# How to Detect Motion using only a Webcam connected to a BeagleBone

# Introduction:

In this guide, you will learn how to detect motion with a webcam using only its image stream. The webcam does this by detecting if the image it is currently capturing differs from a previous image by a certain threshold and if so will signal that there has been motion. Completing this module can aid in implementing other features that may also require motion detection.

**NOTE:** All of the following code will be written in C.

# Pre-requisites:

1. Have a webcam with a proper connection to the BeagleBone
2. Have a working program that gets MJPEG frames from the webcam.
    a. There are many student guides from previous years that will guide you to do this. See the troubleshooting section for links.
    b. There is also this example file you can refer to help you. This guide will require data received in the params just like the process_image method of that example.
3. Have a method ready to process the MJPEG frames, should have an unsigned char pointer to the frame data, and an int hold its size.
4. Have a global variable of the type **uint8_t ***, initialized to NULL, outside the scope of your process_image method. In this guide, we will refer to that variable as *prevFrameData*.
5. Have a global variable or definition called **THRESHOLD** storing the value 30.
6. **MUST BUILD THIS CODE ON TARGET**

# Section 1: Install Dependencies:

1. On your BeagleBone run the following command:

```
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libavutil-dev
```

## Section 2: Writing the code that needs to be run for each frame that is picked up by the webcam:

1. Initialize the Codec and CodecContex:

```
// Create MJPEG codec
AVCodec *codec = avcodec_find_decoder(AV_CODEC_ID_MJPEG);

// Create the context
AVCodecContext *codecContext = avcodec_alloc_context3(codec);
```

2. Open the Codec:

```
if (avcodec_open2(codecContext, codec, NULL) < 0)
{
    printf("avcodec_open2 error\n");
    exit(1);
}
```

3. Create a packet from the mjpeg data you have so it can then be converted to RGB values:

```
AVPacket packet;
av_init_packet(&packet);
packet.data = malloc(sizeof(unsigned char) * size);
memcpy(packet.data, p, size);
packet.size = size;
```

4. Send the pack in the codec we created earlier:

```
// Send packet to avcodec to it can decode with the provided codec
if (avcodec_send_packet(codecContext, &packet) < 0)
{
    // Handle error
    printf("send packet error\n");
    exit(1);
}
```

5. Allocate a Frame variable so we can receive the decoded frame:

```
// Alloc frame
AVFrame *frame = av_frame_alloc();

// Receive the frame from avocdec and now we have the decoded data!
if (avcodec_receive_frame(codecContext, frame) < 0)
{
    printf("receive frame error\n");
    exit(1);
}
```

6. Extract the data that is needed from frame:

```
uint8_t *frameData = frame->data[0];
int frameDataSize = frame->linesize[0] * frame->height;
```

7. Check if prevFrameData has data, if not, give it data. Else proceed to step 8.
   (not sure what prevFrameData is? Read Pre-reqs of this document)

```
if (prevFrameData == NULL)
{
    prevFrameData = malloc(sizeof(uint8_t) * frameDataSize);
}
else {
    // Motion Detection. Continues in step 8 of this guide
```

8. Now we begin detecting motion. First, let's count the number of pixels that differ from the previous frame (prevFrameData) and the current frame (frameData) that differ more than THRESHOLD pixel points:
   (Note: this code would be part of the else block of the if statement in step 7.)

```
    int differentPixels = 0;
    for (int i = 0; i < frameDataSize; ++i)
    {
        uint8_t prevPixel = prevFrameData[i];
        uint8_t curPixel = frameData[i];

        int diff = (int)((int)curPixel) - ((int)prevPixel); // get difference
        diff = diff < 0 ? diff * -1 : diff; // get abs
```

```
        // check diff
        if (diff > THRESHOLD)
        {
            differentPixels++;
        }
    }
```

9. Check if a significant portion of the image was different. In this guide we will consider motion being detected if more than 5% of the pixel were different.

```
    // if more than 5% of pixels are different then motion!
    if (differentPixels >= frameDataSize * 0.05)
    {
        // Motion! Do stuff here as you please...
    }
    else
    {
        // No motion...
    }
```

10. Lastly, copy the current frame (frameData) into the previous frame (prevFrameData) so when the next frame is processed, we can refer to the current frame as the previous frame.

```
// set current frame as prevFrame, so it can be prevFrame for the next frame.
memcpy(prevFrameData, frameData, frameDataSize);
```

## Troubleshooting:

1. Ensure that you are installing all the necessary libraries and building the project in the target environment, not the host.

2. Make sure that all the required libraries, such as libavcodec-dev, libavformat-dev, libswscale-dev, and libavutil-dev, are installed correctly on your BeagleBone. You can check the installation status of these libraries by using the 'dpkg' command.

```
(target)$ dpkg -l | grep <library_name>
```

3. Before running the application, make sure that the webcam is correctly connected to the BeagleBone board. You can use the ‘lsusb’ command in the terminal to confirm that the board recognizes the webcam.

```
(target)$ lsusb
```

4. The guide assumes that you have already obtained the video frames from the V4L2 API on the BeagleBone board. If you require assistance with capturing video from V4L2, you can refer to the student's previous how-to guide on capturing videos from a webcam for instructions. Here are a few helpful examples:
https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2022-student-howtos/DetectMotionUsingUSBWebcamAndOpenCV.pdf
https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/links/files/2015-student-howtos/RecordingWebcamVideos.pdf

## References:


1. http://ffmpeg.org/documentation.html
2. https://ffmpeg.org/doxygen/trunk/index.html