

SESSION 2017

AGREGATION CONCOURS EXTERNE

Section : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR

Option : SCIENCES INDUSTRIELLES DE L'INGÉNIEUR
ET INGÉNIERIE INFORMATIQUE

MODÉLISATION D'UN SYSTÈME, D'UN PROCÉDÉ
OU D'UNE ORGANISATION

Durée : 6 heures

Calculatrice électronique de poche - y compris calculatrice programmable, alphanumérique ou à écran graphique – à fonctionnement autonome, non imprimante, autorisée conformément à la circulaire n° 99-186 du 16 novembre 1999.

L'usage de tout ouvrage de référence, de tout dictionnaire et de tout autre matériel électronique est rigoureusement interdit.

Dans le cas où un(e) candidat(e) repère ce qui lui semble être une erreur d'énoncé, il (elle) le signale très lisiblement sur sa copie, propose la correction et poursuit l'épreuve en conséquence.

De même, si cela vous conduit à formuler une ou plusieurs hypothèses, il vous est demandé de la (ou les) mentionner explicitement.

NB : *La copie que vous rendrez ne devra, conformément au principe d'anonymat, comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé comporte notamment la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de signer ou de l'identifier.*

Ce sujet comporte :

- la présentation et le travail demandé pages 1 à 22 ;
- le document annexe pages 23 à 24.

INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie.

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

Concours	Section/option	Epreuve	Matière
EAE	1417A	109	2680

Vidéo-capsule endoscopique



Image d'une vidéo-capsule endoscopique

Le cancer colorectal représente la deuxième cause de mortalité dans le monde. Dans 80% des cas, il se développe à partir d'un adénome (appelé aussi polype). Un adénome est une excroissance de la muqueuse intestinale et il est le plus souvent de forme circulaire comme le montrent les photographies de la figure 1.

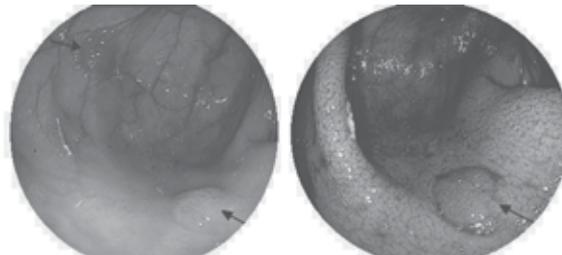


Figure 1 – Photographies d'adénomes

Le risque de transformation d'un adénome en cancer varie en fonction de sa taille et de la composition de ses cellules. Le dépistage au plus tôt et l'amélioration des traitements ont permis de réduire le taux de mortalité, mais il reste encore trop important. Le dépistage précoce de ce cancer peut se faire soit par des tests biologiques et/ou par un examen d'imagerie endoscopique (coloscopie, célioscopie ...). Ces examens sont pratiqués à l'aide de vidéo-endoscopes filaires ou depuis peu par vidéo-capsules endoscopiques. Les praticiens ont recours à ces dernières dans le cas de patients à risques (contre-indications à l'anesthésie ...) ou lorsque l'examen porte sur des portions de l'appareil digestif qui sont inaccessibles par le biais d'instruments classiques.

On donne sur la figure 2 le diagramme des cas d'utilisation d'une vidéo-capsule endoscopique précisant le service rendu ainsi que les acteurs principaux et secondaires.

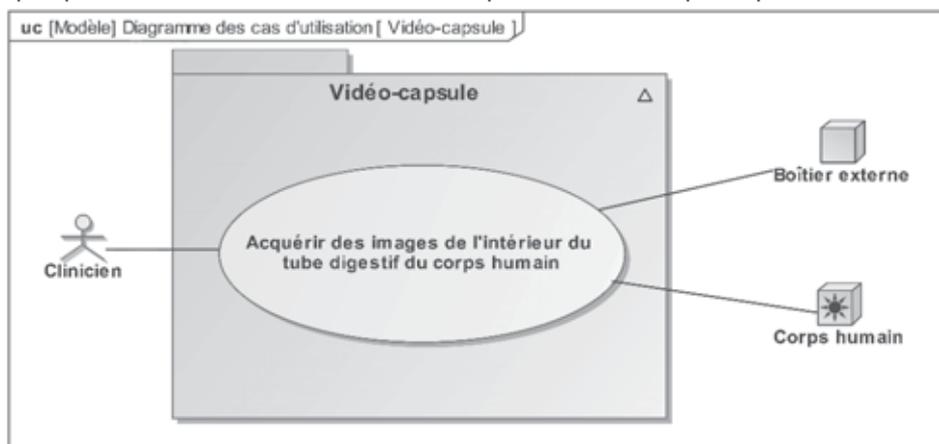


Figure 2 – Diagramme des cas d'utilisation d'une vidéo-capsule endoscopique

Une vidéo-capsule est un dispositif autonome et sans fil qui, une fois ingérée, permet de prendre des images à intervalles réguliers de l'appareil digestif depuis l'œsophage jusqu'au rectum. Leur taille est d'environ 25 mm de longueur et de 10 mm de diamètre. Il existe actuellement trois types de vidéo-capsules (optimisées pour l'œsophage, l'intestin grêle et le gros colon). Aucune ne permet un enregistrement continu de tout l'appareil digestif.

L'examen est réalisé en ambulatoire, le patient ingère la vidéo-capsule et le praticien récupère sous 24 h 00 le boîtier externe d'enregistrement (figure 3). L'analyse par le clinicien des images acquises permet de déceler l'origine des saignements et la présence éventuelle d'adénomes.

Les images acquises par le capteur d'images sont émises à travers le corps humain puis sont récupérées par un boîtier externe (*datalogger*) via des électrodes posées sur le corps du patient. Les images enregistrées dans le boîtier externe sont ensuite traitées « *off-line* » par le clinicien.

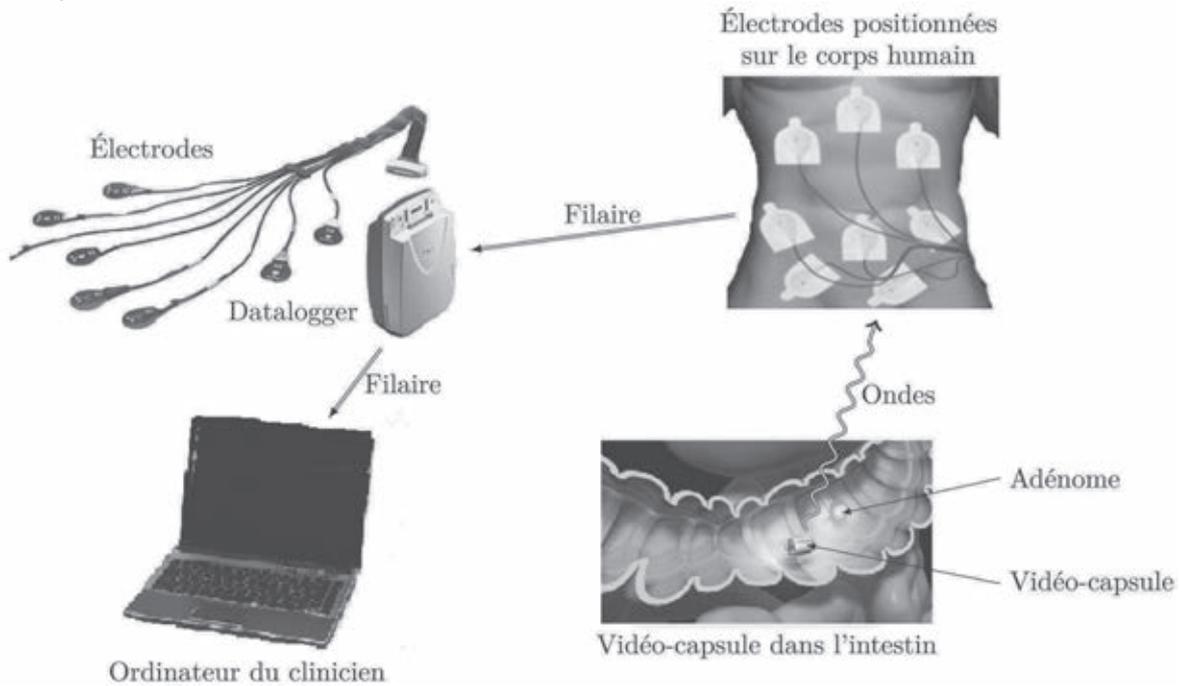


Figure 3 – Synoptique du système complet

Les vidéo-capsules possèdent plusieurs limitations du fait de leur architecture :

- le capteur d'images est de faible résolution. De ce fait, les images délivrées permettent d'obtenir une estimation des tailles des lésions avec 50% d'erreur ;
- l'autonomie est limitée à 12 h 00 ;
- le temps de traitement est important et les outils actuels ne permettent pas d'interpréter le caractère cancéreux des adénomes.

Par ailleurs, les images acquises restent de qualité réduite (résolution, conditions d'éclairage mal-maîtrisées, mouvements sporadiques ...) et la mise en place de méthodes d'analyse reste une problématique ouverte afin d'éviter une majeure partie du post-traitement des données par le clinicien.

Une vidéo-capsule comporte principalement quatre blocs distincts (figure 4) :

- un bloc optique composé des LEDs, des lentilles et du dôme optique permettant l'acquisition des images de la zone du tube digestif ;
- une unité de traitement ;
- une unité de transmission radiofréquence (amplificateur RF) associée à une antenne ;
- une alimentation par piles-bouton.

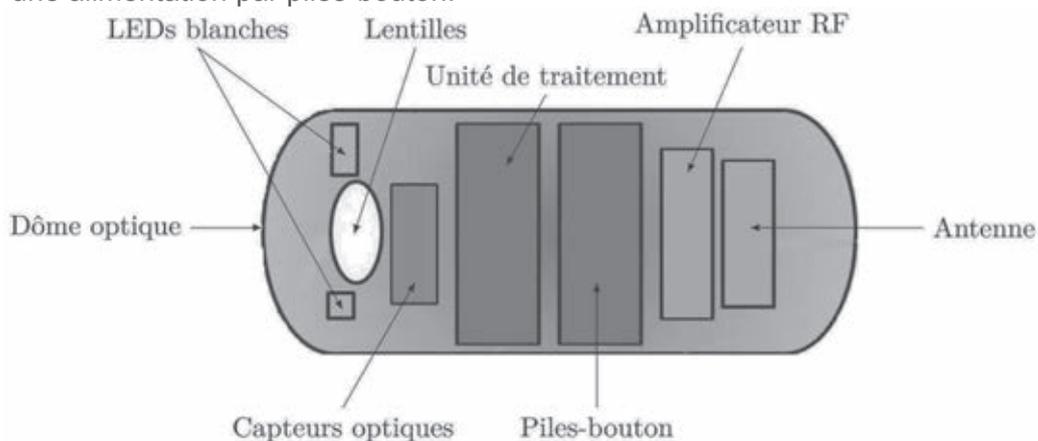


Figure 4 – Descriptif interne de la vidéo-capsule

L'objet de cette étude porte sur la modélisation de la vidéo-capsule afin d'évaluer la pertinence d'embarquer les traitements en vue d'augmenter l'autonomie du dispositif.

Le tableau 1 décrit la qualification des exigences relatives à la vidéo-capsule endoscopique.

