

Manual Block Modifier

Prerrequisitos

Para verificar que Python3 está instalado:

```
python3 -V
```

Instalar PIP para Python3

```
sudo apt-get install python3-pip
```

Instalar Numpy para Python3

```
pip3 install numpy
```

Experimentos

Para hacer los experimentos del block modifier primero hay que obtener los datos necesarios para el experimento (métricas, deltas, desplazamientos). Para los primeros pasos usaremos Linux ya que el desarrollo de LHE de ffmpeg está hecho en este sistema operativo.

1) En Linux.

Para obtener las métricas hay que descargar el repositorio de LHE de [aquí](#). Una vez tenemos el repositorio de LHE hay que descargar el repositorio "Images" de [aquí](#). En este repositorio ya hay carpetas con las métricas y los fotogramas de los vídeos de la librería de ejemplos como el vídeo de los patinadores en la carpeta "Ice", el de la autopista en la carpeta "SD", el del conejo en "SD_Conejo" y el del hombre en el sillón en "SD_sillon". Del vídeo del tren en HD faltaría sacar las métricas. Las métricas de cada vídeo están en la carpeta "MetricasXXXXXX".

El primer paso será instalar ffmpeg con las instrucciones que hay en el repositorio de LHE.

Después solo hay que descargar el video del que se quieran obtener las métricas, obtener los fotogramas en una carpeta con el comando de ffmpeg: **ffmpeg -i Rutavideo frame%d.bmp** y ejecutar el script métricas.sh con este comando: **./metricas directorio/** esto creará un directorio con las métricas de cada frame en archivos de texto. Una vez hecho esto se puede usar Windows o Linux siempre que se tenga instalado Python 3 y Java.

2) En Linux o Windows.

En este momento tenemos que tener una carpeta con todas las imágenes de los fotogramas del vídeo con el que estamos trabajando y una carpeta con las métricas de esos fotogramas. Ahora hay que descargar el repositorio "Block-Modifier" de [aquí](#).

En este repositorio hay varias carpetas donde hay algunos archivos que se han utilizado para hacer distintos experimentos. Los ficheros más importantes son: **CrearFicheroEntrenamiento.py**, **Funciones_BM.py**, **Funciones_div_im.py** y **TempladoSimuladov2.py**.

El fichero que hay que ejecutar es CrearFicheroEntrenamiento.py, este pide que se le pase la ruta de las carpetas donde están las métricas y los fotogramas además de un nombre para el fichero de entrenamiento que se quiere crear.

Para elegir los datos del fichero de entrenamiento que se quiere, en ese archivo hay que modificar la parte donde se crea el csv y seleccionar los datos que se quieren sacar.

```
array_pr1 = bm.metricas(f, bloq-33, rutaMetricas)
array_pr2 = bm.metricas(f, bloq-32, rutaMetricas)
array_pr3 = bm.metricas(f, bloq-31, rutaMetricas)
array_pr4 = bm.metricas(f, bloq-1, rutaMetricas)
array_pr5 = bm.metricas(f, bloq, rutaMetricas)
array_pr6 = bm.metricas(f, bloq+1, rutaMetricas)
array_pr7 = bm.metricas(f, bloq+31, rutaMetricas)
array_pr8 = bm.metricas(f, bloq+32, rutaMetricas)
array_pr9 = bm.metricas(f, bloq+33, rutaMetricas)
array_pr10 = bm.metricas(f-1, bloq-33, rutaMetricas)
array_pr11 = bm.metricas(f-1, bloq-32, rutaMetricas)
array_pr12 = bm.metricas(f-1, bloq-31, rutaMetricas)
array_pr13 = bm.metricas(f-1, bloq-1, rutaMetricas)
array_pr14 = bm.metricas(f-1, bloq, rutaMetricas)
array_pr15 = bm.metricas(f-1, bloq+1, rutaMetricas)
array_pr16 = bm.metricas(f-1, bloq+31, rutaMetricas)
array_pr17 = bm.metricas(f-1, bloq+32, rutaMetricas)
array_pr18 = bm.metricas(f-1, bloq+33, rutaMetricas)
```

```
writer = csv.writer(csvfile, delimiter=',',
                    quotechar=';', quoting=csv.QUOTE_MINIMAL)
writer.writerow([str(contador_linea)+"->", |
array_pr1[0], array_pr1[1], array_pr2[1], array_pr3[1],
array_pr4[0], array_pr4[1], array_pr4[2], array_pr4[3], array_pr5[1], array_pr5[3], array_pr6[1], array_pr6[3],
array_pr7[0], array_pr7[1], array_pr8[1], array_pr9[1],

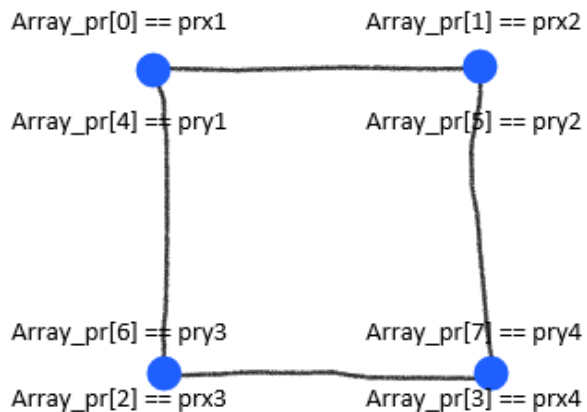
array_pr10[0], array_pr10[1], array_pr11[1], array_pr12[1],
array_pr13[0], array_pr13[1], array_pr13[2], array_pr13[3], array_pr14[1], array_pr14[3], array_pr15[1], array_pr15[3],
array_pr16[0], array_pr16[1], array_pr17[1], array_pr18[1],

array_pr1[4], array_pr1[5], array_pr2[5], array_pr3[5],
array_pr4[4], array_pr4[5], array_pr4[6], array_pr4[7], array_pr5[5], array_pr5[7], array_pr6[5], array_pr6[7],
array_pr7[4], array_pr7[5], array_pr8[5], array_pr9[5],

array_pr10[4], array_pr10[5], array_pr11[5], array_pr12[5],
array_pr13[4], array_pr13[5], array_pr13[6], array_pr13[7], array_pr14[5], array_pr14[7], array_pr15[5], array_pr15[7],
array_pr16[4], array_pr16[5], array_pr17[5], array_pr18[5],

valorx+0.5, valory+0.5])
```

