MVA – Object Recognition and Artificial Vision

Assignment 2

Stitching Photo Mosaics

Jean-Baptiste Fiot

Nov 2008

1 Algorithm outline

1.1 Detection of Harris Points

I used the function **harris.m** given, and implemented a function **LessHarrisPoints.m** to reduce the number of Harris points. This function reduces the overall computing time, and is based on a threshold “harris_threshold”. When the bias/gain-normalized value is above this threshold, the harris point is kept.

1.2 Extraction of a feature descriptor for each feature point

The extraction is done by **GetDescriptors.m**.

For each point, I extract a 45*45 window centered on this point (zero-padded if need be). Then I take the mean of each 5*5 block. Finally, I get this 9*9 matrix, and bias/gain-normalize it.

1.3 Matching of feature descriptors

The matching is done by **MatchDescriptors.m**.

The distance between two descriptors is given by the trace of the output matrix of **dist2.m**.

$$\boxed{if\ \frac{d(point \in I1, closest\ point \in I2)}{d(point \in I1, second\ closest\ point \in I2)} < \varepsilon_{descriptors}, then (point \in I1, closest\ point \in I2)\ match.}$$

1.4 Robust estimation of homographies using RANSAC

The homography estimation is done by **ComputeH.m**.

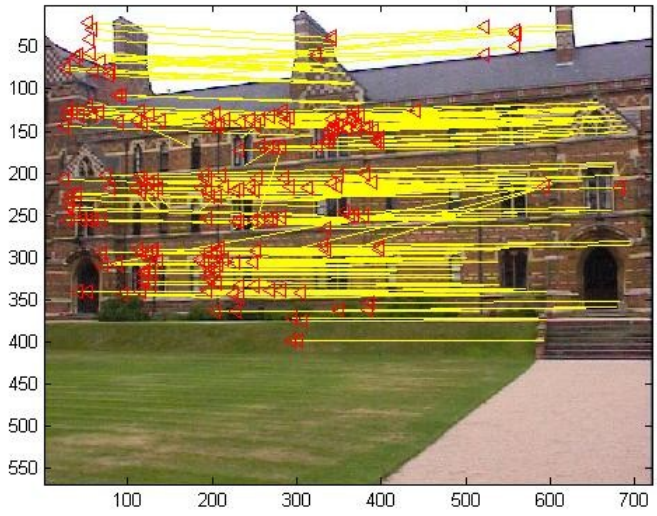
- For n iterations :
  - I get 4 random corresponding points
  - I estimate the homographies given these 4 correspondences.
  - I count the number of inliers, given that :
    - $\boxed{\|H(point \in I1) - corresponding\ point \in I2\|^2 < \varepsilon_{Ransac} \Rightarrow point \in I1\ inlier}$
- I re-estimate the homography with the inliers of the best homography (ie the one with the highest number of inliers).

1.5 Warping and compositing

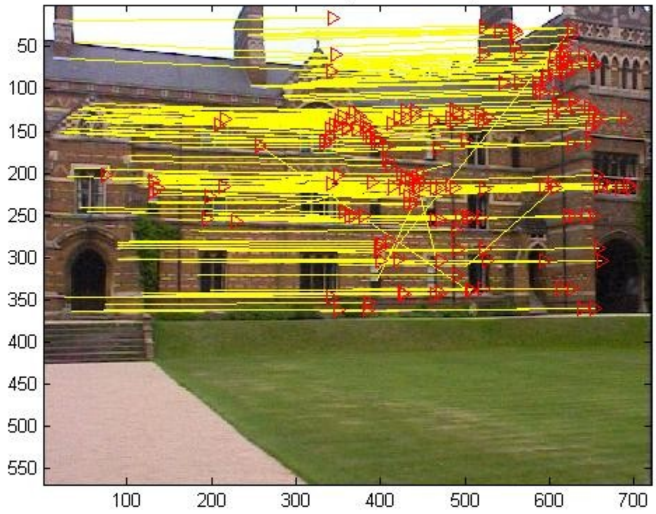
The warping is done by the given function **vgg_warp_H.m**.