Id	Exigence	Critère(s)	Niveau(x)
1.1	Éclairer la zone du tube digestif	Puissance lumineuse	10 mW
1.2	Capturer les images	Colorimétrie	RVB
		Résolution	320 × 280 pixels
1.3	Transmettre les images	Quantité	Toutes les images prises
		Stockage	Aucun en interne
		Débit	500 kbit/s
		Connectique	Sans fil
1.4	Être autonome en énergie	Autonomie	12 h 00
1.5	Stocker les données dans un boîtier externe	Capacité maximale	100 Go
1.6	Mettre à disposition les données	Connectique	Filaire

Tableau 1 – Caractérisation des exigences

La partie 1 du sujet aborde la validation des performances globales de la vidéo-capsule, en termes de capacité de stockage et d'autonomie énergétique. La partie 2 porte sur le traitement des images permettant d'extraire des zones pathologiques d'intérêt, potentiellement des adénomes, appelées ROI (*Region Of Interest*), et d'analyser les performances d'un classifieur. La partie 3 aborde l'étude de la transmission des images via une modulation numérique du type modulation de fréquence. Avant de conclure sur l'intérêt d'embarquer les traitements dans la vidéo-capsule (partie 5), la partie 4 s'intéresse à la modélisation partielle de la structure d'acquisition des images par l'instrumentation optique.

Partie 1 – Étude globale des performances

Objectif : valider les performances globales de la vidéo-capsule décrites dans le cahier des charges.

En France, le nombre d'examen d'endoscopie s'élève à 1 150 000 par an. Un tiers correspond à des examens de colonoscopie (examen du colon). Le coût d'une colonoscopie est d'environ 765€ tandis que le coût d'un examen par vidéo-capsule est de l'ordre de 500€. Tous ces examens d'imagerie ne sont pas dédiés exclusivement au dépistage du cancer du côlon.

La figure 5 décrit le système digestif humain depuis la bouche jusqu'au rectum. Les trois principales parties du tube digestif sont l'estomac, l'intestin grêle et le gros intestin (colon). La durée de transit dans l'estomac est de l'ordre de 4 h 00 pour un sujet sain et les vitesses péristaltiques (propagation de l'onde digestive) sont respectivement de $3 \text{ cm}\cdot\text{s}^{-1}$ dans l'œsophage, et $2 \text{ cm}\cdot\text{min}^{-1}$ dans l'intestin grêle et le gros intestin.

La vidéo-capsule a une fréquence d'acquisition qui dépend du segment intestinal. Elle est :

- de 35 images/seconde dans l'œsophage ;
- de 2 images/seconde dans l'estomac ;

- de 15 images/seconde dans l'intestin grêle et le gros intestin.

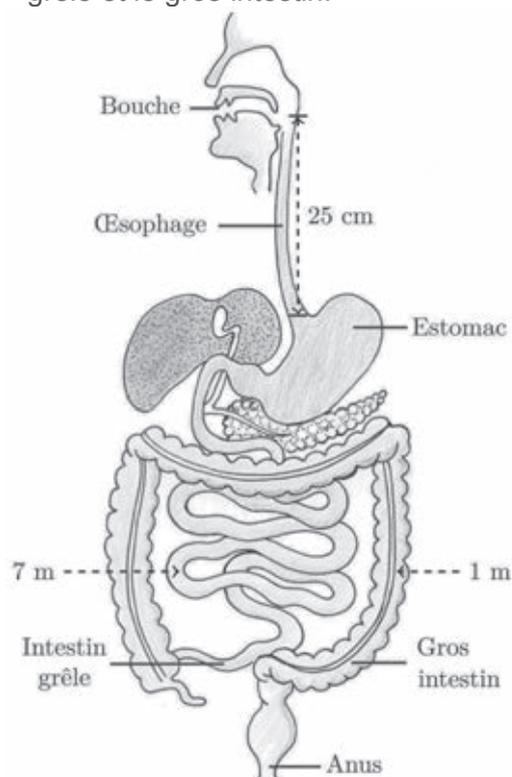


Figure 5 – Synoptique du transit intestinal

Tournez la page S.V.P.

La résolution des images couleurs prises par la vidéo-capsule est de 320 par 280 pixels. Le modèle de colorimétrie utilisé est le RVB (Rouge – Vert – Bleu), et la profondeur de couleur est de 24 bpp (*bit per pixel*), soit 8 bits pour chaque couleur du modèle de colorimétrie.

Q1. Évaluer la durée totale $T_{transit}$ du transit entre l'ingestion de la vidéo-capsule jusqu'à son évacuation dans les selles, et valider l'exigence Id 1.4.

Q2. Évaluer le nombre d'images N_{tot} prises par la vidéo-capsule lors d'un transit complet de l'œsophage à l'anus.

Q3. Évaluer la taille de la mémoire à implanter dans le boîtier externe (*datalogger*) si aucune compression d'images est mise en œuvre, et valider l'exigence Id 1.5.

Dès lors que les performances globales de la vidéo-capsule sont validées, il est nécessaire d'analyser le traitement déporté des images (partie 2), d'étudier la transmission de celles-ci (partie 3), ainsi que de modéliser la chaîne d'acquisition des images dans la partie 4. La partie 5 revient sur la problématique d'embarquer le traitement des images dans la vidéo-capsule.

Partie 2 – Traitement déporté des images

Objectif : étudier la chaîne algorithmique de traitement des images permettant d'identifier une zone présentant potentiellement un adénome et analyser les performances d'un classifieur.

Le cancer colorectal se développe dans 80% des cas à partir d'un adénome (figures 1 et 6). Le risque de transformation d'un adénome en cancer varie en fonction de sa taille et de son architecture tissulaire (nature villositaire, degré de dysplasie). Ainsi, on considère qu'un adénome est « avancé » (donc à risque élevé de dégénérescence en cancer) lorsque sa taille dépasse 10 mm, lorsque qu'un contingent villositaire occupe plus de 25% de sa surface et/ou lorsque qu'il contient une zone de dysplasie de haut grade. Le dépistage des adénomes à un stade précoce (précancéreux) et l'amélioration des traitements ont permis de réduire le taux de mortalité. Un adénome est le plus souvent de forme circulaire comme le montrent les photographies de la figure 6.

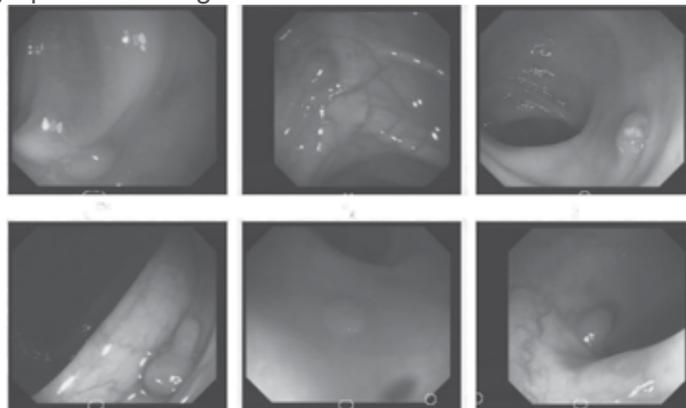


Figure 6 – Exemples d'adénomes à détecter

La détection des adénomes revient au problème de la recherche d'une forme circulaire dans une image 2D. La figure 7 montre le processus de la chaîne de traitement. L'identification d'une zone de type cercle dans une image visible nécessite plusieurs traitements :

- une binarisation de l'image couleur ;
- une transformation de Hough pour obtenir les zones d'intérêt (ROI – *Region of Interest*) (figure 8) ;
- le calcul de la matrice de cooccurrence sur les zones d'intérêt ;
- l'extraction des paramètres de textures pour chaque ROI ;
- une opération de classification pour déterminer s'il s'agit potentiellement d'un adénome cancéreux ou non.

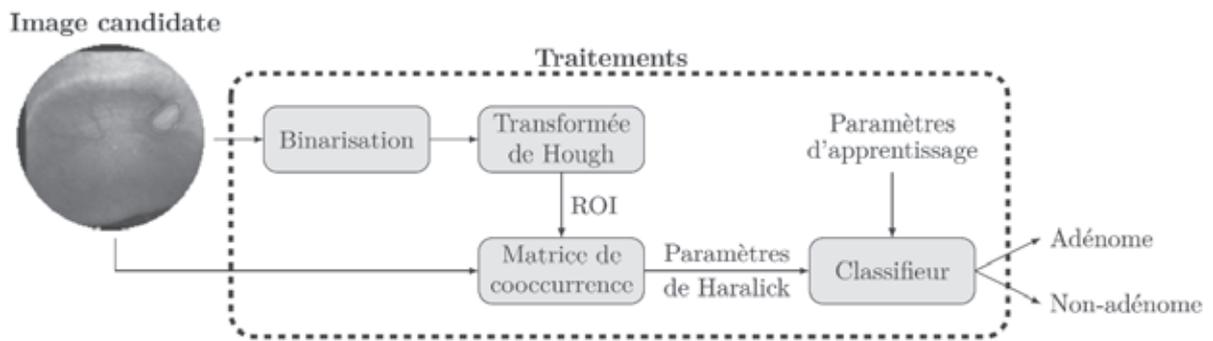


Figure 7 – Synoptique du traitement des images

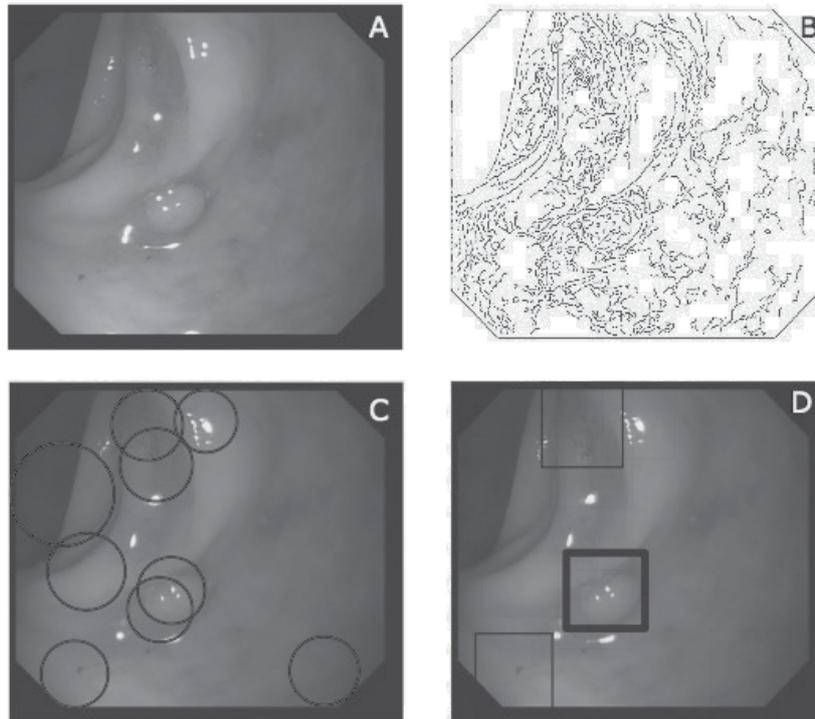


Figure 8 – A, image candidate ; B, binarisation et détection des contours ; C, zones d'intérêt ; D, adénome/non adénome

Avant d'envisager d'extraire les zones d'intérêt (ROI) dans l'image, un premier prétraitement est nécessaire. Il permet de vérifier que l'image est exploitable, c'est-à-dire, peu ou pas de présence de matières résiduelles (aliments en cours de digestion).

Conformément à la figure 9, il n'est pas nécessaire de réaliser les traitements sur toutes les images prises par la vidéo-capsule. Seules celles où la proportion de matières résiduelles n'est « pas trop importante » devront subir le traitement de la figure 7.

