**Flickering Detection (Contd...)**

* **The above Technique was written in C++ Language and implemented using OpenCV Library.**
* **IDE used was Code Blocks.**

**Flickering was detected in each frame by analyzing that, if video containing frame flickers, means that it has area bounded in form of rectangle. Bounded area was covered by the pixels having sum of their BGR values less than threshold value that is 225.**

**Area was detected by finding the lower ith row and the upper jth row, sharing same width as that of the image and having pixels that have r+g+b <= 225. Threshold area for detection taken is 15 X width of image i.e. 15\*150 = 2250.**

* **First the given video was divided into 781 frames by executing the file “convrt.cpp” and each frame was labelled as “frame\_%d.jpg”.**
* **Each frame was then executed separately by applying above measures to it. Later I merged the old and current frame to a new frame.**
* **Resulting new frame depicts whether the given x frame flickers or not.**
* **From Output frames that was analyzed, the resulting output Video was made my running the code “make.cpp”.**

**Input: “flicker.mp4”**

**Output: “flicker\_myresult.avi”**

**Instructions:**

1. **Compile the code using gcc/g++ compiler.**
2. **Run first “covrt.cpp” by having result video in same directory.**
3. **Then run “detect.cpp”, to store the current image.**
4. **Finally run the “make.cpp” to create a final video with step3 frames.**

**Conclusion:**

1. **Blob Detection Technique can also be applied to this problem, but due to shortage of time I implemented the above one. Heading towards implementing that also and submitting the next report soon.**
2. **After Strictly analyzing the Prev. code, I compressed the frames and then found the Output Video from all set of 781 Frames.**
3. **Threshold area was also included in this Code, as that was not implemented in Prev. Code.**