

Filière MP - ENS de Paris-Saclay, Lyon, Rennes et Paris - Session 2022
 Page de garde du rapport de TIPE

NOM : <i>Bordin</i>	Prénoms : <i>Duncan</i>
Classe : <i>MP*</i>	
Lycée : <i>Henri IV</i>	Numéro de candidat : <i>34712</i>
Ville : <i>Paris</i>	

Concours auxquels vous êtes admissible, dans la banque MP inter-ENS (les indiquer par une croix) :

ENS Cachan	MP - Option MP	<input type="checkbox"/>	MP - Option MPI	<input type="checkbox"/>
	Informatique	<input type="checkbox"/>		
ENS Lyon	MP - Option MP	<input type="checkbox"/>	MP - Option MPI	<input type="checkbox"/>
	Informatique - Option M	<input checked="" type="checkbox"/>	Informatique - Option P	<input type="checkbox"/>
ENS Rennes	MP - Option MP	<input type="checkbox"/>	MP - Option MPI	<input type="checkbox"/>
	Informatique	<input type="checkbox"/>		
ENS Paris	MP - Option MP	<input type="checkbox"/>	MP - Option MPI	<input type="checkbox"/>
	Informatique	<input type="checkbox"/>		

Matière dominante du TIPE (la sélectionner d'une croix inscrite dans la case correspondante) :

Informatique	<input checked="" type="checkbox"/>	Mathématiques	<input type="checkbox"/>	Physique	<input type="checkbox"/>
--------------	-------------------------------------	---------------	--------------------------	----------	--------------------------

Titre du TIPE :

Amplification pour la détection du rythme cardiaque

Nombre de pages (à indiquer dans les cases ci-dessous) :

Texte	<input type="checkbox"/>	Illustration	<input type="checkbox"/>	Bibliographie	<input type="checkbox"/>
-------	--------------------------	--------------	--------------------------	---------------	--------------------------

Attention, les illustrations doivent figurer dans le corps du texte et non en fin du document !

Résumé ou descriptif succinct du TIPE (6 lignes, maximum) :

Conception d'un algorithme de traitement vidéo capable de mettre en valeur les variations invisibles du battement cardiaques, et de calculer la fréquences des battement à partir d'une video d'un visage.

À *Paris*

Signature du professeur responsable de
la classe préparatoire dans la discipline

Le *14/06/2022*

É. Tourniaire

Signature du (de la) candidat(e)

Duncan B

LYCÉE HENRI IV
 23, Rue Clovis
 75231 PARIS Cedex 05
 Tél. 01 44 41 21 21
 Fax 01 43 29 95 21

La signature du professeur responsable et le tampon de l'établissement ne sont pas indispensables pour les candidats libres (hors CPGE).

Amplification pour la détection du rythme cardiaque

Duncan Bordin

Abstract—L'objectif de ce type était de mettre en valeur les effets visuels des pulsations sanguines. Pour cela nous avons travaillé sur des outils de traitement vidéo capables d'amplifier les mouvements. Ce domaine se rapproche du traitement du signal, une vidéo n'étant en fait qu'un type de signal avec des caractéristiques spécifiques. Nous avons dans un premier temps implémenté un algorithme naïf afin d'extraire le rythme cardiaque d'une personne à partir d'une vidéo d'un visage. Puis nous avons exploré l'état de l'art sur le sujet pour implémenter un algorithme plus performant capable de transformer la vidéo en amplifiant et en rendant visible des changements dus à l'afflux sanguin.

Index Terms—Traitement vidéo, Amplification de mouvement, Détection de fréquence

I. INTRODUCTION

Si yeux sont capables de distinguer différentes couleur ou encore d'interpréter la profondeur dans une vidéo, il est en revanche très difficile de distinguer de faibles variations. En effet de nombreux changements sont trop petits et passent ainsi sous notre seuil de perceptibilité. Certaines variations faibles nous paraissent alors invisible et même inexistantes. Pourtant lorsque le cœur bat au fond de notre poitrine, des micro-changements ont lieu à la surface. En effet, notre peau change de couleur, elle devient plus rouge quand le sang arrive et devient plus verte quand il repart. Nos veines se gonflent et se dégonflent et ce à un rythme moyen d'une fois par seconde. Cependant ces changements sont rarement perceptibles dans une vidéo (du moins d'une personne au repos). Ils sont pourtant détectables grâce au sens du toucher par exemple. La question que l'on se pose ici est: est-il possible de rendre visible et même d'extraire la fréquence des changements ?