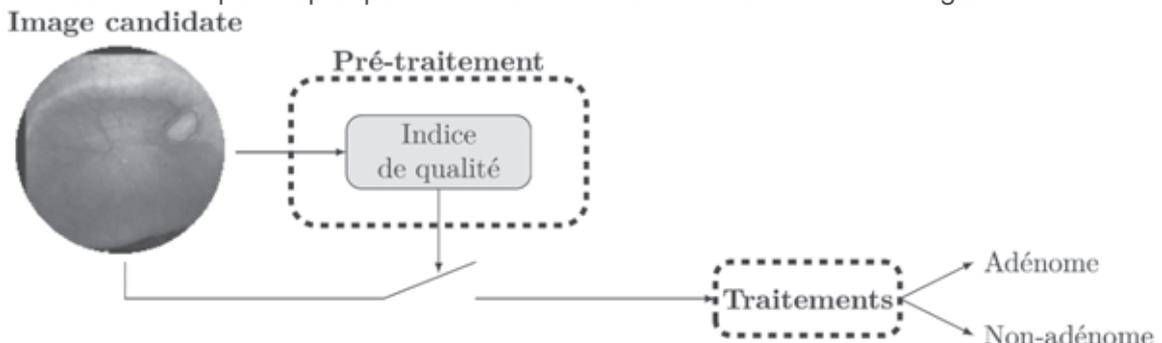


Figure 9 – Synoptique de la chaîne de prétraitement

Un article scientifique¹, propose de quantifier cette proportion de matières résiduelles (représentative de la qualité d'une image) par le coefficient *CAC* (*Computed Assessment of Cleansing*) défini par :

$$CAC = \frac{\text{moy}(R)}{\text{moy}(V)}$$

où *R* et *V* correspondent aux tableaux de rouge et de vert de l'image couleur.

La décision de traiter ou non une image se résume donc simplement à un test tel que :

$\left\{ \begin{array}{ll} \text{Si } CAC > 1,65 & \text{alors image à traiter ;} \\ \text{Sinon} & \text{image à ne pas traiter.} \end{array} \right.$

On donne ci-dessous un code partiel en langage Python de la fonction **calc_CAC** permettant de calculer l'indice de qualité *CAC* d'une image candidate (dont les canaux rouge, vert et bleu sont passés en argument sous la forme de tableaux – variables **R**, **V** et **B** de type *array*) et qui décide si l'image doit être traitée ou non (renvoie **True** ou **False**).

```

1 def calc_CAC(R,V,B):
2     """ Cette fonction calcule le coefficient CAC = moy(R)/moy(V) à partir des 3
3     paramètres R, V et B, et renvoie True si CAC > 1.65, sinon False.
4     Les arguments R, V et B sont des tableaux. """
5     # Détermination des dimensions du tableau R (ou V)
6     nblignes,nbcolonnes=R.shape # .....
7     # .....
8     sumR,sumV=0,0 # .....
9     for ii in range(nblignes):
10        for jj in range(nbcolonnes):
11            sumR=sumR+R[ii][jj]
12            # .....
13        moyR=sumR/(nblignes*nbcolonnes) # .....
14        moyV=sumV/(nblignes*nbcolonnes) # .....
15        CAC=moyR/moyV # .....
16        # .....
17        if CAC > 1.65:
18            return(.....)
19        else:
20            return(.....)

```

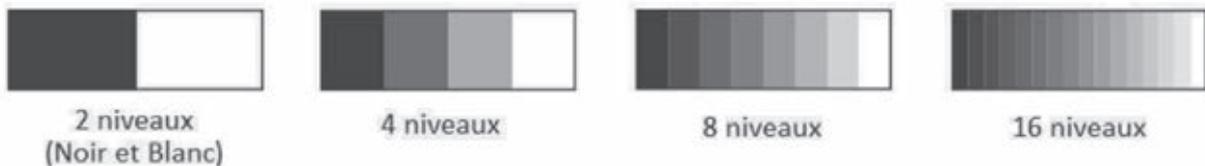
Q4. Compléter les zones manquantes (lignes 5 à 7, 11 à 15, 17 et 19) de la fonction **calc_CAC** afin de satisfaire à l'objectif décrit ci-dessus.

La binarisation d'une image couleur consiste à générer une image de même dimension qu'une image couleur, mais où chaque pixel de l'image binarisée est soit blanc, soit noir. On associe à un pixel de couleur blanche la valeur 1 et 0 pour un pixel noir. Cette binarisation peut se décomposer en trois étapes :

- Étape 1 – détermination d'une image en niveaux de gris (variable **imGray** de dimensions **nblignes*nbcolonnes**) où chacun de ses pixels correspond à une combinaison linéaire des niveaux de couleurs (rouge, vert et bleu) de l'image couleur. Les valeurs des niveaux de gris sont comprises entre 0 et 1 sous la forme d'un flottant ;
- Étape 2 – détermination d'un seuil optimal (variable **seuil**) pour le passage d'une image en niveaux de gris vers une image noir et blanc. De nombreuses méthodes de détermination d'un seuil optimal existent. Dans ce sujet, la méthode d'OTSU basée sur le seuillage de l'histogramme sera mise en œuvre. Celle-ci est décrite en Annexe 1 ;
- Étape 3 – détermination de l'image en noir et blanc (variable **imNB**) par comparaison de chaque pixel de l'image **imGray** au seuil optimal.

¹ Immediate Duodenal Delivery versus Delayed Oral Ingestion of Capsule Endoscopy : is the Sooner the Better ? A Prospective Comparison of Small Bowel Visualization Quality, According to a Computed Assessment of Cleansing Score – Rafaële Gerometta, Gabriel Rahmi, Stéphane Leclaire, Aymeric Histace, Eanes Abou Ali, Olivier Romain, Philippe R. Marteau et Xavier Dray.

La figure 10 présente la notion de niveaux de gris d'une image.



On donne ci-dessous le code de la fonction **imcolor2Gray** permettant de générer un tableau (variable **imGray**) correspondant à l'image en niveaux de gris (étape 1) à partir du tableau correspondant à l'image couleur (variable **imColor**).

```
1 def imcolor2gray(imColor):
2     """ Cette fonction permet de générer un tableau en 2 dimensions d'une image en
3     niveaux de gris à partir d'une image couleur en 3 dimensions.
4     imColor[:, :, 0] correspond au tableau de rouge, imColor[:, :, 1] correspond au
5     tableau de vert et imColor[:, :, 2] correspond au tableau de bleu. """
6     # Détermination des dimensions de l'image couleur
7     taille = imColor.shape
8     # Initialisation d'une variable imGray de zéros (mêmes dimensions que imColor =
9     2D)
10    imGray = np.zeros([taille[0], taille[1]])
11    # Remplissage de la variable imGray
12    for ii in range(0, taille[0]-1):
13        for jj in range(0, taille[1]-1):
14            imGray[ii][jj] = 0.2125*imColor[ii][jj][0] + 0.7154*imColor[ii][jj][1]
15            + 0.0721*imColor[ii][jj][2]
16    return(imGray)
```

Q5. En analysant le code fourni ci-dessus, déterminer la combinaison linéaire utilisée pour générer les pixels en niveaux de gris. Justifier la pondération forte sur la couleur verte.

Q6. En Python, écrire une fonction **seuillageOTSU** permettant de renvoyer la variable **seuil** correspondant au seuil optimal, basée sur le seuillage de l'histogramme (sur 256 niveaux de gris car codage sur un octet) à partir de la méthode d'OTSU (cf. article en annexe 1). La fonction **seuillageOTSU** prend en argument le tableau à 2 dimensions correspondant à l'image en niveaux de gris (variable **imGray**).

Q7. En Python, écrire une fonction **imcolor2NB** permettant de renvoyer une image binarisée (variable **imNB**) où les pixels noirs sont de valeur 0 et les pixels blancs de valeur 1, basée sur le seuillage de l'histogramme à partir de la méthode d'OTSU. Cette fonction prend en argument un tableau en 3 dimensions correspondant à l'image couleur (variable **imColor**).

Afin de minimiser les traitements des images par l'unité de traitement, la recherche de zones d'intérêts (ROI) n'est réalisée que sur les pixels blancs de l'image obtenue après binarisation et détection de contours (non étudié dans ce sujet).

Q8. En Python, écrire une fonction **indexNB** permettant de renvoyer une liste des pixels blancs (de valeur 1) contenus une image en Noir et Blanc passée en argument sous la forme d'un tableau (à 2 dimensions). La liste renvoyée sera composée de sous-listes composées chacune par les coordonnées *i* et *j* d'un pixel blanc.

La transformée de Hough circulaire est une transformation de l'image des contours dans un espace à 3 dimensions (*a*, *b* et *R*), où *a* et *b* représentent l'abscisse et l'ordonnée du centre du cercle recherché et *R* son rayon. Dans notre cas d'étude, la transformation se limite à un espace à 2 dimensions (figure 11), car le rayon des cercles recherchés est de 10 mm.

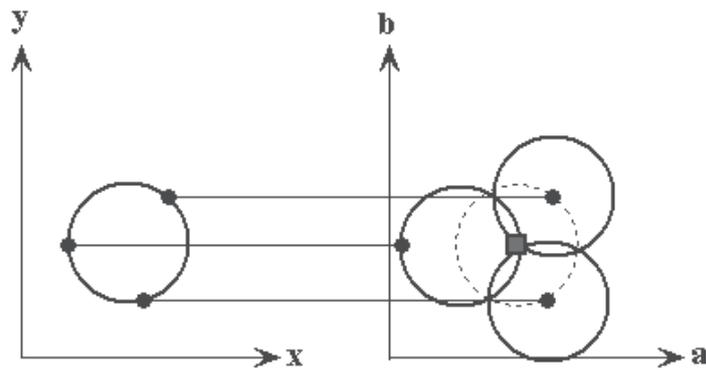


Figure 11 – Transformation vers les paramètres de Hough (a, b)

Q9. Donnez l'équation d'un cercle de centre d'abscisse a , d'ordonnée b et de rayon R , dans le plan (x, y) .

L'algorithme de la Transformée de Hough peut être décrit par le pseudo-code suivant :

```

1 Pour un rayon  $R$  donné - faire
2   Pour tout pixel  $(x, y)$  - faire
3     Tracer le cercle passant par  $x, y$ , de rayon  $R$  dans une matrice appelée
    accumulateur
4     Mettre à jour accumulateur
5   Fin tout pixel
6   Seuiller accumulateur et trouver les maximas (centre probable du cercle de
    rayon  $R$  recherché)
7 Fin tout rayon

```

Q10. Écrire en Python une fonction **Hough**, qui, pour un rayon R donné et prenant pour argument la liste des pixels de contours (pixels blancs), permet de renvoyer le centre probable du cercle d'abscisse a et d'ordonnée b .

La figure 12 présente deux zones d'intérêt où des adénomes (cercles de rayon de 10 mm) ont été décelés par la transformée de Hough et la figure 13 donne les niveaux de gris associés à ces deux zones d'intérêt sur 16 niveaux de gris (Ndg).

