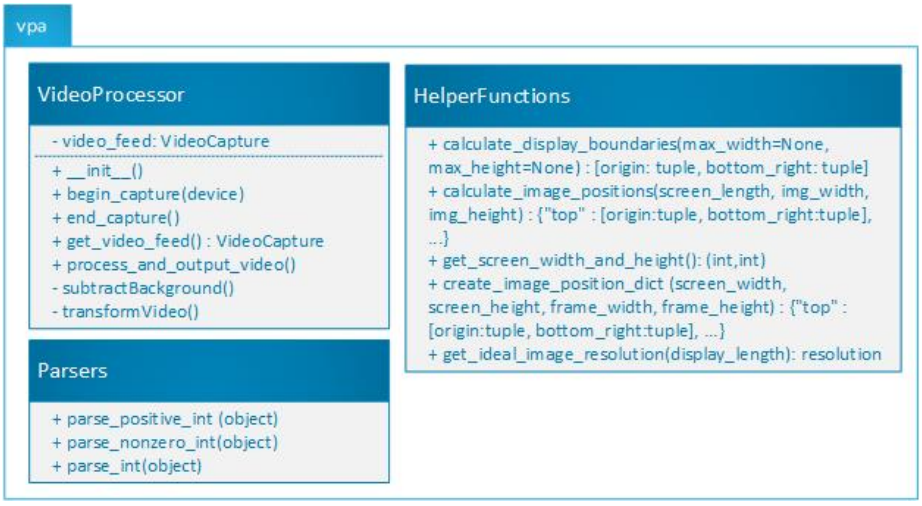
Video Processor Class Diagram

elo9@aber.ac.uk

February 25, 2017

Contents

1 Video Processor Class Diagram



2 Description of Class Diagram

The vpa (Video Processor Application) package contains two modules. The first, the **VideoProcessor** module, contains the **VideoProcessor** class which is the core of the system that will handle video capture and display. The second module will contain helper functions for the **VideoProcessor** class that do not require access to the class variables.

2.1 VideoProcessor

2.1.1 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.

2.1.2 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 is 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.
- **process_and_output_video**: The output_video function will add 4 copies of the video_feed to a window in the correct locations. This function will also handle calls to subtract_background and rotate_image.
- **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.

2.2 HelperFunctions

2.2.1 Functions

- **calculate_display_boundaries**: This function take the specified maximum width and height of the display area and creates the largest possible square that will fit. The function then returns the origin and bottom right co-ordinates of the square that will act as the maximum boundaries of the output window. If no values are provided for this function, the screen resolution is used instead.
- **calculate_image_positions**: Calculates the coordinates that each image (video frame) should be placed in. The images should be in the topmiddle, bottommiddle, leftmiddle and rightmiddle of the screen. The screen length (one require one value for this as the display area is square), image width and image height are parameters of the function. The function will calculate to co-ordinates by finding using half the screen length + or - half the height or width of the image. The function will return a map of lists of tuples containing the co-ordinates. The map will be structured such that:
  {"position" : [origin tuple, bottom right tuple],
  ... }
  An example of this is:
  {"top" : [(100,0), (200,50)],
  ... }

- **get_screen_width_and_height**: Function that returns the screen resolution as a tuple (width, height).
- **create_image_position_dict**: This is a function that controls the flow of execution. Given screen dimensions and frame dimensions, this function calls calculate_image_positions and then applies the displacement from calculate_display_boundaries.
- **get_ideal_image_resolution**: Given the length of one side of the square display area, this functions determines the maximum resolution an image should be. Images will be squares and can, at a maximum, be a third of the length of a side of the display area. This function will traverse a list of possible video resolutions and return that resolution as a tuple.

2.3 Parsers

2.3.1 Functions

- **parse_int**: Takes an object, python variable, and raises an error if it is not a integer.
- **parse_positive_int**: Takes an object, python variable, and raises an error if it is not a positive integer
- **parse_nonzero_int**: Takes an object, python variable, and raises an error if it a zero.