2 Example with Keble College pictures (in Oxford)

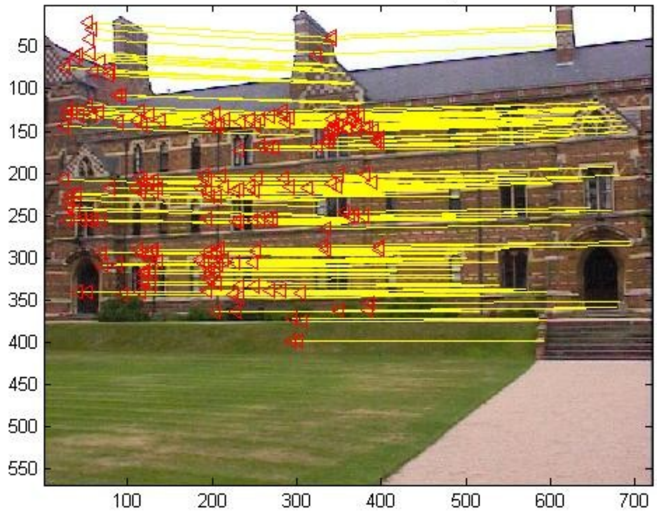
Matches between left and center pics



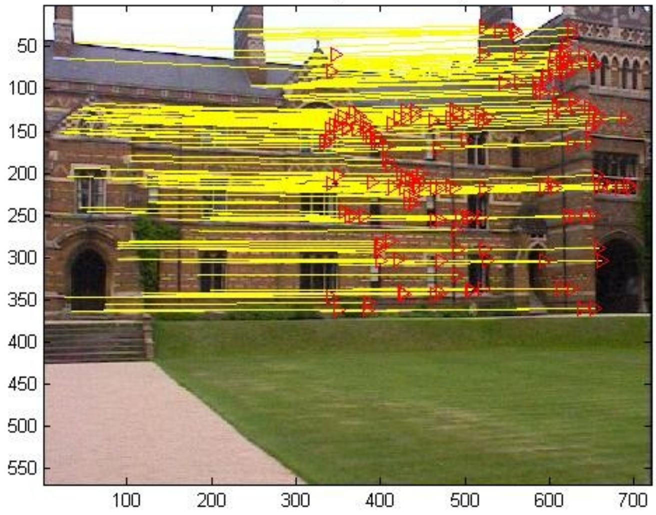
Matches between right and center pics



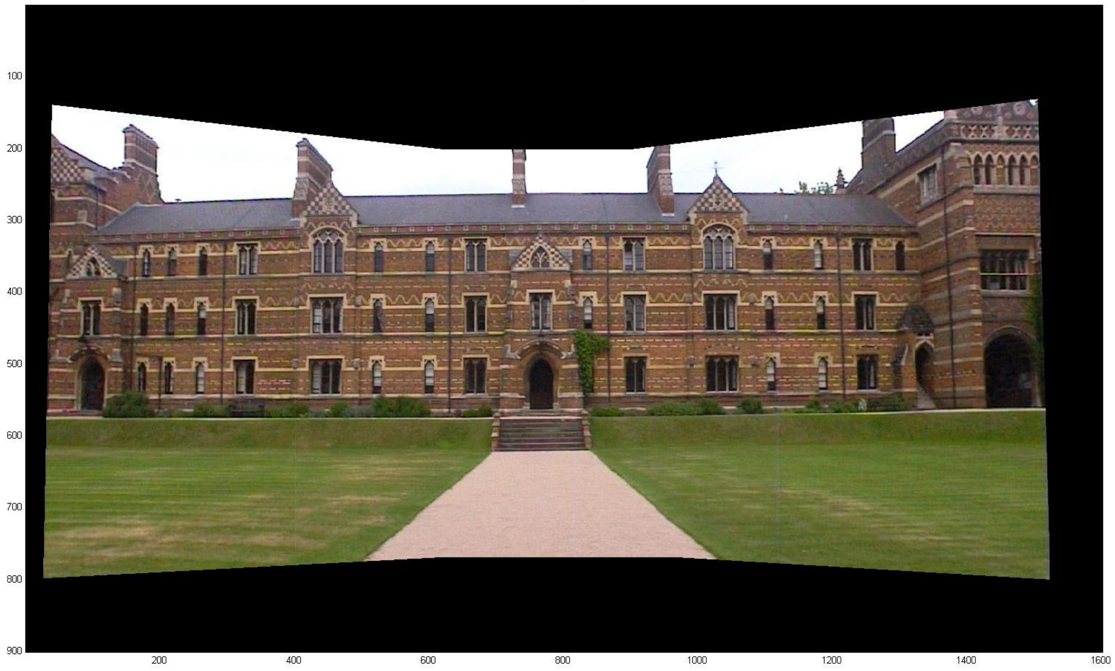
Inliers between left and center pics



Inliers between right and center pics



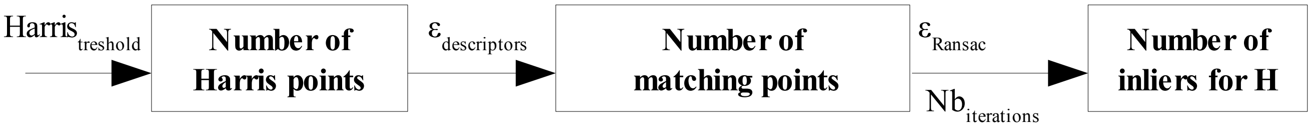
Such an awesome panorama!!



