extraction/tracking_3D.py

```
#suivi de grains

#importations
import os
import numpy as np
#donnes
n_tempo = 16 # nb de temps
n_coupes_xy = 250 # dimensions des colones en nombres de coupes
n_coupes_xz = 80
n_coupes_yz = 80

seuil = 170 #pour les images en noir et blanc




#appel de fc de pink
def command(cmd):
    os.system("../pink/linux/bin/"+cmd)


#pour recuperer les elements d un .list
def parse_list(file_name):
    f = open(file_name, "r")
    line = f.readline()
    mode = line[0]
    n_els = int(line[2:][:-1])
    liste = []
    for i in range(n_els):
        liste.append([])
        line = f.readline()[:-1];
        for s in line.split(" "):
            liste[i].append(int(s))
    f.close()
    return liste

def export_list(liste, file_name):
    os.system("rm "+file_name)
    f = open(file_name, "w")
    f.write("_ "+str(len(liste))+"\n")
    for elements in liste:
        f.write(" ".join(str(el) for el in elements)+"\n")
    f.close()


#fonction pour numeroter les fichiers
def numerote(n, l):
   s = str(n)
   while len(s) < l:
       s = '0' + s
   return s



def retrouve_grain(x,y,z,t):

    resultats=np.load("../extraction/tracking_3D/resultats.npy")
```

```
    grains=np.load("../extraction/tracking_3D/grains.npy")
    vitesses=np.load("../extraction/tracking_3D/vitesses.npy")
    moyenne=np.load("../extraction/tracking_3D/vitesse_moy_grains.npy")
    vitesse_moyenne=moyenne[0]
    acc_moyenne=moyenne[1]
    padding_temporel = numerote(t, 2)


    #retrouve le grain et sa coupe
    command ("selectcomp
../extraction/images_3D/image_3D_superpose_inv_t"+padding_temporel+".pgm 26
"+str(x)+" "+str(y)+" "+str(z)+
             "
../extraction/tracking_3D/track_t_"+padding_temporel+"_"+str(x)+"_"+str(y)+
"_"+str(z)+".pgm")
    #vtk
    command("mcube
../extraction/tracking_3D/track_t_"+padding_temporel+"_"+str(x)+"_"+str(y)+
"_"+str(z)+
            ".pgm 0 5 0 VTK
../extraction/tracking_3D/track_t_"+padding_temporel+"_"+str(x)+"_"+str(y)+
"_"+str(z)+".vtk")

    lien_vtk=
"../extraction/tracking_3D/track_t_"+padding_temporel+"_"+str(x)+"_"+str(y)
+"_"+str(z)+".vtk"
    #barycentre
    command("3dlabel
../extraction/tracking_3D/track_t_"+padding_temporel+"_"+str(x)+"_"+str(y)+
"_"+str(z)+
            ".pgm
../extraction/labels_3D/label_t_"+padding_temporel+"_"+str(x)+str(y)+str(z)
+".pgm")

    command("barycentrelab
../extraction/labels_3D/label_t_"+padding_temporel+"_"+str(x)+str(y)+str(z)
+
            ".pgm
../extraction/bary_3D/bary_3D_t"+padding_temporel+"_"+str(x)+str(y)+str(z)+
".pgm")

    command("pgm2list
../extraction/bary_3D/bary_3D_t"+padding_temporel+"_"+str(x)+str(y)+str(z)+
            ".pgm B
../extraction/bary_3D/liste/bary_list_t"+padding_temporel+"_"+str(x)+str(y)
+str(z)+".list")


    bary=
parse_list("../extraction/bary_3D/liste/bary_list_t"+padding_temporel+"_"+s
tr(x)+str(y)+str(z)+".list")

    if (bary==[]):
        return 0
    else:
        xb=bary[0][0]
        yb=bary[0][1]
        zb=bary[0][2]
        volume=0
        for grain in grains:
            if(grain[1][t]==[xb, yb, zb]):
```

```
                coord0=grain[1][0]
                volume=grain[0]
                break
        for grain in vitesses:
            if( grain[0]==coord0):
                v=grain[1]
                a=grain[2]
                break


        for grain in range(len(resultats)):
            if(resultats[grain][0][t]==xb and resultats[grain][1][t]==yb
and resultats[grain][2][t]==zb):

                return
[volume,resultats[grain],v,a,vitesse_moyenne,acc_moyenne,lien_vtk]
```