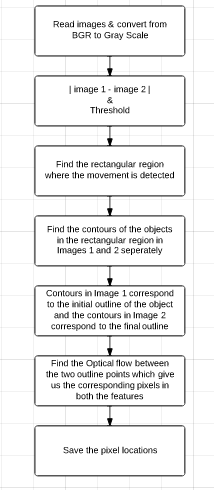
Programming Exercise

**Notes:**

The following exercise was done using OpenCV 2.4.6 and Visual C++ 2010. The flow chart below shows the steps used in the process.



The First image displays the rectangular region where the motion was detected using subtracting frames. The second image displays the initial and final outlines of the object that moved and the third images iterates and displays the corresponding points on the outline.

The motion is characterized as a vector in pixels. It is stored as a vector pointPositions of class objects motion which contain two cv::Point s representing the initial and the final pixel locations.

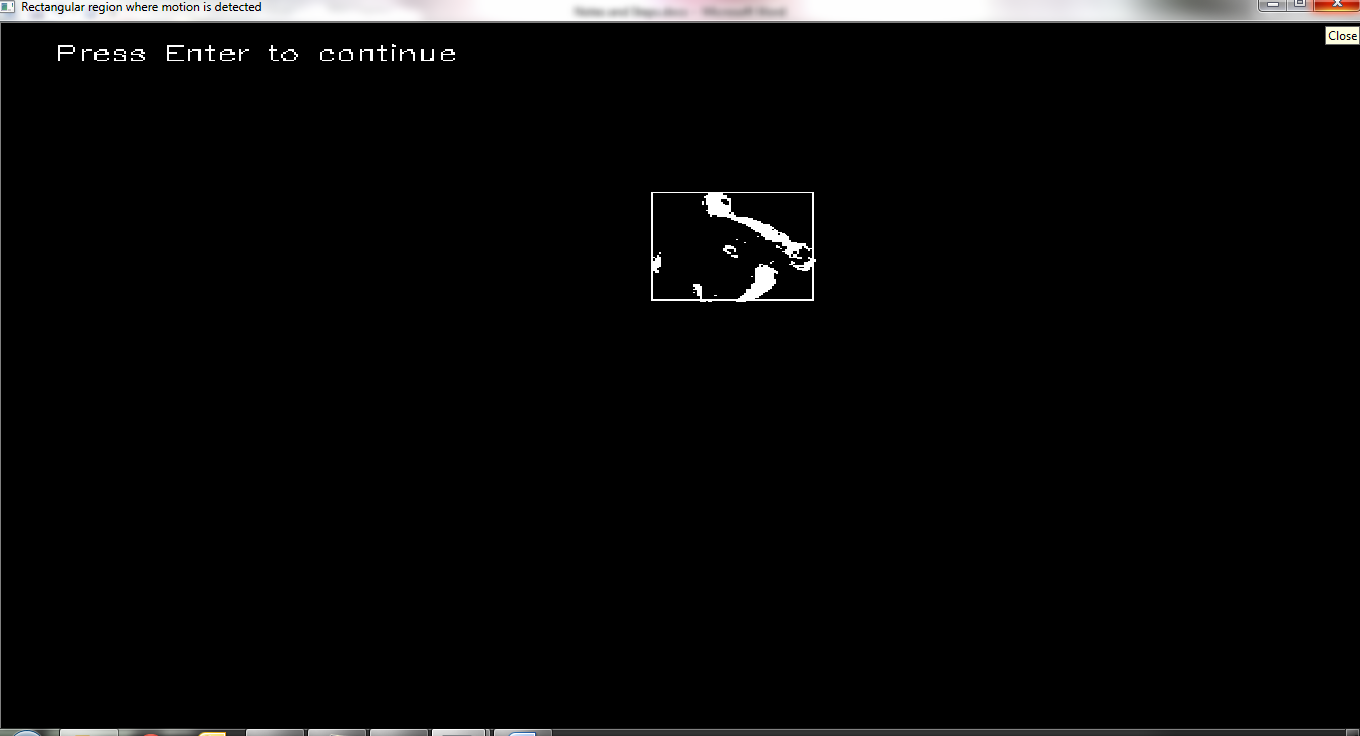
Assumptions made:

