**IBM-Technion Airborne camera project, for detection of movements in recorded video**

Selected article for starting implementation is “***fast video object segmentation technique***”:

<http://calvin.inf.ed.ac.uk/wp-content/uploads/Publications/papazoglouICCV2013-camera-ready.pdf>

our code located in :

<https://github.com/surferran/AirBorneCamera_C_code/>

**detailed algorithm scheme for implementation:**

**optical flow (Dense)**

between subsequent frames

**Section 3.1**

The flow is :

is not accurate because we cannot differ the source is true boundary or noise.

The last element is wrong when the flow is slow. Or static. So it’s not reliable.

Total criteria can be :

while T is 0.5

. It includes open boundaries. And errors from wrong optical flow inputs.

----

Part of voting :

More then 4 intersections of rays with motion boundaries curves – the pixel is inside the object.

It uses ‘integral intersections’ based on integral images concept.

4 matrices of

(0,0) pixel (and 0 index rows/columns) are assumed as background. Rays of 0,45,90 deg are starting from there.

Odd number of cross-sections of motion boundaries bt the ray – meaning that the ray origin point is ‘inside’.

X as number of intersections, for pixel p(x,y). and:

: inside-outside map, for each time-frame ‘t’.

-----

Calc optical flow – run for all frames of video (option : store as Mat files).

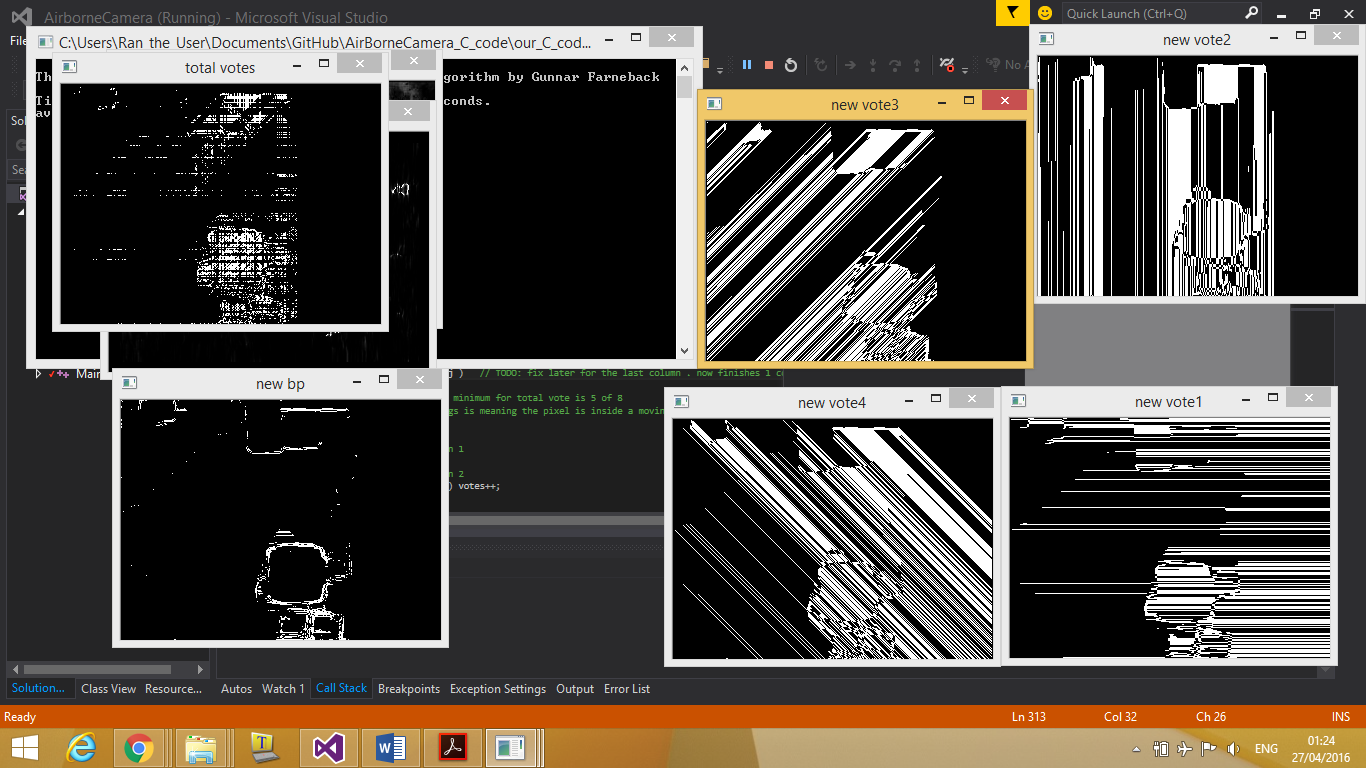
SuperPixels – run for all frames of video