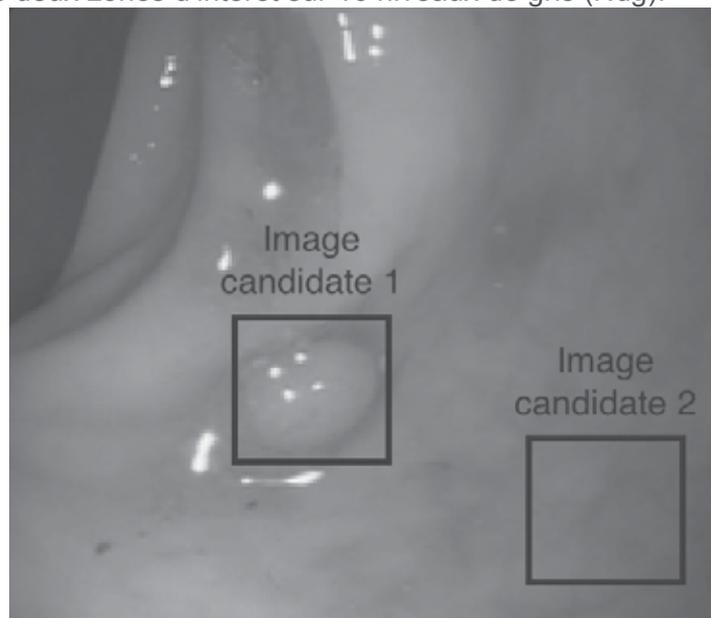


Figure 12 – Photographie prise par une vidéo-capsule endoscopique présentant deux zones candidates 1 et 2

9	9	9	8	8	8	9	9	9	9	8	8
9	9	9	9	9	9	9	9	8	7	7	8
10	10	10	9	9	9	8	8	7	7	7	7
10	10	9	9	8	9	9	10	9	9	8	7
10	9	9	8	8	10	11	10	10	9	8	7
9	9	8	8	8	9	11	10	10	9	8	7
8	8	8	8	8	9	10	9	9	8	7	8
8	8	13	10	7	8	8	8	8	7	8	8
8	9	13	9	7	8	8	9	8	8	9	8
8	8	9	8	9	8	9	12	10	8	9	8
8	8	8	8	8	8	8	9	9	9	9	8
9	9	9	9	9	9	9	9	9	9	9	9

0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0
0	1	2	3	4	5	5	4	3	2	1	0

a) Niveaux de gris de l'image candidate 1 b) Niveaux de gris de l'image candidate 2
 Figure 13 – Niveaux de gris associés aux 2 images candidates en 16 Ndg

Une matrice de cooccurrence d'une image en niveaux de gris caractérise la périodicité et la directivité des textures et représente le nombre de transition d'un niveau de gris à un autre sous forme matricielle.

La matrice de cooccurrence, introduite dans [Haralick, 1973], est caractérisée par le nombre de niveaux de gris (Ndg) d'une image, ce dernier lui donne sa taille (Ndg×Ndg). Une image à 32 niveaux de gris donne une matrice à 1 024 éléments (32×32). Ceci nous permet d'anticiper la problématique au niveau des ressources qui vont être nécessaires afin de calculer et traiter les matrices de cooccurrence. Une matrice de cooccurrence est aussi caractérisée par une distance d et une orientation θ . La distance d détermine le nombre de cases entre deux éléments et l'orientation θ détermine le sens. La figure 14 illustre le calcul de la matrice de cooccurrence pour une image en 8 niveaux de gris avec une distance $d = 1$ et une orientation $\theta = 0^\circ$ (de la gauche vers la droite, pour chaque ligne).

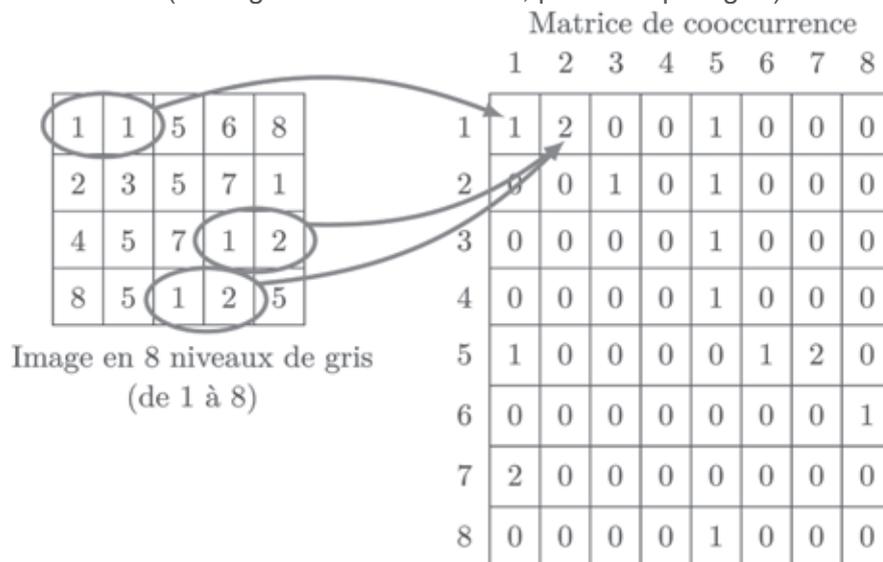


Figure 14 – Exemple de calcul de la matrice de cooccurrence pour $d = 1$ et $\theta = 0^\circ$

Q11. Pour l'image en 8 Ndg de la figure 14, donner la matrice de cooccurrence pour une distance $d = 1$ et une orientation de $\theta = 90^\circ$ (de haut en bas pour chaque colonne).

On donne ci-dessous les 2 matrices de cooccurrence des images candidates 1 et 2 de la figure 12 pour $d = 1$ et $\theta = 0^\circ$ avec 16 niveaux de gris compris entre 0 et 15.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	4	5	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	7	25	12	1	0	0	1	0	0
9	0	0	0	0	0	0	0	1	17	34	2	1	1	1	0	0
10	0	0	0	0	0	0	0	1	1	7	5	1	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Matrice de cooccurrence de l'image candidate 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	12	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	12	0	12	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	12	0	12	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	12	0	12	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	12	12	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Matrice de cooccurrence de l'image candidate 2

Q12. En Python, écrire une fonction `mat_cooccurrence` permettant de renvoyer la matrice de cooccurrence d'une image en niveaux de gris passée en argument (sous la forme d'un tableau) avec **Ndg** niveaux de gris, passé aussi en argument. Cette fonction se limitera au cas où $d = 1$, et $\theta = 0^\circ$ ou $\theta = 90^\circ$. L'angle sera passé en argument et vous utiliserez à bon escient la méthode `transpose` accessible pour toute variable de type tableau (`array`).

L'article de Haralick publié en 1973 sur l'analyse de la texture, donne la définition de 28 paramètres calculables à partir d'une matrice de cooccurrence, permettant ainsi de caractériser la texture d'une image en niveaux de gris par une liste de 28 éléments. Dans le cadre de ce sujet, nous utiliserons uniquement 3 paramètres :

- le moment angulaire d'ordre 2 appelé ASM. Ce paramètre mesure l'homogénéité de l'image. Plus cette valeur est faible, moins l'image est uniforme et dans ce cas, l'image présente de fortes transitions de couleurs ;
- le contraste. Ce paramètre informe sur les variations locales des couleurs. Si ces variations sont importantes, alors le contraste sera élevé. Ce paramètre permet aussi

de caractériser la dispersion des valeurs de la matrice de cooccurrence par rapport à sa diagonale principale ;

- la corrélation. Elle mesure la dépendance linéaire des niveaux de gris. Elle permet ainsi de caractériser le contenu de l'image.

Ces 3 paramètres sont définis mathématiquement par :

$$ASM = \sum_{i=1}^{Ndg} \sum_{j=1}^{Ndg} \left(\frac{P(i,j)}{R} \right)^2$$

$$contraste = \sum_{n=0}^{Ndg-1} n^2 \cdot \left(\sum_{\substack{i=1 \\ |i-j|=n}}^{Ndg} \sum_{j=1}^{Ndg} \left(\frac{P(i,j)}{R} \right) \right)$$

$$corr\u00e9lation = \frac{\sum_{i=1}^{Ndg} \sum_{j=1}^{Ndg} (i - \mu_x) \cdot (j - \mu_y) \cdot \frac{P(i,j)}{R}}{\sigma_x \cdot \sigma_y}$$

où $P(i,j)$ est l'élément situé à la $i^{\text{ème}}$ ligne et à la $j^{\text{ème}}$ colonne de la matrice de cooccurrence P , R le nombre d'éléments dans la matrice de cooccurrence, μ_x , μ_y , σ_x et σ_y sont respectivement les moyennes et écart-types définis par :

$$\mu_x = \sum_{i=1}^{Ndg} \sum_{j=1}^{Ndg} i \cdot \frac{P(i,j)}{R}$$

$$\mu_y = \sum_{i=1}^{Ndg} \sum_{j=1}^{Ndg} j \cdot \frac{P(i,j)}{R}$$

$$\sigma_x = \sqrt{\sum_{i=1}^{Ndg} \sum_{j=1}^{Ndg} (i - \mu_x)^2 \cdot \frac{P(i,j)}{R}}$$

$$\sigma_y = \sqrt{\sum_{i=1}^{Ndg} \sum_{j=1}^{Ndg} (j - \mu_y)^2 \cdot \frac{P(i,j)}{R}}$$

Q13. En Python, écrire une fonction **calcul_param_Haralick** permettant de renvoyer les 3 paramètres de Haralick (**ASM**, **contraste** et **corr\u00e9lation**) d'une matrice de cooccurrence passée en argument. Le r\u00e9sultat de cette fonction correspondra \u00e0 une liste de 3 \u00e9l\u00e9ments. Une d\u00e9composition en sous-fonctions est fortement conseill\u00e9e tout en pr\u00e9cisant les arguments de chacune d'elle.

On s'int\u00e9resse maintenant \u00e0 l'optimisation des calculs de ces trois param\u00e8tres. La figure 15 d\u00e9crit l'ex\u00e9cution s\u00e9quentielle du calcul des param\u00e8tres de Haralick sur un monoprocesseur. Les temps de calcul de chaque param\u00e8tre pour des matrices de cooccurrence de dimension 16\u00d716 sont $T_{ASM} = 1\,020 \mu s$, $T_{CONTRASTE} = 2\,938 \mu s$ et $T_{CORRELATION} = 4\,807 \mu s$.

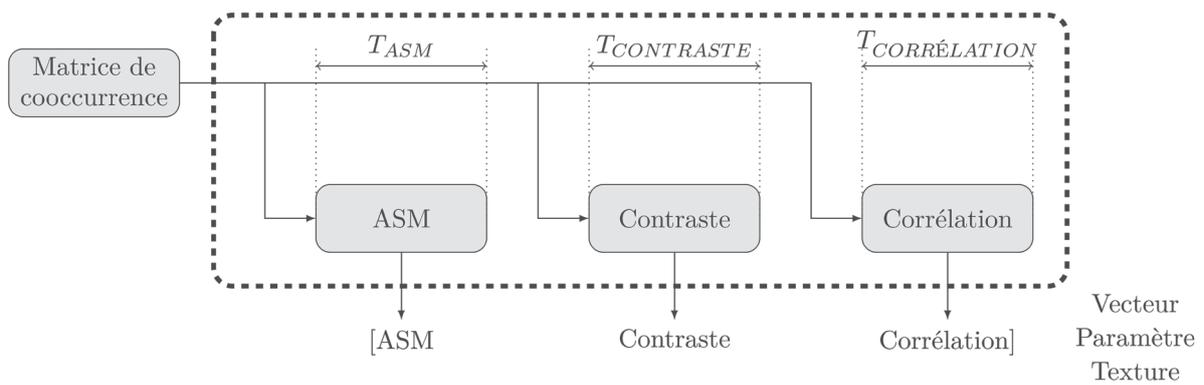


Figure 15 – Calcul de trois des paramètres de Haralick sur un monoprocesseur qui exécute le programme `calcul_param_Haralick`

Q14. Calculer le temps de traitement concernant uniquement le calcul des trois paramètres dans la configuration séquentielle. Les contraintes d’acquisition définies dans la partie 1 sont-elles respectées ? Argumenter votre réponse. Proposer une solution, sous la forme d’un diagramme d’états, permettant d’optimiser la durée de calculs de ces 3 paramètres et d’obtenir un temps de traitement égal au maximum de $(T_{ASM}, T_{CONTRASTE}, T_{CORRELATION})$.