3 Parameters and limits of the algorithm.

To get good results, we have to set the parameters so have we find an homography with a reasonable number of inliers for the homography (ie a "pretty good" homography).

We have :



with Number of Harris points > Number of matching points > Number of inliers for H.

For Keble, this set of parameters works well:

harris_threshold = 1.25;

epsilon_descriptors = 0.7;

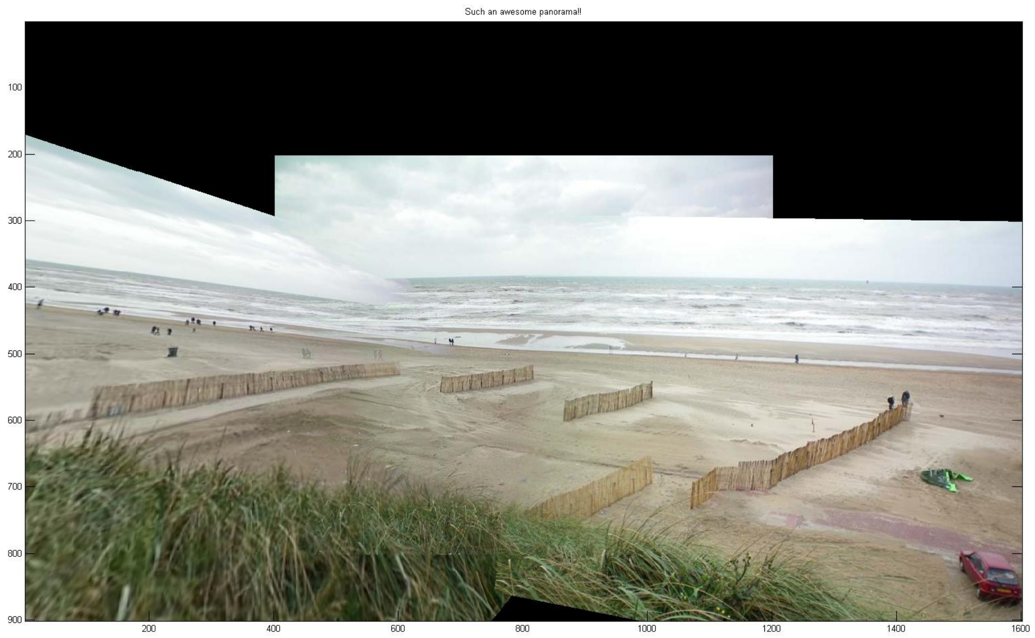
epsilon_RANSAC = 10;

nb_iter = 3000;

It gives :

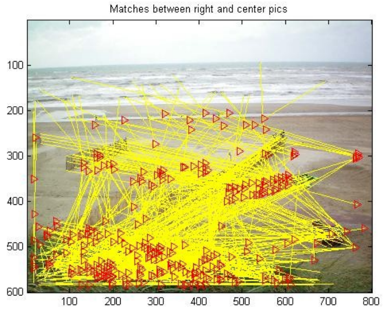
- Left : 441 harris points.
- Center : 473 harris points.
- Right : 428 harris points.
- Center and Left : 214 matches.
- Center and Right : 195 matches.
- H_C2L with 186 inliers.
- H_C2R with 173 inliers.

- With the same parameters, the set of pictures Zandvoort6, can have bad homographies. For instance, H_C2L has **only 6 inliers**, which leads to a pretty bad result.
- This can be explained by the low number of "interesting points" in these large areas of sand, sea, and sky.
- To solve this problem, we increase the number of harris points used (by **descreasing the Harris threshold**), and we **increase the number of iterations** in the RANSAC algorithm, to find a **better homography**.
- With these changes, we get enough inliers (thanks to the fence on the picture mainly), and **this fixes our problem** ☺(see next page).