En este ejemplo se crea un fichero de entrenamiento con las PRx de 4 líneas y las PRy de las mismas líneas. Los array de PR contienen primero las 4 PRx del bloque que se le ha pasado en la función “bm.metricas(fotograma, bloque, ruta)” y a continuación las 4 PRy. El orden de las PR es el siguiente:



Hay casos en el que los fotogramas tienen el nombre de frameSDXX.bmp y otros en el que es frameXX.bmp. En el caso de que se quieran usar los dos nombres hay que cambiar este nombre en la ruta de los ficheros en el archivo CrearFicheroEntrenamiento.py:

```

im = Image.open(archivo+"/frameSD"+str(f)+".bmp")
im = im.convert('L')
im2 = Image.open(archivo+"/frameSD"+str(f-1)+".bmp")
im2 = im2.convert('L')

```

Y en el fichero Funciones_BM.py:

```

543 with open(rutaMetricas + "/metricasframe"+str(fotograma)+".txt") as archivo_json:

```

En el caso de que no se quieran tener los movimientos en los bloques porque ya se tengan o porque no sean necesarios, conviene comentar en CrearFicheroEntrenamiento.py la línea:

```

66 (resultado, minimo, valorx, valory) = bm.experimento_despl_v2(a1, a2, bloq, ancho_bloq)

```

Ya que es la función que más tiempo gasta de todo el programa.

Utilizando el script create_training_file.sh

Para ejecutar el script basta con hacer

```

chmod +x ./create_training_file.sh

```

```

./create_training_file.sh <frames> <metricas> <nombre_csv>

```

Por ejemplo,

```

./create_training_file.sh Ice MetricasIce Test

```

ATENCION. Se deben modificar las variables FRAMES_ROUTE y METRICS_ROUTE según la ruta en la que se encuentren las carpetas que contienen las métricas y los frames. Por ejemplo:

```
FRAMES_ROUTE="$HOME/Vectores/Images/$1"
```

```
METRICS_ROUTE="$HOME/Vectores/Images/$2"
```

Por otro lado, las modificaciones de código mencionadas en el apartado anterior se pueden realizar, de manera rápida, gracias al script `create_training_file.sh`. Hay dos parámetros:

- EXTENSION

Si se quiere usar ficheros que tengan el nombre `frameSDXX.bmp` para los frames y `metricasframeSD` para las métricas, bastará con poner.

```
EXTENSION = "SD"
```

Si no, hay que dejarlo en blanco.

```
EXTENSION = ""
```

- BLOCK_MOVEMENT

Si se quiere que el programa calcule los movimientos, se escribirá

```
BLOCK_MOVEMENT=yes
```

En caso contrario,

```
BLOCK_MOVEMENT=no
```

Utilizando el script `create_training_file_1block.sh`

Para ejecutar el script basta con hacer

```
chmod +x ./create_training_file_1block.sh
```

```
./create_training_file_1block.sh <frames> <metricas> <nombre_csv>
```

Por ejemplo,

```
./create_training_file_1block.sh Ice MetricasIce Test
```

ATENCIÓN. Se deben modificar las variables `FRAMES_ROUTE` y `METRICS_ROUTE` según la ruta en la que se encuentren las carpetas que contienen las métricas y los frames. Por ejemplo:

```
FRAMES_ROUTE="$HOME/Vectores/Images/$1"
```

`METRICS_ROUTE="$HOME/Vectores/Images/$2"`

Se añade un nuevo parámetro con respecto al script anterior.

- `N_BLOCKS`

El número de bloques que se quieren procesar. El resultado de cada bloque irá en un fichero diferente.