On donne dans le tableau 2 les résultats des paramètres de Haralick pour 2 classes, une classe « Adénomes » et une classe « Non-adénomes ».

Adénomes	Zones d'intérêt								
Paramètres	A1	B1	C1	D1	E1	F1	G1	H1	I1
ASM	5,5	5,4	5,3	5,0	5,5	5,1	5,3	5,6	5,6
Contraste	12,7	13,3	12,6	13,3	12,7	12,6	12,9	13,0	12,7
Corrélation	-0,7	-0,2	-1,0	-0,9	-0,2	-0,9	-0,8	-0,6	-0,4

Non-adénomes	Zones d'intérêt								
Paramètres	A2	B2	C2	D2	E2	F2	G2	H2	I2
ASM	0,7	1,4	1,2	0,9	0,9	1,2	1,3	0,7	0,7
Contraste	3,1	4,0	3,5	3,2	3,6	3,3	3,8	3,3	3,9
Corrélation	0,3	-0,1	-0,2	0,6	0,0	0,5	0,3	0,7	0,3

Tableau 2 – Résultats des paramètres de Haralick pour 9 images tests des deux classes

Les classifications « Adénomes » et « Non-adénomes » sont basées sur les valeurs des trois paramètres ASM, contraste et corrélation. Ces trois paramètres sont vus comme des coordonnées dans un espace à 3 dimensions. Tous les points caractérisant ou non un adénome sont donc localisés dans deux zones distinctes de l’espace. L’étude suivante vise à déterminer la localisation d’un point dans l’une ou l’autre des deux zones caractéristiques.

Q15. Écrire en Python une fonction `calc_barycentre` permettant de calculer le barycentre d’un nuage de points passé en argument, sous la forme d’une liste de trois éléments (**ASM**, **contraste**, **corrélation**).

Q16. Après avoir donné les composantes du vecteur \overrightarrow{AB} passant par les deux barycentres (de coordonnées ASM, contraste, corrélation) des deux classes (A et B étant respectivement les barycentres des deux classes), donner l’équation de l’hyperplan Π séparant ces 2 classes.

Q17. En Python, écrire une fonction `test`, qui sur un candidat lambda, décide si la présence d’un adénome est détectée ou non (renvoie **'Adénome'** ou **'Non-adénome'**). Le critère associé est la plus courte distance (norme euclidienne) entre le point K de coordonnées (**ASM**, **contraste**, **corrélation**) du candidat lambda et l’hyperplan Π .

On donne dans le tableau 3 les résultats obtenus sur le test de 30 images. Il donne la classe d'appartenance de l'image et le diagnostic réalisé par le clinicien appelé vérité terrain.

Numéro d'image	Classe détectée	Vérité terrain
Numéro 1	Adénome	Adénome
Numéro 2	Adénome	Non-adénome
Numéro 3	Adénome	Adénome
Numéro 4	Adénome	Adénome
Numéro 5	Adénome	Adénome
Numéro 6	Adénome	Adénome
Numéro 7	Adénome	Adénome
Numéro 8	Adénome	Adénome
Numéro 9	Adénome	Adénome
Numéro 10	Adénome	Adénome
Numéro 11	Adénome	Adénome
Numéro 12	Adénome	Adénome
Numéro 13	Adénome	Non-adénome
Numéro 14	Adénome	Non-adénome
Numéro 15	Adénome	Adénome
Numéro 16	Non-adénome	Non-adénome
Numéro 17	Non-adénome	Non-adénome
Numéro 18	Non-adénome	Non-adénome
Numéro 19	Non-adénome	Non-adénome
Numéro 20	Non-adénome	Adénome
Numéro 21	Non-adénome	Non-adénome
Numéro 22	Non-adénome	Non-adénome
Numéro 23	Non-adénome	Non-adénome
Numéro 24	Non-adénome	Non-adénome
Numéro 25	Non-adénome	Non-adénome
Numéro 26	Non-adénome	Adénome
Numéro 27	Non-adénome	Adénome
Numéro 28	Non-adénome	Adénome
Numéro 29	Non-adénome	Non-adénome
Numéro 30	Non-adénome	Non-adénome

Tableau 3 – Résultats des traitements sur 30 images tests

La matrice de confusion est un outil servant à mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée (classe détectée), tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (vérité terrain). Elle prend la forme du tableau 4.

DéTECTÉ \ Réel	Adénome	Non-adénome	Total
Adénome	a	b	$a + b$
Non-adénome	c	d	$c + d$
Total	$a + c$	$b + d$	$n = a + b + c + d$

Tableau 4 – Forme d'une matrice de confusion

On définit à partir de cette matrice de confusion différents indicateurs de performances du classifieur, notamment :

- la sensibilité S_e (ou taux de vrais positifs - TVP) représente la fraction des positifs intégrés dans la cible. Elle est définie par

$$S_e = \frac{a}{a + b}$$

- la spécificité S_p est la fraction des négatifs qui sont exclus de la cible, définie par

$$S_p = \frac{d}{c + d}$$

Si le système de classification est correct, la matrice de confusion doit tendre vers une matrice diagonale (à l'exception de la ligne et de la colonne Total).

Q18. Donner, à partir des résultats du tableau 3, la matrice de confusion du classifieur mis en œuvre dans ce sujet. Calculer la sensibilité S_e et la spécificité S_p . Conclure sur la pertinence des études menées, et proposer des solutions permettant d'améliorer les performances du traitement des images prises par la vidéo-capsule étudié dans ce sujet.

Partie 3 – Transmission des informations

Objectif : modéliser les différents éléments intervenants dans la transmission des données de la vidéo-capsule vers le boîtier externe (datalogger).

Les données binaires issues des capteurs optiques doivent être transmises aux électrodes placées sur le corps du patient par des ondes électromagnétiques. La vidéo-capsule est équipée d'une antenne qui transmet ces données numériques après modulation. La modulation utilisée est une modulation numérique du type 4-FSK. On se propose dans un premier temps d'étudier les caractéristiques fréquentielles d'un signal modulé par une modulation numérique 4-FSK.

Les données binaires sont rassemblées dans l'alphabet 4-aire $c_k \in [-3/2; -1/2; +1/2; +3/2]$, f_p est la fréquence centrale et on note Δf le pas de fréquence entre 2 fréquences discrètes de la modulation 4-FSK.

La fonction porte dans le domaine temporel de durée d'un symbole T_s , notée $\pi_{T_s}(t)$, est définie par :

$$\pi_{T_s}(t) = \begin{cases} 1 & \text{pour tout } t \in]-T_s/2; T_s/2] \\ 0 & \text{ailleurs.} \end{cases}$$

En première approche, on considère la fréquence du signal modulé stable (constante) pendant un temps symbole T_s . Les signaux FSK peuvent s'écrire comme la superposition de signaux modulés en amplitude autour de la fréquence centrale notée f_p :

- lorsque le mot binaire transmis est égal à $a_k = (00)_2 = 0$, le signal modulé en fréquence correspond à $s_{F0}(t) = A \times \cos\left(2\pi\left(f_p - \frac{3}{2}\Delta f\right)t\right)$;
- lorsque le mot binaire transmis est égal à $a_k = (01)_2 = 1$, le signal modulé en fréquence correspond à $s_{F1}(t) = A \times \cos\left(2\pi\left(f_p - \frac{1}{2}\Delta f\right)t\right)$;
- lorsque le mot binaire transmis est égal à $a_k = (10)_2 = 2$, le signal modulé en fréquence correspond à $s_{F2}(t) = A \times \cos\left(2\pi\left(f_p + \frac{1}{2}\Delta f\right)t\right)$;
- lorsque le mot binaire transmis est égal à $a_k = (11)_2 = 3$, le signal modulé en fréquence correspond à $s_{F3}(t) = A \times \cos\left(2\pi\left(f_p + \frac{3}{2}\Delta f\right)t\right)$.

De part le côté aléatoire des données binaires transmises, on considère l'apparition des ces 4 signaux équiprobable. On note $b_{i,k}(t)$ la variable binaire égale à 1 lorsque le $k^{\text{ème}}$ mot transmis est égal à i , 0 sinon ($i \in [0, 1, 2, 3]$).

Le signal modulé 4-FSK, noté $s_{4-FSK}(t)$, peut donc s'écrire comme la superposition de 4 signaux modulés en amplitude :

$$s_{4-FSK}(t) = \sum_{k=-\infty}^{+\infty} b_{0,k} \times \pi_{T_s}(t - kT_s) \times A \times \cos\left(2\pi\left(f_p - \frac{3}{2}\Delta f\right)t\right) + \dots$$

$$\sum_{k=-\infty}^{+\infty} b_{1,k} \times \pi_{T_s}(t - kT_s) \times A \times \cos\left(2\pi\left(f_p - \frac{1}{2}\Delta f\right)t\right) + \dots$$

$$\sum_{k=-\infty}^{+\infty} b_{2,k} \times \pi_{T_s}(t - kT_s) \times A \times \cos\left(2\pi\left(f_p + \frac{1}{2}\Delta f\right)t\right) + \dots$$

$$\sum_{k=-\infty}^{+\infty} b_{3,k} \times \pi_{T_s}(t - kT_s) \times A \times \cos\left(2\pi\left(f_p + \frac{3}{2}\Delta f\right)t\right)$$

Pour tout signal temporel $x(t)$, on note $\gamma_x(f)$ sa densité spectrale de puissance définie par :

$$\gamma_x(f) = \underline{X}(f) \cdot \underline{X}^*(f) = |\underline{X}(f)|^2 = TF\{C_{xx}(t)\}$$

où $\underline{X}(f) = TF\{x(t)\}$ représente la transformée de Fourier de $x(t)$, $\underline{X}^*(f)$ son complexe-conjugué, $|\underline{X}(f)|$ le module de $\underline{X}(f)$, TF la transformation de Fourier et $C_{xx}(t)$ la fonction d'autocorrélation de $x(t)$. Le produit de convolution entre 2 grandeurs est noté $x * y$, et la distribution de Dirac dans le domaine fréquentiel $\delta(f)$.

1 Détermination de la densité spectrale de puissance de $s_{4-FSK}(t)$

Q19. Donner, pour chacun des 4 signaux de $s_{4-FSK}(t)$, l'expression $m_i(t)$ des modulateurs en amplitude, et l'expression des porteuses, notées $p_i(t)$. Pour chaque porteuse $p_i(t)$, déterminer sa fréquence, notée f_i , et donner la transformée de Fourier de $p_i(t)$, notée $\underline{P}_i(f)$.

Q20. En déduire l'expression du signal $s_{4-FSK}(t)$ en fonction des $m_i(t)$ et $p_i(t)$ et celle de la transformée de Fourier du signal $s_{4-FSK}(t)$, notée $\underline{S}_{4-FSK}(f)$ en fonction de $\underline{M}_i(f)$ et $\underline{P}_i(f)$.