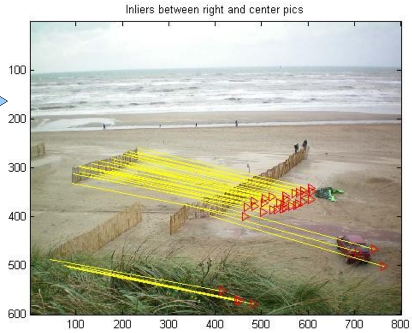


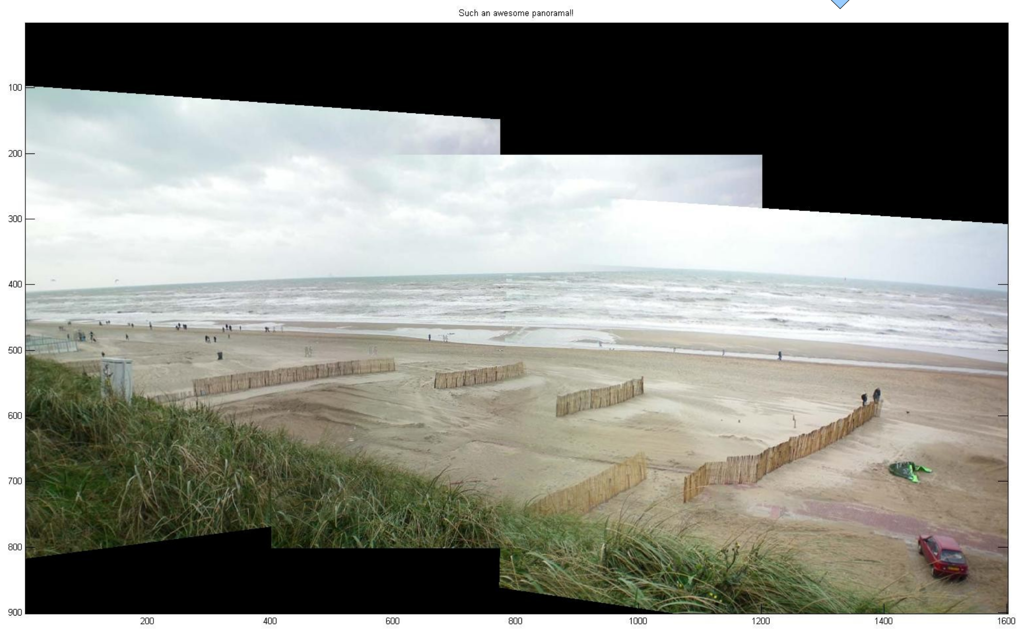
Bad result on the left side due to a number of inliers too low.

Harris$_{\textbf{treshold}}$ **lower** (=0.5 eg)



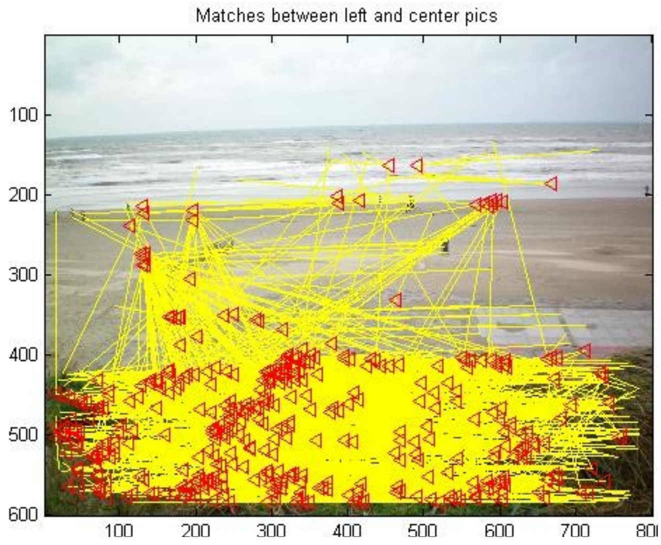
Nb$_{\textbf{iterations}}$ **higher** (=15000 eg)

