

# Video Processor Class Diagram

elo9@aber.ac.uk

February 21, 2017

## Contents

<b>1</b>	<b>Video Processor Class Diagram</b>	<b>2</b>
<b>2</b>	<b>Description of Class Diagram</b>	<b>2</b>
2.1	Variables . . . . .	2
2.2	Functions . . . . .	2

## 1 Video Processor Class Diagram



## 2 Description of Class Diagram

The video processing will be performed by a single class, the **VideoProcessor**. The justification for doing this in a single class, is to remove the possibility that the video object will have to be passed between classes. In addition the functionality required for video feed manipulation doesn't have many stages so can be easily represented as functions in a single class.

### 2.1 Variables

- **video\_feed**: The `video_feed` variable is of type `VideoCapture`. This class type holds a video stream and can be imported from OpenCV. To maximise the performance of the system the video object will be a global to the class object to avoid passing into and out of functions. However for test access there will be a get method for the `video_feed`.
- **video\_feed\_array**: Holds four copies of the `video_feed` that will be displayed on each side of the pyramid in the output window

### 2.2 Functions

- **init**: Python constructor for initialising the `video_feed` object to `None`.
- **begin\_capture**: Function to start the `video_feed` with specified device. This function takes an integer (referring to the device number to use - 0 is default) and sets the `video_feed` to capture the video from that device. The function has a check to ensure that the parameter is an integer and raises a `ValueError` if it is not.

- **end\_capture**: Function to release the camera feed handle and set the video\_feed variable to None.
- **get\_video\_feed**: Returns the video\_feed object.
- **copy\_video\_feed**: Creates a deep copy of a video\_feed and returns a new object containing the copy.
- **populate\_video\_feed\_array**: Calls the copy\_video\_feed 3 times and stores each object in the video\_feed\_array.
- **get\_video\_feed\_array**: Returns the video\_feed\_array object.
- **subtractBackground**: The subtractBackground function will be used to ensure that only the Actor is shown in the video feed and the background is removed (set to black (RGB(0,0,0))). Based on research stated in the Outline Project Specification, This will use simple background subtraction or moving average background subtraction. Spike solutions will be carried out during the creation of the prototype to assess if the image processing machine is capable of more complex background subtraction techniques. Furthermore, it will test if simple background subtraction techniques produce the required effect.
- **transformVideo**: The transformVideo function will duplicate the video feed into 4 identical. The feeds are rotated by 90 from one another around the centre of the display monitor. This is the format that is expected for the pyramid to produce holograms.
- **outputVideo**: The outputVideo function will add the video feed to a window and this will be displayed on the monitor in the Viewing Area to create the hologram.