La formule de Bennett dans le cas de signaux aléatoires, stationnaires et ergodiques fournit l'expression de la densité spectrale de puissance du signal $m_i(t)$, notée $\gamma_{m_i}(f)$ où μ_b et σ_b représentent respectivement la valeur moyenne et l'écart-type des réalisations de la variable aléatoire $b_{i,k}$:

$$\gamma_{m_i}(f) = A^2 \left(\frac{\sigma_b^2}{T_s} + \frac{\mu_b^2}{T_s^2} \sum_{n=-\infty}^{+\infty} \delta\left(f - \frac{n}{T_s}\right) \right) \times |\gamma_{\pi_{T_s}}(f)|$$

Q21. Calculer numériquement les valeurs de μ_b et σ_b . Calculer la densité spectrale de puissance $\gamma_{\pi_{T_s}}(f)$ de la fonction porte $\pi_{T_s}(t)$. Simplifier l'écriture de la densité spectrale de puissance des signaux modulateurs $\gamma_{m_i}(f)$ en fonction de $\gamma_{\pi_{T_s}}(f)$ en tirant profit de la parité de la fonction porte $\pi_{T_s}(t)$ et des caractéristiques de $\gamma_{\pi_{T_s}}(f)$. En déduire l'expression de la densité spectrale $\gamma_{m_i}(f)$ de $m_i(t)$, puis celle de $|\underline{M}_i(f)|$.

Pour la suite du sujet, on pose $\underline{M}_i(f) = M \times \text{sinc}(\pi f T_s)$.

Q22. Déterminer l'expression de la densité spectrale de puissance $\gamma_{s_{4-FSK}}(f)$ du signal $s_{4-FSK}(t)$.

On souhaite que le choix des 4 fréquences des porteuses $p_i(t)$ soit tel que les signaux associés à chaque fréquence discrète soient orthogonaux. Cela signifie que pour 2 signaux $s_m(t)$ et $s_n(t)$ avec $m \neq n$, le produit scalaire $\langle s_m(t), s_n(t) \rangle = 0$.

Les signaux $s_m(t)$ et $s_n(t)$ sont définis mathématiquement par :

$$s_m(t) = A \cdot \sqrt{2} \cdot \cos(\omega_0 \cdot t + 2 \cdot \pi \cdot m \cdot f_d \cdot t + \varphi_m) \cdot \pi_{T_s}(t)$$

$$s_n(t) = A \cdot \sqrt{2} \cdot \cos(\omega_0 \cdot t + 2 \cdot \pi \cdot n \cdot f_d \cdot t + \varphi_n) \cdot \pi_{T_s}(t)$$

Q23. Quelle est la relation entre f_d et T_s permettant d'avoir les signaux $s_m(t)$ et $s_n(t)$ orthogonaux ($\forall (m, n)$ et $m \neq n$) ? En déduire la relation devant lier Δf à T_s .

La figure 16 présente la densité spectrale de puissance d'un signal modulé par une modulation numérique 4-FSK dans le cas où les signaux sont orthogonaux.

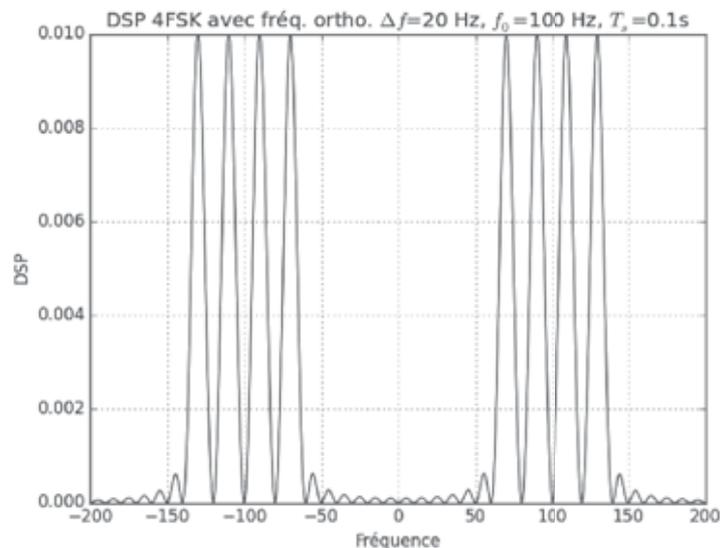


Figure 16 – Allure de la densité spectrale de puissance d'un signal modulé 4-FSK avec signaux orthogonaux

Une analyse rapide amène à la conclusion simple que l'occupation fréquentielle théorique est infinie. Ceci n'est pas souhaitable dans la majorité des cas. On souhaite donc déterminer la bande de fréquences occupée par le signal $s_{4-FSK}(t)$ comprenant 95% de la puissance totale, par une fonction **calc_BF95** écrite en langage Python. Cette fonction prend deux arguments, une liste **DSP** correspondant aux valeurs de la densité spectrale de puissance du signal $s_{4-FSK}(t)$ pour les valeurs de la fréquence contenue dans la liste **freq**.

Q24. Donner les réflexions et raisonnements nécessaires ainsi que l'ensemble des étapes de programmation pour mener à bien cet objectif. **Ne pas procéder à l'écriture de la fonction calc_BF95 ni des sous-fonctions éventuelles.**

2 Regroupement bits – symbole et tracé du signal $s_{4-FSK}(t)$ dans le domaine temporel

La modulation utilisée est une modulation 4-FSK. Cela signifie que les données de l'image sont regroupées par « paquets » de 2 bits (pour obtenir une valence de 4), appelés symboles. Cette opération est appelée *mapping*.

On donne ci-dessous un exemple de code Python permettant de tracer la fonction f définie par $f(x) = 3 \cdot x^2 - 2 \cdot x + 1/(x - 2)$ sur l'intervalle $[-1,1]$.

```

1 import matplotlib.pyplot as plt
2 # Définition de la variable x des abscisses
3 x=[ii/100-1 for ii in range(0,201)]
4 # Définition de la liste y=f(x) des ordonnées
5 y=[3*a**2-2*a+1/(a-2) for a in x]
6 # Tracé de la courbe
7 plt.figure(1) # Création de la figure 1
8 plt.plot(x,y,'r') # Description du tracé de y=f(x) en rouge (r)
9 plt.grid(True) # Mise en place de la grille
10 plt.title('Tracé de la fonction f(x)') # Mise en place du titre
11 plt.xlabel('Variable x') # Mise en place du label des abscisses
12 plt.ylabel('f(x)') # Mise en place du label des ordonnées
13 plt.show() # Affichage de la figure

```

Q25. Écrire en Python, un programme permettant de tracer, dans le domaine temporel, l'allure du signal modulé $s_{4-FSK}(t)$ pour une suite de données binaires fournie sous la forme d'une liste (le regroupement bits - symbole - *mapping* - n'ayant donc pas encore été réalisé).

3 Modélisation de la chaîne de transmission

Q26. Représenter, sous la forme d'un diagramme **ibd** (diagramme de blocs internes), le diagramme fonctionnel de l'étage de transmission de la vidéo-capsule. La description ne se limitera pas aux seuls blocs étudiés dans cette partie.

Partie 4 – Acquisition des images

Objectif : analyser et modéliser la structure de la chaîne d'acquisition des images.

Le capteur d'images est composé notamment d'un élément sensible à la puissance lumineuse, appelé photodiode, d'un montage électronique de traitement et d'un convertisseur analogique numérique (CAN).

1 Analyse en régime statique de l'association photodiode et montage trans-impédance

On cherche dans un premier temps à vérifier la compatibilité des photodiodes vis-à-vis de la lumière générée par les LEDs blanches. La caractéristique de transfert des photodiodes est fournie sur la figure 17.

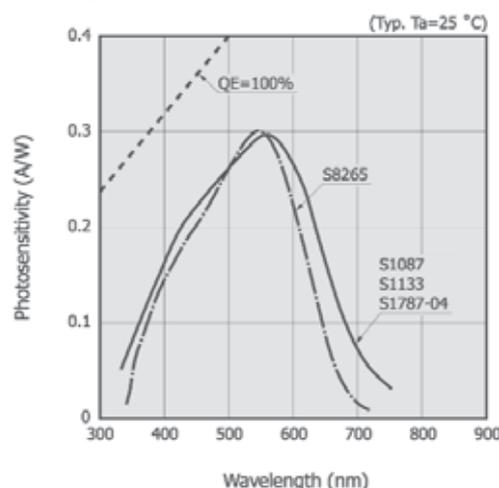


Figure 17 – Caractéristique de transfert de la photodiode

On rappelle que la lumière blanche générée par les LEDs est caractérisée par un spectre de longueur d'onde comprise entre 350 et 1 100 nm.

Q27. Valider le choix des photodiodes (S1087) et conclure sur les longueurs d'onde présentes dans le signal en sortie de la photodiode.

Une photodiode est une diode à jonction semi-conductrice. Lorsqu'elle est soumise à un éclairage lumineux de longueur d'onde convenable λ , elle est le siège d'un effet photo-électrique (génération du courant I_p) dont un modèle de connaissance est donné par :

$$I_p = I_{sat} \cdot \left(e^{\frac{q \cdot V_{pol}}{k \cdot T}} - 1 \right) - \frac{\eta \cdot \lambda \cdot q}{h \cdot c} \cdot P$$

où I_{sat} et h sont des constantes qui dépendent de la nature de la photodiode. P est la puissance reçue par la photodiode, supposée uniforme sur sa surface, λ la longueur d'onde qui correspond à la fréquence de l'onde lumineuse ν . T est la température en Kelvin (on suppose $T = 300$ K).

On rappelle les constantes physiques et les valeurs numériques :

- la charge élémentaire d'un électron $q = 1,602 \times 10^{-19}$ C ;
- la constante de Planck $h = 6,626 \times 10^{-34}$ J·s ;
- la constante de Boltzmann $k = 1,381 \times 10^{-23}$ J·K⁻¹ ;
- la vitesse de la lumière dans le vide $c = 2,998 \times 10^8$ m·s⁻¹ ;
- le rendement photonique $\eta = 0,1$.

Q28. Écrire un programme Python qui permet de tracer les 6 caractéristiques de transfert courant – tension, $I_p = f(V_{pol})$, où I_p est le courant généré par la photodiode et V_{pol} la tension de polarisation de la photodiode comprise entre -100 mV et 500 mV avec 50 valeurs, pour les puissances P de 0 à 0,01 W avec un pas constant ($n = 6$ valeurs), $I_{sat} = 100$ pA, et une longueur d'onde $\lambda = 560$ nm.

Le montage électronique de l'association photodiode et du trans-impédance est fourni sur la figure 18. La tension de sortie s'exprime par $v_{out}(t) = -R_F \cdot i_p(t)$.

Q29. Proposer un programme Python, permettant de générer la liste des tensions de sortie V_{out} (de même longueur que celle des puissances lumineuses) pour une tension de polarisation $V_{pol} = 0,25$ V et $R_F = 10$ kΩ.

On considère par la suite que la loi d'entrée-sortie du montage trans-impédance peut être modélisée par une fonction affine du type $y = a \cdot x + b$. On souhaite déterminer les coefficients a et b par la méthode des moindres carrés.