Afin de calculer le rythme cardiaque nous nous proposons d'explorer deux pistes. Une première avec une méthode simple, capable dans un premier temps de détecter les changements et d'en extraire la fréquence. Une seconde approche plus complexe, reposant sur des travaux existants permettant non seulement de détecter la fréquence mais aussi de rendre les changements visibles dans la vidéo de manière naturelle en les amplifiant.

II. ÉTAT DE L'ART ET NOTIONS DE TRAITEMENT D'IMAGE

Nous introduisons dans cette section quelques notions nécessaire pour l'implémentation d'algorithmes sur le traitement vidéo puis abordons les articles qui se penchent sur notre sujet.

Les bases du traitement vidéo reposent sur la théorie du traitement du signal. En effet, en travaillant sur une vidéo, on

considère la vidéo comme un signal. Ce signal possède une composante temporelle, les valeurs décrivant chaque pixel évoluent effectivement au cours du temps. Le signal a aussi des composantes spatiales, une image est aussi un signal avec des directions selon sa hauteur x et sa largeur y . Ainsi un mouvement d'un objet dans une vidéo est un mouvement selon ces trois composantes. La variation de la couleur d'un pixel est en revanche un mouvement dans une seule direction: le temps t . Il est donc possible d'appliquer des outils de traitement du signal tels que la transformée de Fourier à des images ou encore à des vidéos. On décompose ainsi une image en somme de composantes de fréquences spatiales différentes. Les hautes fréquences composent alors les détails fin (faibles variations de mouvements ou de couleurs), alors que les basses fréquences représentent les variations plus lentes (zones de couleurs uniformes de l'image ou de la vidéo).

Afin de décomposer spatialement l'image pour traiter séparément les différentes fréquences, il est courant d'utiliser un outil: la pyramide laplacienne [1]. Cette représentation d'une image permet de la décomposer en une somme de bandes de fréquences spatiales décroissante. La construction de la pyramide nécessite tout d'abord d'effectuer une succession de sous-échantillonnage de l'image intercaler par l'application de passe bas afin de lisser l'image. Cette première représentation est une pyramide Gaussienne G_n où n est le nombre de sous-échantillonage effectué. Autrement dit, pour U et D respectivement des fonctions qui double et divise par deux la taille de l'image et F une fonction de flou gaussien(passe-bas) on a alors:

$$G_{n+1} = D(F(G_n)) \quad (1)$$

et la pyramide laplacienne L_n est obtenue en soustrayant l'image agrandi à l'image précédente, c'est à dire:

$$L_n = G_n - F(U(G_{n+1})) \quad (2)$$

Le processus de création de l'image est illustré en Figure 1.

Pour l'amplification des mouvements, plusieurs méthodes existent déjà. [4] se proposent de considérer un objet spécifique délimité à la main et d'étudier son mouvement sur un fond statique. [3] font une analyse sur un objet unique entouré d'autres objets statiques pour en amplifier le mouvement. Même si ces méthodes font de l'amplification de mouvement, elles nécessitent en revanche la délimitation de l'objet au préalable. De plus, ces méthodes choisissent d'amplifier des mouvements non périodique et ces mouvements ne correspondent pas nécessairement à des faibles variations.

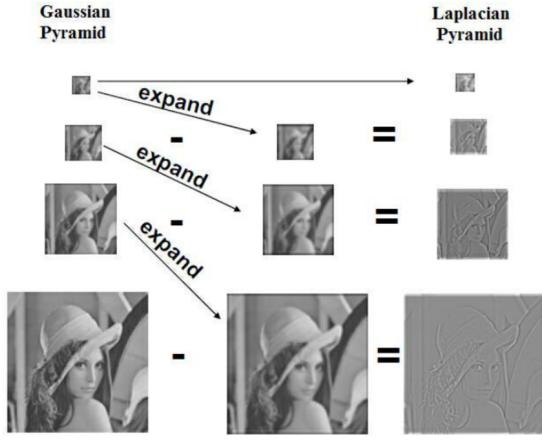


Fig. 1. Exemple de pyramide laplacienne extraite de [1].

Par conséquent il n'est pas possible de s'en inspirer pour les appliquer aux amplifications liées à l'afflux sanguin.

Une autre approche qui est celle que dont nous nous sommes inspiré ici est d'utiliser la pyramide laplacienne explicitée ci-dessus. En travaillant sur différents ordres de la pyramide laplacienne, on peut choisir d'amplifier les hautes ou basses fréquences et par conséquent les mouvements ou les couleurs. [5] montrent qu'en utilisant cette méthode il est possible de mettre en valeur les fréquences périodique mais aussi de cibler une ou plusieurs fréquences spécifiques.