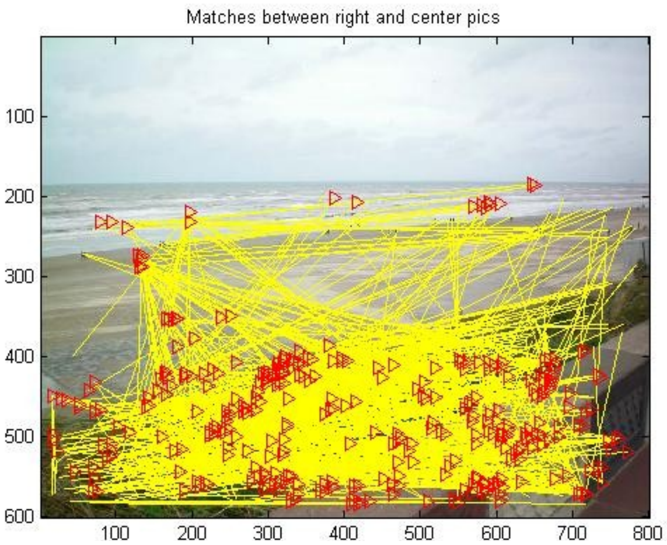


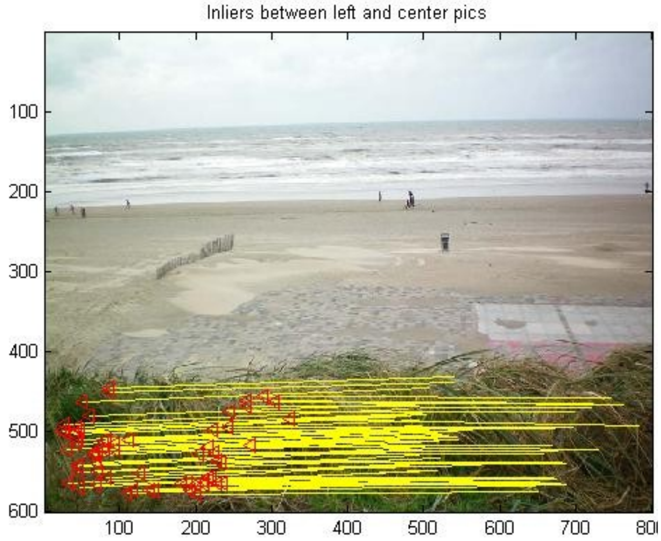


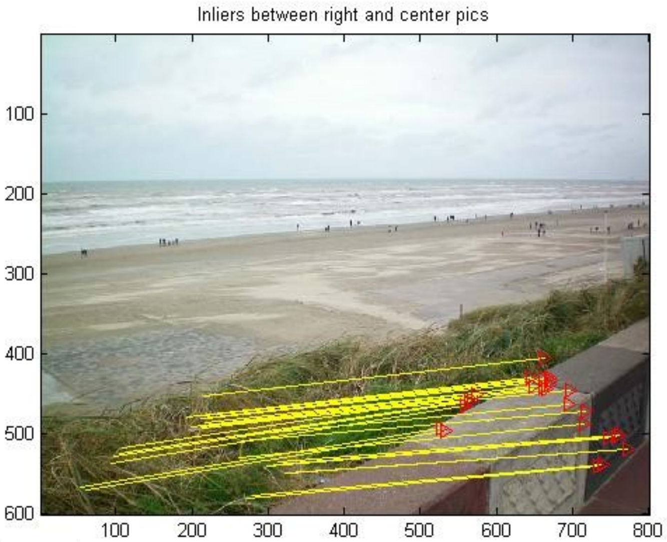
Now we have a good result. ☺

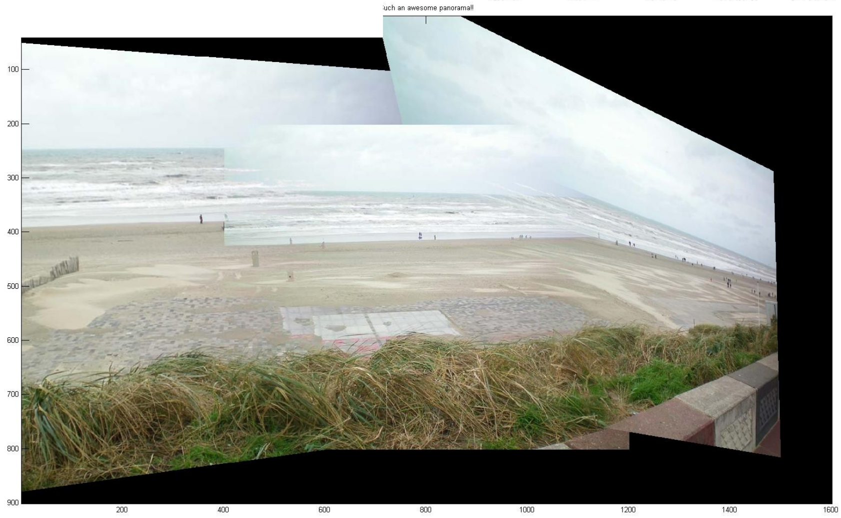
- However, for pictures with no interesting points at all, except in a texture area, this gives a bad result.
- Indeed, even if we change the parameters as above to get tons of matches, we get only a couple of inliers in the textured area (area subject to repetitions).
- For instance, on the Zandvoort5 set of pics, the algorithm finds homographies with inliers only in the grass on the bottom of the pictures.
- Consequently, the result is not really good.