1. The object being moved is not very far from the initial position
2. The background where the object was moved is not very clumsy

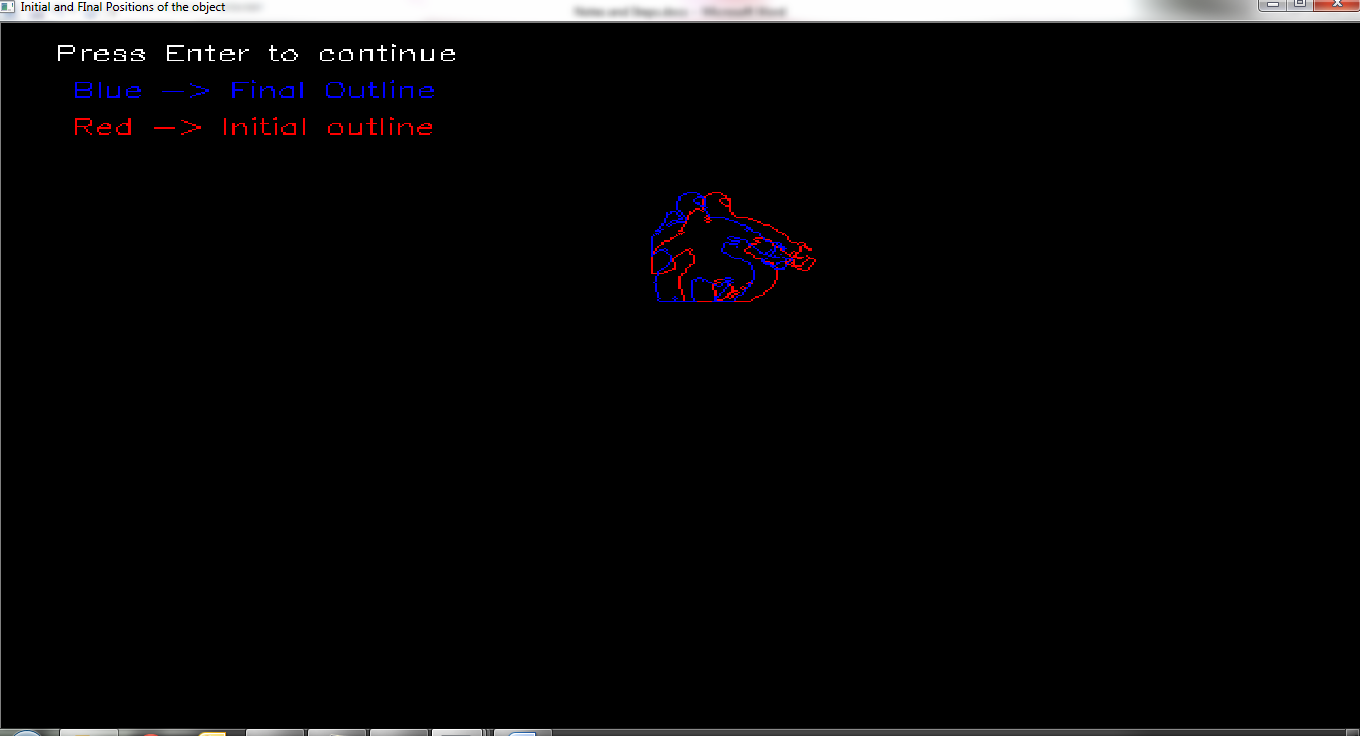
**STEPS TO RUN THE PROGRAM:**

1. Unzip *Motion Detection\_VTTI\_Amoghavarsha.zip*
2. Open the folder *Motion Detection\_VTTI*
3. To run the program
   1. Run *Debug/Motion Detection\_VTTI.exe*
4. To view the source code
   1. Open *Motion Detection\_VTTI/Motion Detection\_VTTI.sln*
   2. Press *Run* button
5. To make changes in the code or load different set of images
   1. Images should be stored under *Motion Detection\_VTTI/Debug*
   2. Make changes in the lines 6 and 8 (with appropriate image names) of *Motion Detection\_VTTI/Motion Detection\_VTTI/main.cpp*
6. Run the program under Visual studio once for the changes to show effect in the .exe file