III. UNE APPROCHE NAÏVE

Nous avons tout d'abord souhaité produire un algorithme qui était capable d'extraire le rythme cardiaque d'une personne à partir d'une vidéo du visage. Pour ce faire, l'intuition nous laisse à penser que l'afflux sanguin est corrélé avec la présence de la couleur rouge dans l'image. Il faut cependant prendre en compte que la variation de rouge dépend plus des mouvements importants(hochement de tête, ouverture de la bouche) que de l'afflux sanguin. Dans le but d'augmenter l'importance des faibles variations, nous avons choisi d'amplifier uniquement les variations inférieur à un certain threshold, voir Figure 2. Ainsi les petites variations de couleurs deviennent alors plus importantes et donc visibles.

La méthode implémenté se formule donc ainsi: pour un pixel donné, on note $i(t)$, l'intensité en rouge de ce pixel. On suppose que i est de la forme:

$$i(t) = C + A \cos(\omega t) + E(t) \quad (3)$$

Avec C la couleur moyenne de la peau et E l'influence de l'éclairage sur la couleur. L'afflux sanguin étant invisible à l'oeil nu, $A \cos(\omega t)$ est donc négligeable devant les autres composantes. La variation d'éclairage étant souvent plus intense que A , on choisit un threshold α tel que $E(t)$ ne soit pas amplifié et ainsi pour tout pixel:

$$i_{\text{amplified}} = C + \alpha A \cos(\omega t) + E(t) \quad (4)$$

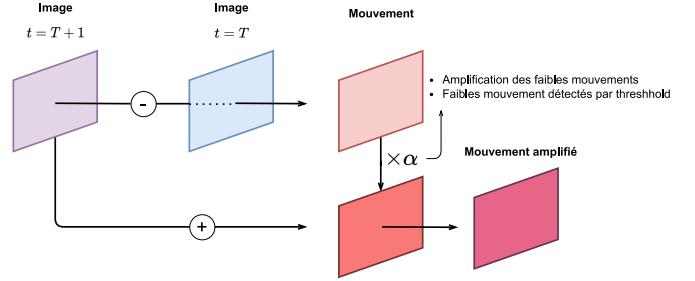


Fig. 2. Illustration de la première méthode. Amplification du rouge dans l'image.

Pour α assez grand, $A \cos(\omega t)$ devient alors une variation importante: la pulsation cardiaque est alors visible.

Afin d'extraire le rythme cardiaque ω , on peut étudier l'évolution de la quantité de rouge dans l'image. Il est néanmoins démontré que la peau étant rouge, il est plus intéressant de regarder l'évolution du vert [2]. En appliquant une transformée de Fourier au signal de la moyenne de vert par image on retrouve ainsi ω .

Cette méthode fonctionne, mais la quantité de bruit est trop importante pour que l'amplification des mouvements soit réellement visible. Il existe en effet des bruits avec des variations suffisamment faible pour passer sous le threshold. On ne peut donc l'utiliser pour amplifier le mouvement.

IV. AMPLIFICATION DU MOUVEMENT

La seconde méthode implémentée repose sur des travaux existants [5]. Cet algorithme permet d'amplifier des micros changements périodiques et donc permet à la fois d'amplifier les changements de couleur de la peau que le premier algorithme peinait à faire, mais aussi d'amplifier le mouvement des veines qui se gonflent périodiquement.

Cette méthode consiste à choisir la bande de fréquence temporelle (et non spatiale) d'intérêt (dans notre cas la fréquence cardiaque) puis de l'amplifier.

Il est de même possible de choisir d'amplifier les couleurs ou bien les mouvements. En effet en utilisant une pyramide laplacienne il suffit de choisir la bande de fréquence cette fois-ci spatiale que l'on souhaite amplifier.

La bande spatiale choisie est donc soumise à un passe-bande temporelle pour extraire les fréquences désirées. On choisit les fréquences de passage du passe bande à une plage qui contient les fréquences correspondante aux battements du coeur: 0.8 – 1.5Hz soit 48 – 90 battement par minutes.

Le signal de sortie est ensuite multiplié par un coefficient α d'amplification. Il ne nous reste alors plus qu'à l'additionner au signal d'origine pour obtenir une vidéo avec des mouvements amplifiés 3. Puisque que l'on se contente de faire passer

certains ordre de pyramides laplacienne au travers d'un passe-bande, il peut paraître surprenant que cela permette d'amplifier des mouvement. Une justification est proposée en annexe.