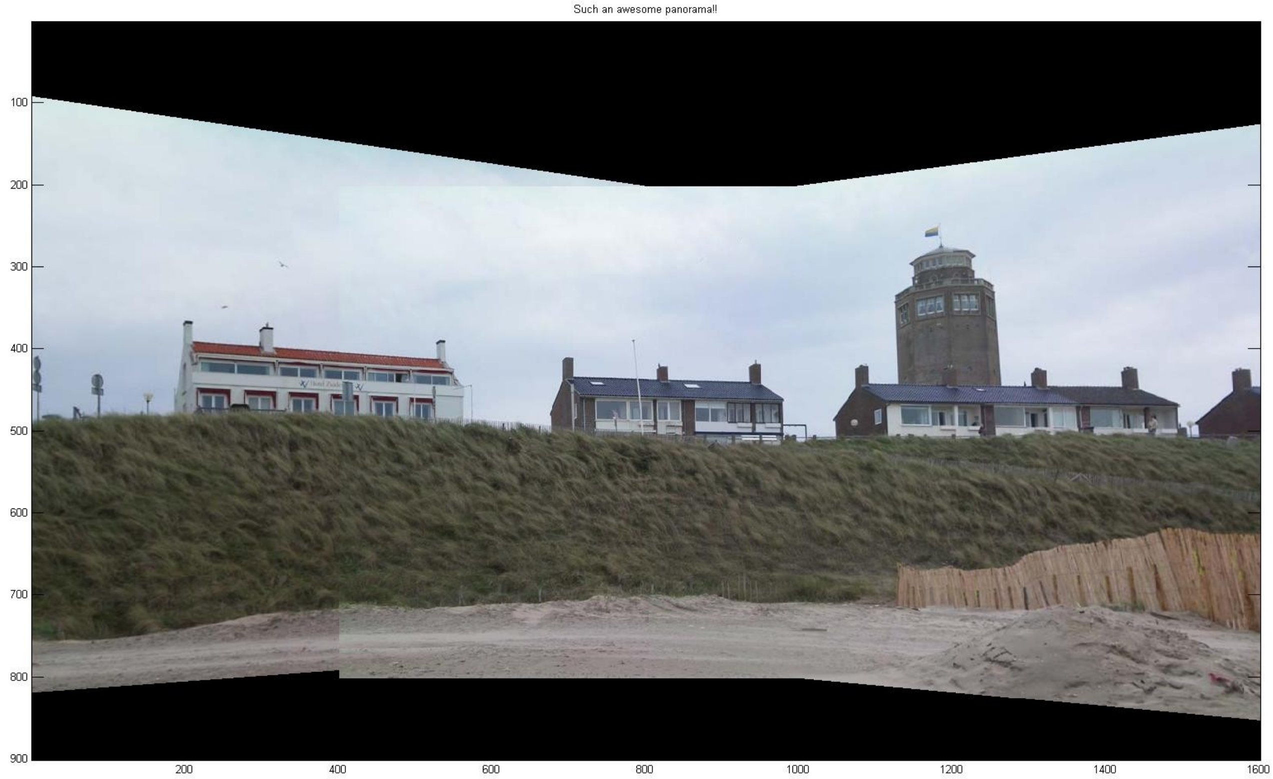


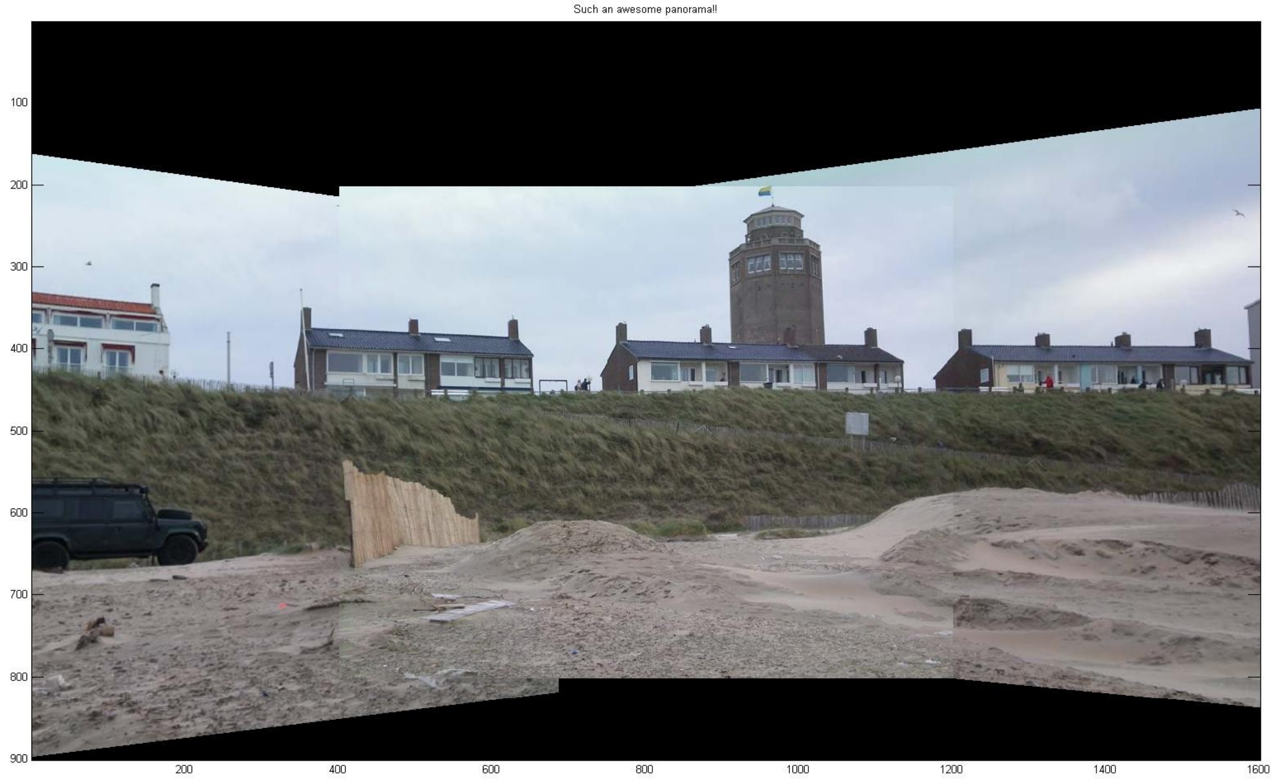