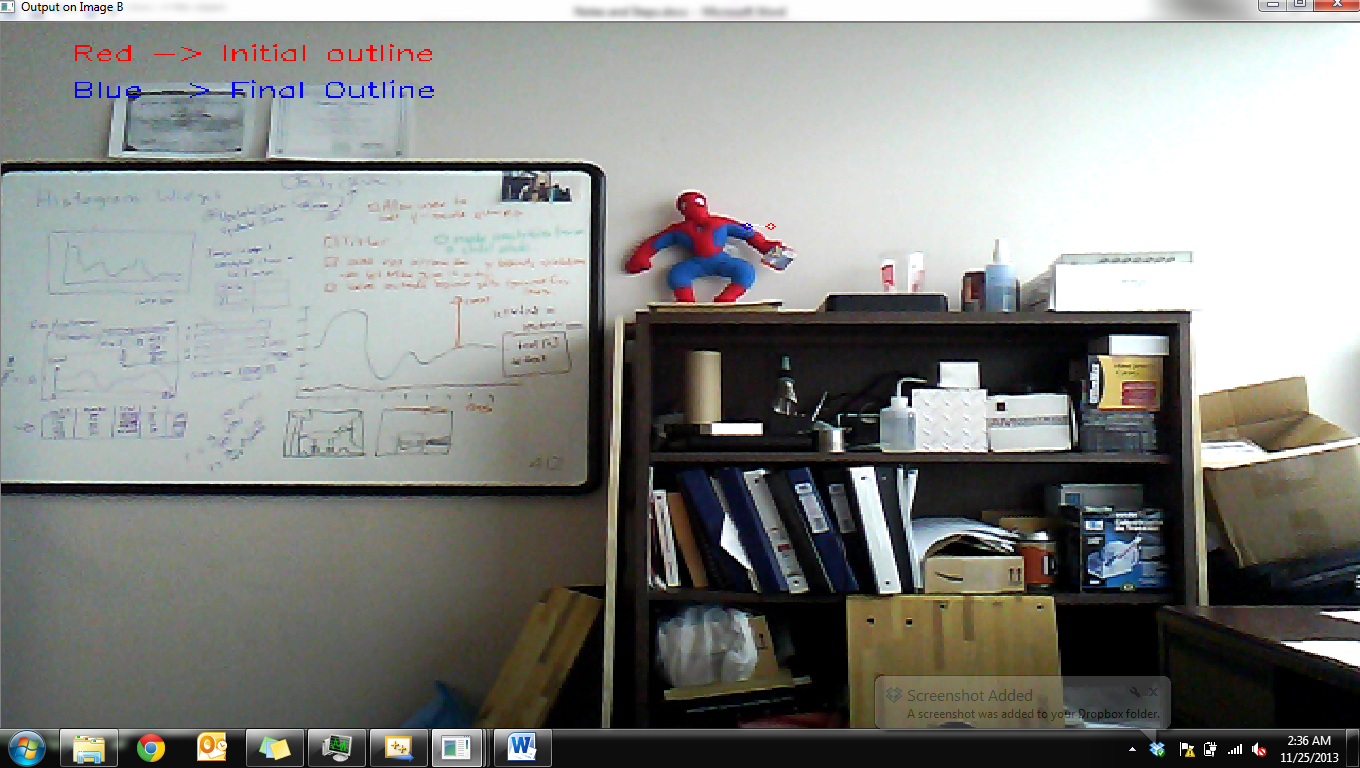
**Results:**



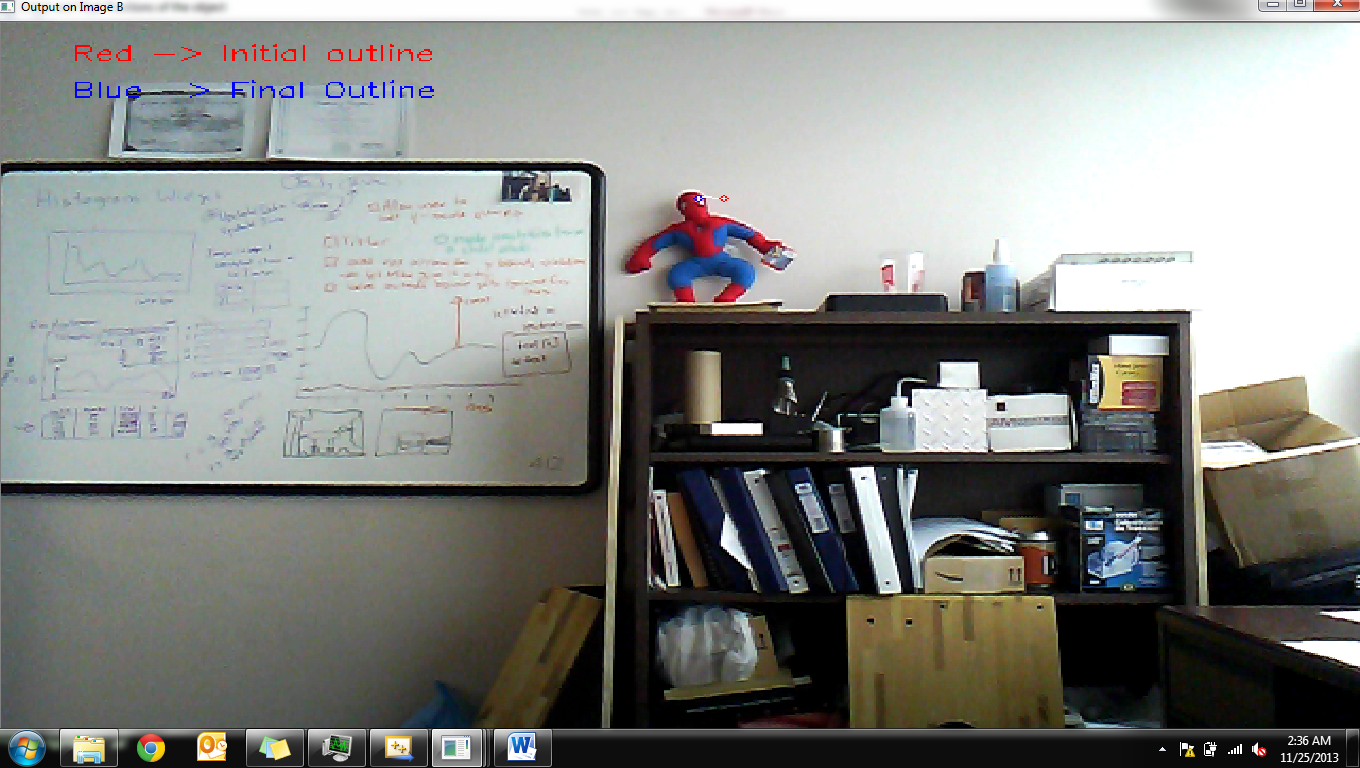
*Fig 1. Subtracted image with a rectangular region around it*



*Fig 2. Image displaying the initial and final outlines of the object being moved*



*Fig 3. One instance of image showing the initial and the final pixel location*



*Fig 4. Another instance of image showing the initial and the final pixel location*