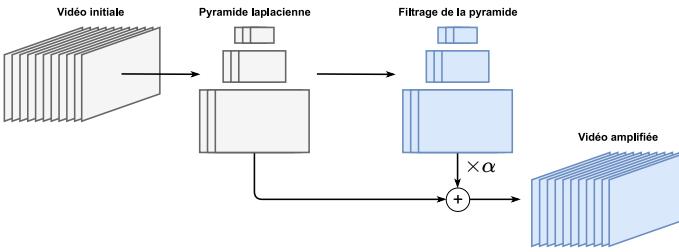


Fig. 3. Algorithme récapitulatif de la seconde méthode.

Filtrer et amplifier qu'un nombre limité de fréquences spatiales permet de diminuer la puissance de calcul et l'espace en mémoire nécessaire, mais surtout de limiter le bruit souvent très basse fréquence. Malgré cette considérations, du bruit reste amplifié. En sélectionnant une unique fréquence il serait possible de le restreindre encore plus mais avec une perte sur les performances d'amplification du mouvement.

V. DÉTAILS D'IMPLÉMENTATION ET RÉSULTATS

Lors de l'implémentation de la première méthode, j'ai fait face a deux problèmes. A cause d'une implémentation rapide et de boucle `for` imbriqué les calculs étaient conséquent: 20 minutes pour une vidéo 512,512 de 30s. De plus, comme mentionné précédemment, les vidéos utilisées présentent trop de bruit pour rendre visible la pulsation 4. Cependant les résultats d'extraction restent satisfaisant. Pour une pulsation mesurées à 63bpm pendant la vidéo l'analyse de Fourier retrouve cette valeur 5.



Fig. 4. L'augmentation du rouge dans l'image n'est pas pertinent visuellement à cause du bruit, par exemple due au mouvement des cheveux.

Pour la seconde méthode nous avons utilisé des outils de calculs rapide sur les tableaux grâce à la bibliothèque numpy ainsi que scipy permettant de grandement réduire les temps de calculs. On obtient ainsi des résultat en moins de 5 min pour une vidéo de 13 secondes de même dimension. Les résultats

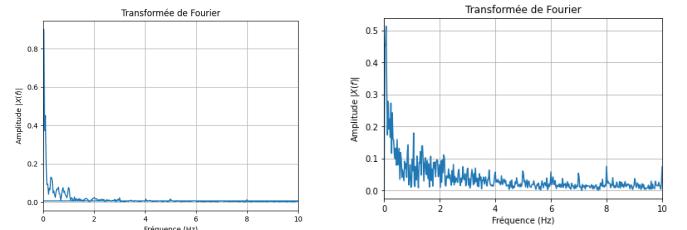


Fig. 5. L'analyse de Fourier sur la vidéo montre un pic à 63bpm après amplification.

sont satisfaisant et permettent bien de voir l'amplification du mouvement des veines, mais on peut cependant remarquer la présence tout aussi importante de bruit 6.

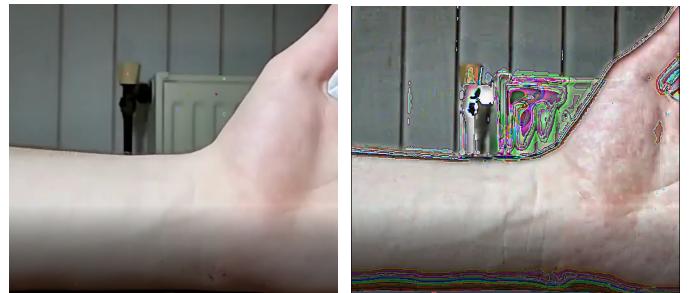


Fig. 6. Amplification du mouvement. Visible notamment sur les veines mais aussi sur la paume de la main. Malheureusement pas aussi bien rendu en image qu'en vidéo.

Une méthode employée pour réduire la quantité de bruits à été de resserrer les fréquences de coupures du passe-bandes. Pour cela nous avons pu employer l'approche naïve pour tout d'abord extraire la fréquence du rythme cardiaque.

Si la complexité temporelle ne m'a pas fait défaut pour le calcul des résultats, la complexité spatiale est elle importante. En effet, manipuler des images en RAM pose problème, la représentation étant faite sur des entiers en 8bits mais les calculs de fréquences sur 64 bits. Ne possédant que 8G de RAM pour les calculs, nous nous sommes contentés d'amplifier sur une vidéo de 15 secondes. Ce fut cependant suffisant pour observer des pulsations visibles.

