A. A word document, named HW2.doc. The word document should include:

a. Your names.

b. Answers to all questions.

c. A short description of what you implemented in each of part of the HW.

d. Pointers to the output videos with the image names, the parameters used, etc.

e. A few words on how to use your functions.

B. Your documented code. In addition, a script that demonstrates how to invoke the functions with the appropriate parameters.

Ben Menachem  
Nir Barzilay

**HW2**

**Question 1:**

* Short description of the implementation: (nbs\_tracking.m)

We implemented naïve tracking algorithm based on the NBS from HW1. On the binary output of the NBS we do some holes-filling & noise reduction, the result is that only big white blobs are drawn onto a black background. We compute bounding-boxes [bboxes] around the blobs in each frame. If in two consecutive frames, the same bboxes of the same blob are overlapping, we assume that this is the same blob, and draw the bbox around it.

In addition, we keep for each tracked blob a single-bbox-color through all the video.

* How to use:

The algorithm itself is in the file "track\_nbs.m". This is the matlab function that executes the algorithm.

The algorithm accepts a "BlobSize" parameter which points what's the minimum size of an object to be tracked. This is one of the noise-control techniques.

There's a script called "alg\_exe\_script.m". This is our entry point for the execution of the algorithm.

* Answers to the questions:

1. We controlled detection of significant blobs only, by setting the "BlobSize" parameter. We also used 2 MatLab functions: bwareopen(…) for additional noise recution and imfill(…) for emphasizing filtered blobs.
2. With the BlobAnalysis tool we could compute bounding-boxes around blobs.
3. The blobs-overlapping technique is in the file "BoundingBoxOverlap.m":
4. We chose the second option. Before looping over the video frames, we calculate bbox for the first frame, in which the pedestrians are not detected = no pedestrians = no overlapping is found.
5. Quality report:

* The following 6 images are snapshots of 60 consecutive frames (movement from left to right):

****

As can be observed, the pedestrian isn't big enough for the algorithm to track after him. One snapshot spotted the guy as big enough blob, but later it detected him as a small blob again. We could detect the pedestrian earlier if we would set the BlobSize parameter to a smaller value.

* However when the blob became big enough, the algorithm tracked him as a single blob (movement from left to right):



* In the following frames, another pedestrian comes in and is detected as 2 different blobs, **with the same BlobSize** **as before**, but then again as a single one (movement is from right to left):



* And also the clouds are detected as blobs at some point:



To sum the quality of the algorithm up: In general, the smaller the BlobSize is, a pedestrian will be divided into more blobs & more non-pedestrian-blobs will appear (false positives). The bigger the BlobSize is, small pedestrians won't be detected (more false negatives), because the algorithm ignore small blobs (which can be an actual pedestrian moving in the Z axis).