Musical Gestures Toolbox
Documentation

Frida Furmyr & Marcus Widmer

July 15, 2019

`motionfilter`

`mg_videoreader`

```
    mg_videoreader(filename, starttime = 0, endtime = 0, skip = 0, contrast = 0, brightness
= 0, crop = 'none')
```

Class MgObject

Initializes Musical Gestures data structure from a given parameter video file.
Parameters:

- filename (str): Name of input parameter video file.
- method (str): Currently 'Diff' is the only implemented method.
- filtertype (str): 'Regular', 'Binary', 'Blob' (see function motionfilter). thresh (float): a number in [0,1]. Eliminates pixel values less than given threshold. starttime (float): cut the video from this start time (min) to analyze what is relevant. endtime (float): cut the video at this end time (min) to analyze what is relevant. blur (str): 'Average' to apply a blurring filter, 'None' otherwise. skip (int): When proceeding to analyze next frame of video, this many frames are skipped. color (bool): True does the analysis in RGB, False in grayscale. contrast (float): apply +/- 100 contrast to video brightness (float): apply +/- 100 brightness to video crop (str): 'none', 'manual', 'auto' to select cropping of relevant video frame size

mg_videoreader

```
    mg_videoreader(filename, starttime = 0, endtime = 0, skip = 0, contrast = 0, brightness = 0, crop = 'none')

    Reads in a video file, and by input parameters user decide if it: trims the length, skips frames, applies c

    filename (str): Name of input parameter video file.
    starttime (float): cut the video from this start time (min) to analyze what is relevant.
    endtime (float): cut the video at this end time (min) to analyze what is relevant.
    skip (int): When proceeding to analyze next frame of video, this many frames are skipped.
    contrast (float): apply +/- 100 contrast to video
    brightness (float): apply +/- 100 brightness to video
    crop (str): 'None', 'Auto' or 'Manual' to crop video.
```

```
return:
- vidcap: cv2 video capture of editevideo file
- length, fps, width, height from vidcap
- of: filename gets updated with whaprocedures it went through
```

motionvideo

```
motionvideo(self, method = 'Diff', filtertype = 'Regular', thresh = 0.001, blur = 'None', kernel_size = 5):

Finds the difference in pixel value from one frame to the next in an input video, and saves the frames into
Describes the motion in the recording.
Outputs a video called filename +'_motion.avi'.

Parameters:
kernel_size (int): Size of structuring element.
method (str): Currently 'Diff' is the only implemented method.
filtertype (str): 'Regular', 'Binary', 'Blob'(see function motionfilter)
thresh (float): a number in [0,1]. Eliminate spixel values less than given threshold.
blur (str): 'Average' to apply a blurring filter, 'None' otherwise.

Returns:
None
```

filter

```
motionfilter(motion_frame, filtertype,thresh,kernel_size)

Apply a filter to a picture/videoframe

motion_frame (array(uint8)): input motion image
filtertype (str):
            'Regular', turns all values below thresh to 0,
            'Binary' turns all values below thresh to 0, above thresh to 1,
            'Blob' removes individual pixels with erosion method.
thresh (float): for 'Regular' and 'Binary' option, thresh is a value of threshold [0,1];
kernel_size(int): Size of structuring element

return: filtered frame (array(uint8))
```

mg_centroid

```
mg_centroid(image, width, height):

Computes the centroid of an image/frame.

Parameters
- image (uint8)
```

```
    - width/height of image

    Returns:
    - Centroid of motion: Where was the maximum change in pixel value
    - Quantity of motion: How large was the change in pixel value
```

constrainNumber

```
    constrainNumber(n, minn, maxn)
    Constrains number to having a value between minn and maxn

    Parameters:
    - n (number)
    - minn (lower limit n can be)
    - maxn (lower limit n can be)

    return:
    Constrained number
```

cropvideo

```
    cropvideo(fps,width,height, length, of, crop_movement = 'auto', motion_box_thresh = 0.1, motion_box_margin
Crops the video.

Parameters:
- crop_movement: {'auto','manual'}
'Auto' finds the bounding box that contains the total motion in the video.
Motion threshold is given by motion_box_thresh.
'manual' opens up a simple GUI that is used to crop the video manually
by looking at the first frame

- motion_box_thresh: float
Only meaningful is crop_movement = 'auto'. Takes floats between 0 and 1,
where 0 includes all the motion and 1 includes none

- motion_box_margin: int
Only meaningful is crop_movement = 'auto'. Add margin to the bounding box.
Returns:
- None
```

input_test

```
    input_test(filename,method,filtertype,thresh,starttime,endtime,blur,skip):
    """ Gives feedback to user if initialization from input went wrong. """
    Ex: raise InputError(msg)
    msg = 'Please specify a filter type as str: Regular or Binary'
```

motionhistory

```
    motionhistory(self, history_length = 20, kernel_size = 5, method = 'Diff', filtertype = 'Regular', thresh =

    Finds the difference in pixel value from one frame to the next in an input video, and saves the difference
    The history frames are summed up and normalized, and added to the current difference frame to show the hist
    Outputs a video called filename + '_motionhistory.avi'.

    Parameters:
    history_length (int): How many frames will be saved to the history tail.
    kernel_size (int): Size of structuring element.
    method (str): Currently 'Diff' is the only implemented method.
    filtertype (str): 'Regular', 'Binary', 'Blob' (see function motionfilter)
thresh (float): a number in [0,1]. Eliminates pixel values less than given threshold.
    blur (str): 'Average' to apply a blurring filter, 'None' otherwise.

    Returns:
    None
```

contrast_brightness

```
    contrast_brightness(of,vidcap,fps,width,height,contrast,brightness):
    Edit contrast and brightness of the video.

    of (str): filename without extension
    vidcap: cv2 capture of video file, with all frames ready to read with vidcap.read().
    fps, width, height are simply info about vidcap
    contrast (float): apply +/- 100 contrast to video
    brightness (float): apply +/- 100 brightness to video

    return: cv2 video capture of edited video file
```

skip_frames

```
    skip_frames(of, vidcap, skip, fps, width, height)

    Frame skip, convenient for saving time/space in an analysis of less detail looking at big picture movement.

    of (str): filename without extension
    vidcap: cv2 capture of video file, with all frames ready to read with vidcap.read().
    fps, width, height are simply info about vidcap
    skip (int): When proceeding to analyze next frame of video, this many frames are skipped.

    return:
        cv2 video capture of edited video file
        length, fps, width, height from this video capture
```

motionaverage

Post-processing tool. Finds and saves an average image of entire video.

```
    Usage:
    from _motionaverage import motionaverage
    motionaverage('filename.avi')
```