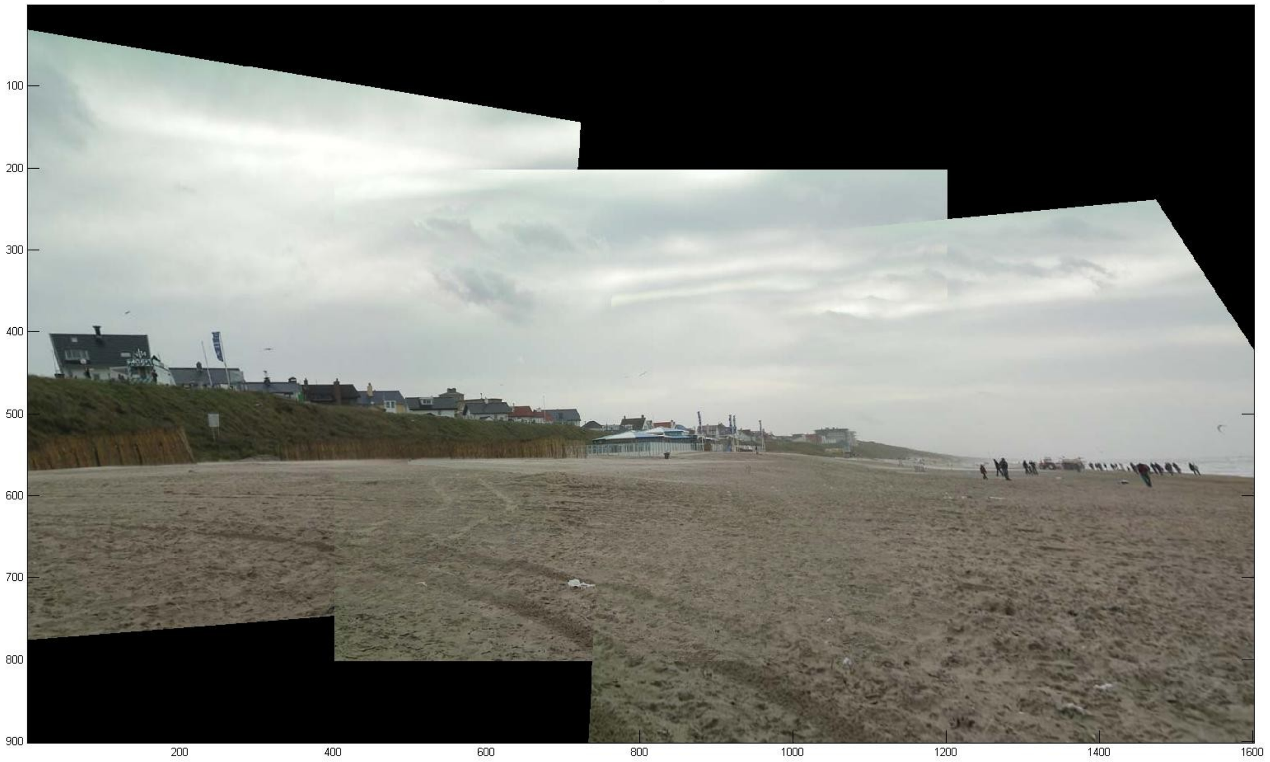
Bad result given the few inliers just in one textured area.