VI. CONCLUSION ET PERSPECTIVES

Nous avons réussi à produire un algorithme de traitement vidéo capable de détecter le rythme cardiaque d'une personne immobile dont on filmerai le visage. Nous avons aussi réussi à produire un algorithme capable d'amplifier de petits changements périodiques permettant ainsi de rendre les effets de l'afflux sanguin visible. Cependant le premier programme est un peu lent et pourrait bénéficier d'une optimisation. Le deuxième souffre aussi d'une production importante d'artefacts et mériterait donc peut être d'être amélioré afin de réduire le bruit de fond.

VII. BIBLIOGRAPHIE

REFERENCES

- [1] Peter J Burt and Edward H Adelson. “The Laplacian pyramid as a compact image code”. In: *Readings in computer vision*. Elsevier, 1987, pp. 671–679.
- [2] Weijia Cui, Lee E Ostrander, and Bok Y Lee. “In vivo reflectance of blood and tissue as a function of light wavelength”. In: *IEEE transactions on biomedical engineering* 37.6 (1990), pp. 632–639.
- [3] Ce Liu et al. “Motion magnification”. In: *ACM transactions on graphics (TOG)* 24.3 (2005), pp. 519–526.
- [4] Jue Wang et al. “The cartoon animation filter”. In: *ACM Transactions on Graphics (TOG)* 25.3 (2006), pp. 1169–1173.
- [5] Hao-Yu Wu et al. “Eulerian video magnification for revealing subtle changes in the world”. In: *ACM transactions on graphics (TOG)* 31.4 (2012), pp. 1–8.

VIII. ANNEXE

Montrons que le procédé amplifie aussi les mouvements. Pour simplifier la chose nous nous plaçons en une dimension et non plus sur un pixel et nous supposons que la vidéo va présenter une translation. On a donc que i dépend de x et de t . Si on fixe x , comme l'image subit une translation au cours du temps on peut supposer que si ce déplacement suit la fonction d on a alors

$$i(x, t) = f(x + d(t))$$

Dans le but d'amplifier le mouvement par α il faudrait avoir :

$$i_{amplified} = f(x + (1 + \alpha)d(t))$$

On va partir du principe que $d(t)$ est petit et que i est dérivable. Alors en faisant un développement limité à l'ordre 1 on a alors :

$$i \approx f(x) + \frac{\partial f}{\partial x}(x)d(t)$$

Si on suppose que d est dans la fréquence de passage de notre passe bande on a alors :

$$i_{filtered} \approx \frac{\partial f}{\partial x}(x)d(t)$$

donc

$$i_{amplified} \approx f(x) + \frac{\partial f}{\partial x}(x)(1 + \alpha)d(t)$$

On a donc, toujours avec Taylor que :

$$i_{amplified} = f(x + (1 + \alpha)d(t))$$

Le mouvement a bien été amplifié.

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Dec 27 18:37:25 2021
4
5 @author: duncan bordin
6 """
7
8
9 from PIL import Image
10 import numpy as np
11
12
13
14
15
16 def numero(i):
17     number = str(i)
18     return "0"*(5-len(number))+number
19
20 # Importer l'image
21
22
23 directory = r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\ensemble
d'image\image63bpm0000-"
24
25
26 #image = Image.open(directory+"00001.jpg" )
27 #image2=image.crop((400, 750, 900,1550))
28 #print(image)
29 #image2.show()
30 tableau_image=np.array(image)
31
32
33 def difference(i):
34     nom = directory
35     image1 = Image.open(nom+numero(i)+".jpg")
36     image2 = Image.open(nom+numero(i+1)+".jpg")
37     image1 = image1.crop((400, 950, 900, 1450))
38     image2 = image2.crop((400, 950, 900, 1450))
39     tab1 = np.array(image1, dtype='int64')
40     tab2 = np.array(image2, dtype='int64')
41     return tab2-tab1
42
43
44 def filtrecell(tab, i, j):
45     coefmulti = 25
46     s = 0
47     for l in range(3):
48         for k in range(3):
49             for m in range(3):
50                 s += tab[i*3+l][j*3+k][m]
51     if s/9 < 30 and s/9 > 1.5:
52         for l in range(3):
53             for k in range(3):
54                 for m in range(3):
55                     tab[i*3+l][j*3+k][m] = int(tab[i*3+l][j*3+k][m]*coefmulti)
56                     #tab[i*3+l][j*3+k][0]=255
57
58
59 def filtre_image(tab):
60     for i in range(len(tab)//3):
61         for j in range(len(tab[1])//3):
62             filtrecell(tab, i, j)
63
64
65 def filtreapplique(i):
66     nom = directory
67     image1 = Image.open(nom+numero(i)+".jpg")
68     image1 = image1.crop((400, 950, 900, 1450))
69     tab1 = np.array(image1)
70     diff = difference(i)
71     filtre_image(diff)

```