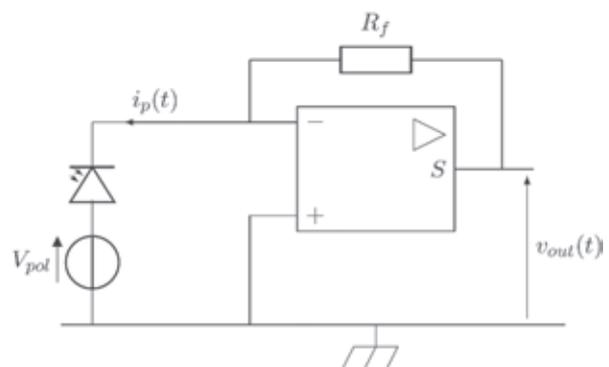


Figure 18 – Montage électronique du trans-impédance

Q30. Démontrer que les coefficients a et b permettant de minimiser la fonction $f(a, b) = \sum_{i=1}^n (y - (a \cdot x_i + b))^2$ s'écrivent $a = \frac{cov(x,y)}{\sigma_x^2}$ et $b = \bar{y} - a \cdot \bar{x}$, où :

- \bar{x} et \bar{y} les valeurs moyennes des variables x et y ;
- $cov(x, y)$ est la covariance entre les variables x et y définie par $cov(x, y) = \frac{1}{n} \cdot \sum_{i=1}^n x_i y_i - \bar{x} \bar{y}$;
- σ_x^2 la variance de la variable x définie par $\sigma_x^2 = \frac{1}{n} \cdot \sum_{i=1}^n x_i^2 - \bar{x}^2$.

Afin de valider la modélisation linéaire, on souhaite calculer le coefficient de corrélation linéaire de Bravais-Pearson r_{cc} défini par $r_{cc} = \frac{2 \cdot cov(r, e)}{\sigma_r^2 + \sigma_e^2 + (m_r - m_e)^2}$ où :

- σ_r^2 et σ_e^2 sont respectivement les variances des valeurs réelles et des valeurs estimées par le modèle linéaire ;
- m_r et m_e les valeurs moyennes respectivement du réel et du modèle linéaire ;
- $cov(r, e)$ la covariance définie par $cov(r, e) = \frac{1}{n} \cdot \sum_{i=1}^n (r_i - m_r) \cdot (e_i - m_e)$.

Q31. En Python, écrire une fonction `calc_coeffCorBP` prenant 2 listes en arguments (*liste1* et *liste2* correspondant respectivement aux valeurs réelles et à celles du modèle) et renvoyant le coefficient de corrélation linéaire de Bravais-Pearson r_{cc} .

2 Analyse en régime dynamique de l'association photodiode et montage trans-impédance

En régime dynamique (modèle petits signaux), la photodiode peut être modélisée par une source de courant I_p , une résistance parallèle R_D (supposée infinie) et une capacité de jonction C_j (10 pF) en parallèle. L'ALI (Amplificateur Linéaire Intégré) possède une capacité différentielle notée C_{diff} (1 pF), et une capacité en mode commun C_{com} (3 pF) entre l'entrée non-inverseuse de l'ALI et la masse. On note $C_D = C_j + C_{diff} + C_{com}$, la capacité équivalente à ces 3 capacités en parallèle. L'ALI (figure 19) fonctionne en régime linéaire, la tension de sortie V_{out} s'exprime en fonction de la tension différentielle ε par (p est la variable de Laplace) :

$$V_{out}(p) = A_{vd}(p) \cdot \varepsilon(p) = A_{vd}(p) \cdot (V^+(p) - V^-(p)) \quad \text{avec} \quad A_{vd}(p) = \frac{A_0}{1 + \frac{p}{\omega_0}}$$

2.1 Modélisation en régime dynamique

Le schéma équivalent de l'amplificateur à trans-impédance et de la photodiode pour des petits signaux est fourni sur la figure 19. La capacité de la photodiode C_j est supposée constante, et indépendante de la tension de polarisation V_{pol} . La mise en place de la capacité C_F permet d'augmenter la bande passante du montage trans-impédance, qui est l'objectif de la suite de l'étude.

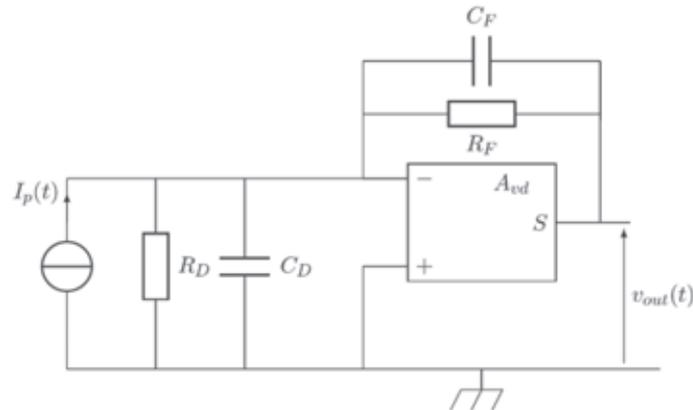


Figure 19 – Schéma équivalent petits signaux du montage trans-impédance

Le schéma équivalent petits signaux de la figure 19 peut être mis sous la forme d'un système bouclé tel que représentée sur la figure 20. Les fonctions de transfert $H_1(p)$, $H_2(p)$ et $H_3(p)$ sont telles que :

$$H_1(p) = -\frac{Z_F(p)}{1 + Z_F(p) \cdot p \cdot C_D} ; \quad H_2(p) = A_{vd}(p) ; \quad H_3(p) = \frac{1}{1 + Z_F(p) \cdot p \cdot C_D}$$

avec $Z_F(p)$ l'impédance équivalente de la résistance R_F en parallèle avec la capacité C_F .

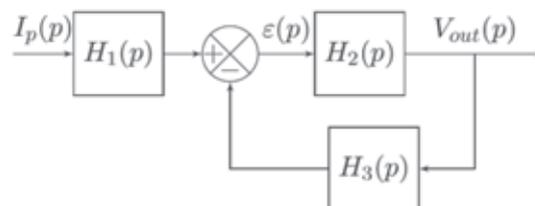


Figure 20 – Modélisation de l'ensemble photodiode et amplificateur à trans-impédance

Q32. Déterminer analytiquement la fonction de transfert $\frac{V_{out}(p)}{I_p(p)}$ et la mettre sous la forme $\frac{V_{out}(p)}{I_p(p)} = \frac{b_0}{(1+A_0)+c_0 \cdot p+c_1 \cdot p^2}$ où vous préciserez les expressions de b_0 , c_0 et c_1 en fonction de A_0 , ω_0 , R_F , C_F et C_D . Déterminer les paramètres caractéristiques de ce modèle de connaissance du second ordre lorsque $1/A_0 \ll 1$.

On souhaite désormais déterminer la valeur de la capacité C_F permettant d'obtenir un diagramme de Bode du gain ne présentant aucune résonance. La documentation de l'amplificateur propose pour une application en montage trans-impédance de choisir la capacité C_F telle que $C_F \approx \sqrt{\frac{2.C_D}{A_0.\omega_0.R_F}}$. Ce choix permet *a priori* de respecter le critère de non résonance du diagramme de Bode du gain.

On admet pour la suite que la fonction de transfert $\frac{V_{out}(p)}{P(p)}$ peut s'écrire sous la forme :

$$\frac{V_{out}(p)}{P(p)} = \frac{d_0}{1 + e_1.p + e_2.p^2}$$

où $P(p)$ représente la transformée de Laplace de la puissance lumineuse reçue par la photodiode, et où seuls les coefficients e_1 et e_2 dépendent de C_F .

2.2 Vérification des analyses

On souhaite vérifier informatiquement par analyses du gain (*via* le diagramme de Bode du gain) de la fonction de transfert $\frac{V_{out}(p)}{P(p)}$, pour différentes valeurs de C_F , si la présence d'une résonance est confirmée ou non.

On donne ci-dessous le code d'une fonction récursive **mystere**, prenant en arguments une liste **L** non vide et un entier naturel k vérifiant $0 \leq k$.

```

1 def mystere(L,k):
2     if k>=len(L):
3         return(None)
4     if k==len(L)-1:
5         return True
6     if L[k] < L[k+1] :
7         return False
8     return mystere(L,k+1)

```

Q33. Justifier que la fonction **mystere** est bien récursive, et vérifier qu'elle se termine bien. Que renvoient les exécutions de **mystere**([1,2,3,4,5,4,3,2],9), **mystere**([1,2,3,4,5,4,3,2],0), **mystere**([1,2,3,4,5,4,3,2],3) et **mystere**([1,2,3,4,5,4,3,2],5) ? En déduire le rôle de la fonction **mystere**.

Q34. Écrire en Python, un programme permettant de déterminer la valeur de C_F qui ne présente pas de résonance dans le diagramme de Bode du gain, exprimé en décibel. Ce programme devra, générer les listes des gains pour 20 valeurs de C_F linéairement réparties entre 1 pF et 20 pF pour 10^3 pulsations logarithmiquement réparties entre 10^5 et 10^8 rad·s⁻¹, et utiliser la fonction **mystere** ou une adaptation de celle-ci si nécessaire.

3 Analyse du filtre anti-repliement

Communément, dans une chaîne d'acquisition d'une grandeur analogique, on place un filtre anti-repliement entre la grandeur analogique à convertir et le Convertisseur Analogique Numérique (CAN). Le gabarit du filtre anti-repliement attendu est décrit sur la figure 21.

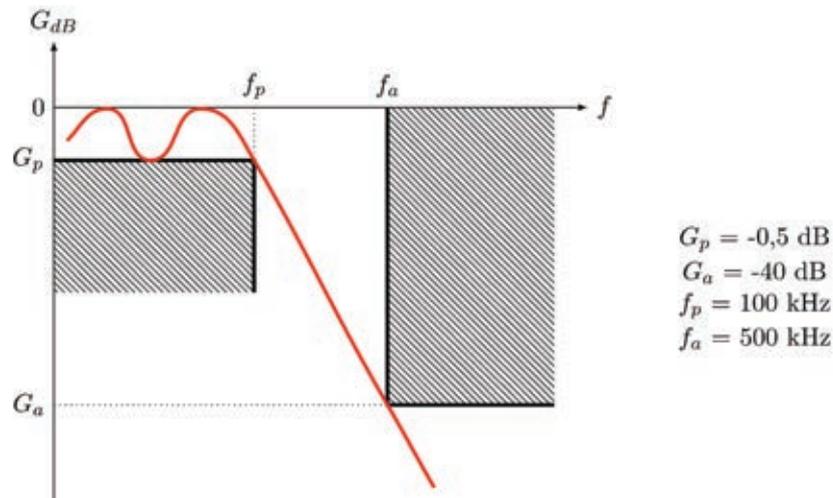


Figure 21 – Gabarit attendu du filtre anti-repliement

On note f_p la fréquence haute de la bande passante et f_a la fréquence basse de la zone atténuée. De même, on note G_p et G_a les gains exprimés en décibel associés à ces deux fréquences. De plus, on pose $T_n(x) = \cos(n \cdot \theta)$ avec $x \in [-1; 1]$ et $x = \cos \theta$. On appelle $T_n(x)$ le polynôme de Tchebychev de degré n .

Q35. Démontrer que $\forall n \in \mathbb{N}, T_{n+2}(x) = 2 \cdot x \cdot T_{n+1}(x) - T_n(x)$.

Dans le langage Python, l'instruction `numpy.poly1d(L)` permet de renvoyer le polynôme associé à la liste **L** où `L[len(L)-1]` correspond au coefficient du monôme de plus faible degré (degré 0) et `L[0]` correspond au coefficient du monôme de plus fort degré. À titre d'exemples, l'instruction `numpy.poly1d([2,3,1])` est le polynôme $2 \cdot x^2 + 3 \cdot x + 1$, l'instruction `numpy.poly1d([2,0])*numpy.poly1d([1,1])` est le polynôme $2 \cdot x^2 + 2 \cdot x$. L'opérateur addition (ainsi que soustraction) est aussi défini en termes de polynômes.