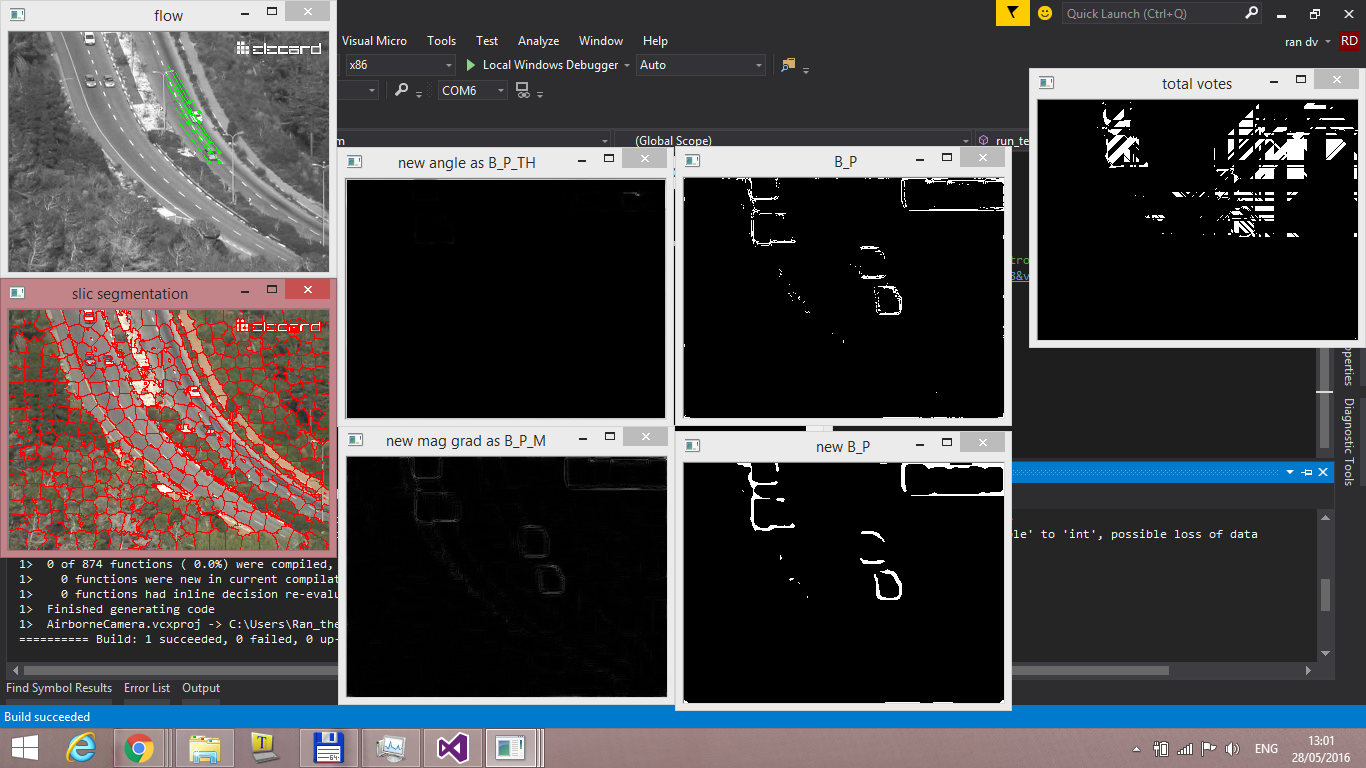
Calc Bpm

Calc BpTheta

Calc Votes

Calc Mt





----

Next:

Implement build of cost function matrices.

Potentials matrices:

V,W as spatial and temporal smoothness, across 1 frame, or 2-frames, respectively.

///////

Checks:

Keeping each frame phase as Mat files : flow, bp, totalVotes.

Next is merging all flows to 1 Mat to use as input for the Matlab code of the article.

Saving their frames as separate Mat files, in those check points. And compare.