```

72 newimage = np.copy(tab1)
73 for i in range(len(tab1)):
74     for j in range(len(tab1[1])):
75         for k in range(3):
76             s = tab1[i][j][k]+diff[i][j][k]
77             if s < 0:
78                 s = 0
79             if s > 255:
80                 s = 255
81             newimage[i][j][k] = s
82 return newimage
83
84
85
86
87
88
89
90
91
92
93 l = filtreapplique(8)
94 nouvelle = Image.fromarray(l, 'RGB')
95 nouvelle.show()
96
97 def quantite(i):
98     l = filtreapplique(i)
99     r, g, b = 0,0,0
100    for i in range(len(l)):
101        for j in range(len(l[1])):
102            r += l[i][j][0]
103            g += l[i][j][1]
104            b += l[i][j][2]
105    return r, g, b
106
107
108 rouge, vert, bleu=[], [], []
109 for i in range(1, 10):
110     r, g, b = quantite(i)
111     rouge.append(r)
112     vert.append(g)
113     bleu.append(b)
114
115
116 np.savetxt(r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\fichier
117 texte\rouge.txt", rouge)
118 np.savetxt(r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\fichier
119 texte\1vert.txt", vert)
120 np.savetxt(r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\fichier
texte\1bleu.txt", bleu)

```

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Mon Dec 27 19:30:18 2021
4
5 @author: duncan bordin
6 """
7
8
9 import numpy as np
10 from numpy.fft import fft, fftfreq
11 import matplotlib.pyplot as plt
12
13 rouge=np.genfromtxt(r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\fichier
14 texte\rouge.txt","int")
15 vert=np.genfromtxt(r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\fichier
16 texte\vert.txt","int")
17 bleu=np.genfromtxt(r"C:\Users\duncan bordin\Desktop\tipe\video_63_bpm\fichier
18 texte\bleu.txt","int")
19
20 rouge = rouge/250000
21 vert = vert/250000
22 bleu = bleu/250000
23
24 def normalisation(l):
25     moyenne=np.sum(l)/len(l)
26     ecart = 0
27     for i in l:
28         ecart += abs(i-moyenne)**2
29     ecart = (ecart/len(l))**(0.5)
30     ll = []
31     for i in l:
32         ll.append((i-moyenne)/ecart)
33     return ll
34
35 rouge2 = normalisation(rouge)
36 vert2 = normalisation(vert)
37 bleu2 = normalisation(bleu)
38
39
40 duree = 1/30*1215
41 Te = 1/30
42
43 vert2 = np.array(vert2)
44 rouge2 = np.array(rouge2)
45 bleu2 = np.array(bleu2)
46 te = np.linspace(0,duree,1215)
47
48 N = 1215
49
50 X = fft(vert2)
51 freq = fftfreq(vert2.size, d =Te)
52
53 # On prend la valeur absolue de l'amplitude uniquement pour les fréquences positives
54 X_abs = np.abs(X[:N//2])
55 # Normalisation de l'amplitude
56 X_norm = X_abs*2.0/N
57 # On garde uniquement les fréquences positives
58 freq_pos = freq[:N//2]
59
60 plt.plot(freq_pos, X_norm, label="Amplitude absolue")
61 plt.xlim(0, 10) # On réduit la plage des fréquences à la zone utile
62 plt.grid()
63
64
```

```
65 plt.xlabel(r"Fréquence (Hz)")
66 plt.ylabel(r"Amplitude $|X(f)|$")
67 plt.title("Transformée de Fourier")
68 plt.show()
69
70
71
72 def cardf():
73     card=0
74     puiss=0
75     for i in range(len(freq_pos)):
76         if freq_pos[i]>0.75 and freq_pos[i]<4:
77             if X_norm[i]>puiss:
78                 card =freq_pos[i]
79                 puiss=X_norm[i]
80     return card*60
81
82 print(cardf())
83
```