel et  $H_1$  et  $H_2$  les histogrammes des deux sous-images à comparer.

Attention !!! si  $H_{12}(i,j)$  est nul il ne faut pas évaluer  $\log(0)$  mais partir sur le fait que  $0 \cdot \log(0) = 0$ .

L'algorithme obtenu est très lent, vous pouvez l'accélérer en calculant vos histogrammes sur des ensemble de valeurs plus larges (par exemple grouper les niveaux de gris par 8).