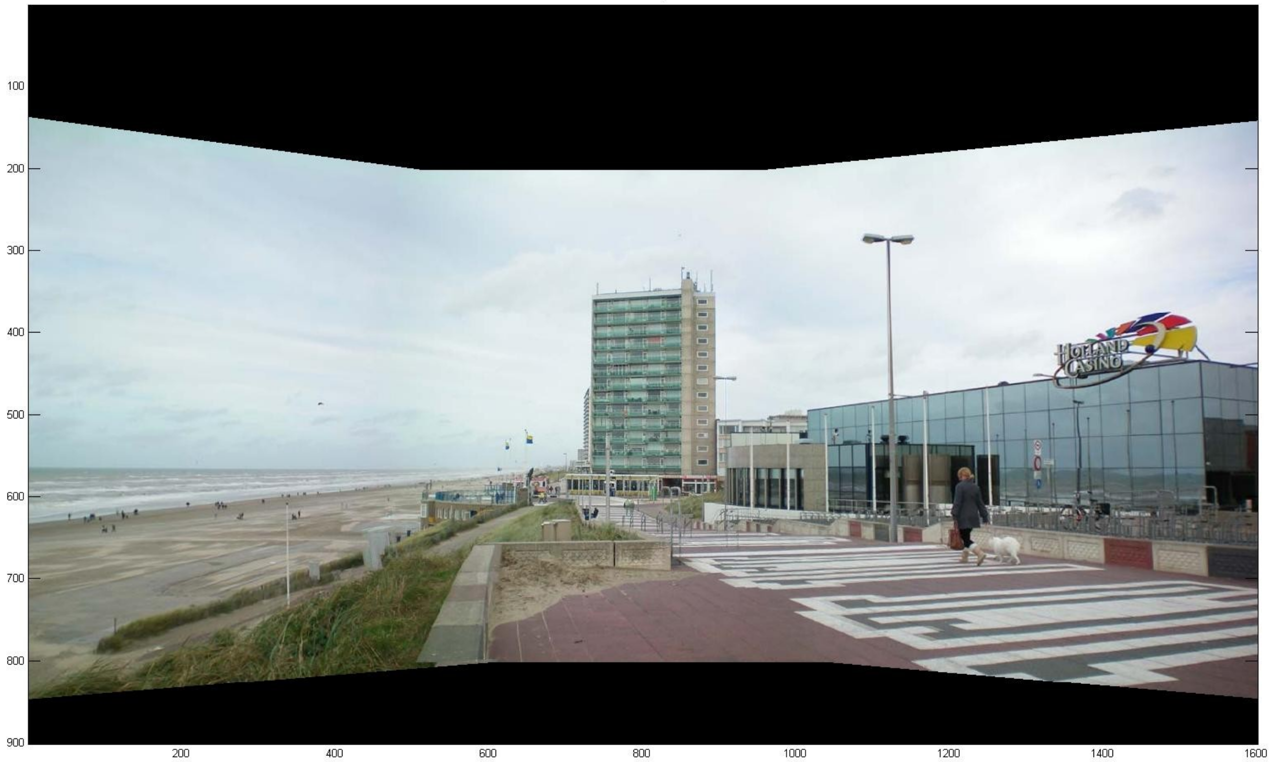
4 Other awesome panoramas!

Here are a couple of panoramas from pictures from Zandvoort, in the Nederlands.






Such an awesome panorama!!


Such an awesome panorama!!