Q36. Écrire en Python la fonction récursive **Tchebychev** prenant en argument l'entier naturel **n** et renvoyant le polynôme de Tchebychev de degré n . Déterminer la complexité $C(n)$ de cet algorithme.

Le module de la fonction de transfert complexe $|H(x)|$ des filtres de Tchebychev de polynôme $T_n(x)$ de degré n s'écrit :

$$|H(x)| = \frac{1}{\sqrt{1 + \varepsilon^2 \cdot T_n^2(x)}}$$

On donne sur la figure 22 l'allure des fonctions $T_n(x)$ pour $x \in [-2; 2]$ et $n \in [1; 5]$.

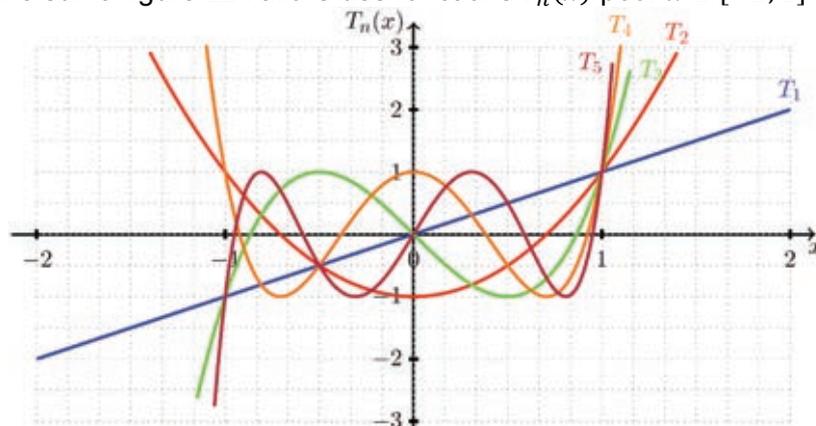


Figure 22 – Évolution de $T_n(x)$ pour $x \in [-2; 2]$

Q37. Après avoir proposé un encadrement de $T_n(x)$ sur l'intervalle $[-1; 1]$ (sans chercher à le démontrer mathématiquement), déterminer la valeur numérique de ε afin de satisfaire l'ondulation dans la bande passante pour le gabarit attendu du filtre de la figure 21.

Q38. Après avoir calculé la pente du gain (exprimé en décibel) dans la bande atténuée des filtres de Tchebychev d'ordre n , déterminer la valeur numérique de n afin de satisfaire le gabarit attendu du filtre de la figure 21. Vous préciserez les hypothèses de simplification adoptées.

4 Modélisation de l'étage d'acquisition des images

Q39. Représenter sous la forme d'un diagramme **ibd** (diagramme de blocs internes) le diagramme fonctionnel de l'acquisition par le capteur d'images de 320×280 pixels. Vous ne vous limiterez pas aux études menées dans cette partie.

Partie 5 – Synthèse

Objectif : étudier l'intérêt d'une éventuelle intégration du traitement des images dans la vidéo-capsule, en vue d'augmenter son autonomie.

Pour cette étude, la vidéo-capsule endoscopique est supposée équipée d'un microprocesseur basse consommation de type MIPS. La consommation dynamique pour 1 MIPS de $6 \mu\text{W}$. Pour traiter une image depuis son acquisition jusqu'à la classification (Adénome ou Non-adénome), le nombre d'instructions nécessaires est de 27 238 MI (*Million Instruction*). Par ailleurs, l'unité de transmission radiofréquence de la vidéo-capsule endoscopique a une efficacité énergétique de 115 nJ/bit transmis avec un débit maximum de $500 \text{ kbit} \cdot \text{s}^{-1}$.

Q40. Selon vous, quels seraient les intérêts de déporter les traitements à l'intérieur de la vidéo-capsule ? Au moins trois arguments sont attendus.

Q41. Compléter le tableau 5 et déterminer le gain énergétique des deux cas d'étude proposés (en intégrant les traitements d'images dans la vidéo-capsule endoscopique ou non).

Cas d'étude	Nombre d'images transmises par la vidéo-capsule	Nombre d'images traitées par la vidéo-capsule	Énergie consommée due à la transmission RF (J)	Énergie consommée due aux traitements des images (J)	Énergie totale consommée (J)
Sans traitement en interne	150 000				
Avec traitements en interne	887				

Tableau 5 – Tableau de synthèse

Q42. Déterminer la nouvelle autonomie théorique de la capsule, et conclure quant à la problématique d'embarquer ou non les traitements dans la vidéo-capsule endoscopique.

Document annexe

Annexe 1 – Extrait de « A Threshold Selection Method from Gray-Level Histograms »

NOBUYUKI OTSU

Abstract – A nonparametric and unsupervised method of automatic threshold selection for picture segmentation is presented. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth- and the first-order cumulative moments of the gray-level histogram.

1 Introduction

It is important in picture processing to select an adequate threshold of gray level for extracting objects from their background. A variety of techniques have been proposed in this regard. In an ideal case, the histogram has a deep and sharp valley between two peaks representing objects and background, respectively, so that the threshold can be chosen at the bottom of this valley. However, for most real pictures, it is often difficult to detect the valley bottom precisely, especially in such cases as when the valley is flat and broad, imbued with noise, or when the two peaks are extremely unequal in height, often producing no traceable valley. There have been some techniques proposed in order to overcome these difficulties. They are, for example, the valley sharpening technique, which restricts the histogram to the pixels with large absolute values of derivative (Laplacian or gradient), and the difference histogram method, which selects the threshold at the gray level with the maximal amount of difference. These utilize information concerning neighboring pixels (or edges) in the original picture to modify the histogram so as to make it useful for thresholding. Another class of methods deals directly with the gray-level histogram by parametric techniques. For example, the histogram is approximated in the least square sense by a sum of Gaussian distributions, and statistical decision procedures are applied. However, such a method requires considerably tedious and sometimes unstable calculations. Moreover, in many cases, the Gaussian distributions turn out to be a meager approximation of the real modes.

In any event, no « goodness » of threshold has been evaluated in most of the methods so far proposed. This would imply that it could be the right way of deriving an optimal thresholding method to establish an appropriate criterion for evaluating the

« goodness » of threshold from a more general standpoint. In this correspondence, our discussion will be confined to the elementary case of threshold selection where only the gray-level histogram suffices without other a priori knowledge. It is not only important as a standard technique in picture processing, but also essential for unsupervised decision problems in pattern recognition.

A new method is proposed from the viewpoint of discriminant analysis; it directly approaches the feasibility of evaluating the « goodness » of threshold and automatically selecting an optimal threshold.

2 Formulation

Let the pixels of a given picture be represented in L gray levels $[1, 2, \dots, L]$. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + \dots + n_L$. In order to simplify the discussion, the gray-level histogram is normalized and regarded as a probability distribution :

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^L p_i = 1 \quad (1)$$

Now suppose that we dichotomize the pixels into two classes C_0 and C_1 (background and objects, or vice versa) by a threshold at level k ; C_0 denotes pixels with levels $[1, \dots, k]$, and C_1 denotes pixels with levels $[k+1, \dots, L]$. Then the probabilities of class occurrence and the class mean levels, respectively, are given by :

$$\omega_0 = Pr(C_0) = \sum_{i=1}^k p_i = \omega(k) \quad (2)$$

$$\omega_1 = Pr(C_1) = \sum_{i=k+1}^L p_i = 1 - \omega(k) \quad (3)$$

and

$$\mu_0 = \sum_{i=1}^k i \cdot Pr(i|C_0) = \sum_{i=1}^k i \cdot \frac{p_i}{\omega_0} = \frac{\mu(k)}{\omega(k)} \quad (4)$$

$$\mu_1 = \sum_{i=k+1}^L i \cdot Pr(i|C_1) = \sum_{i=k+1}^L i \cdot \frac{p_i}{\omega_1} = \frac{\mu_T - \mu(k)}{1 - \omega(k)} \quad (5)$$

where

$$\omega(k) = \sum_{i=1}^k p_i \quad (6)$$

and

$$\mu(k) = \sum_{i=1}^k i \cdot p_i \quad (7)$$

are the zeroth- and the first-order cumulative moments of the histogram up to the k^{th} level, respectively, and

$$\mu_T = \mu(L) = \sum_{i=1}^L i \cdot p_i \quad (8)$$

is the total mean level of the original picture. We can easily verify the following relation for any choice of k :

$$\omega_0 \cdot \mu_0 + \omega_1 \cdot \mu_1 = \mu_T, \quad \omega_0 + \omega_1 = 1 \quad (9)$$

The class variances are given by

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 \cdot Pr(i|C_0) = \sum_{i=1}^k (i - \mu_0)^2 \cdot \frac{p_i}{\omega_0} \quad (10)$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 \cdot Pr(i|C_1) = \sum_{i=k+1}^L (i - \mu_1)^2 \cdot \frac{p_i}{\omega_1} \quad (11)$$

These require second-order cumulative moments (statistics). In order to evaluate the « goodness » of the threshold (at level k), we shall introduce the following discriminant criterion measures (or measures of class separability) used in the discriminant analysis :

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad \kappa = \frac{\sigma_T^2}{\sigma_W^2}, \quad \eta = \frac{\sigma_B^2}{\sigma_T^2} \quad (12)$$

where :

$$\sigma_W^2 = \omega_0 \cdot \sigma_0^2 + \omega_1 \cdot \sigma_1^2 \quad (13)$$

$$\sigma_B^2 = \omega_0 \cdot (\mu_0 - \mu_T)^2 + \omega_1 \cdot (\mu_1 - \mu_T)^2$$

$$\sigma_T^2 = \omega_0 \cdot \omega_1 \cdot (\mu_1 - \mu_0)^2 \quad (14)$$

(due to (9)) and

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 \cdot p_i \quad (15)$$

are the within-class variance, the between-class variance, and the total variance of levels, respectively. Then our problem is reduced to an optimization problem to search for a threshold k that maximizes one of the object functions (the criterion measures) in (12).

This standpoint is motivated by a conjecture that wellthresholded classes would be separated in gray levels, and conversely, a threshold giving the best separation of classes in gray levels would be the best threshold.

The discriminant criteria maximizing λ , κ , and η , respectively, for k are, however, equivalent to one another ; e.g., $\kappa = \lambda + 1$ and $\eta = \lambda / (\lambda + 1)$ in terms of λ , because the following basic relation always holds :

$$\sigma_W^2 + \sigma_B^2 = \sigma_T^2 \quad (16)$$

It is noticed that σ_W^2 and σ_B^2 are functions of threshold level k , but σ_T^2 is independent of k . It is also noted that σ_W^2 is based on the second-order statistics (class variances), while σ_B^2 is based on the first-order statistics (class means). Therefore, η is the simplest measure with respect to k . Thus we adopt η as the criterion measure to evaluate the « goodness » (or separability) of the threshold at level k .

The optimal threshold k^* that maximizes η , or equivalently maximizes σ_B^2 is selected in the following sequential search by using the simple cumulative quantities (6) and (7), or explicitly (6) using (2)-(5) :

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_T^2} \quad (17)$$

$$\sigma_B^2(k) = [\mu_T \cdot \omega(k) - \mu(k)]^2 \cdot \omega(k) \cdot [1 - \omega(k)] \quad (18)$$

and the optimal threshold k^* is :

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L} \sigma_B^2(k) \quad (19)$$