Product Recognition on Store Shelves

Marco Boschi – 0000829970

A.Y. 2018–2019

`https://github.com/piscoTech/ProductShelves`

1 Multiple Objects-Single Instance Detection

The first task is to detect a single instance of multiple objects in each of the proposed scenes, which can be done using just a local invariant feature detector and then matching the key-points of the scene with those of the different models to be found.

The chosen detector is SIFT paired with a Flann based matcher using KD-trees, as provided by OpenCV. As the models are the same for multiple scenes their key-points are learned a single time at startup to speed up the program, also the image itself is not useful and so dropped, just the key-points and respective descriptors are needed alongside the product number and dimensions of the model.

For each scene the key-points are computed and then are matched with those of each model and the found matches are filtered using Lowe's ratio test to keep only the good ones and if enough are found the matched key-points are used to compute an homography from the model to the scene using RANSAC. The homography allows to find the center of the product and its corners, which are used to highlight the product in the scene and also compute width and height of the box as the average length of the two corresponding sides.

This basic setup however has some problems with very similar products. To overcame this limitation all key-points of the scene that have been matched with a product are discarded and not used for the others which makes the program more able to distinguish similar products with instances no longer overlapped, but not always, so instances at the same position of previously found products are detected they are ignored. Further improvements can be obtained by choosing the right order for products.

2 Multiple Objects-Multiple Instances Detection

The basic structure used in the previous environment is not suited at all as matching key-points between scene and models allows to find only a single instance of each object. To overcome this limitation the generalized Hough transform has to be used combined with SIFT key-points to find correspondences between scene and models.

Like before the scene key-points are matched against the key-points of each model (learned a single time at startup), but now the found matches (still filtered by Lowe's ratio test) are not used to directly find an homography using RANSAC, but GHT is applied considering the center of the model image as the reference point to compute the joining vectors, which are computed a single time at startup along side the key-points.

The joining vectors are applied to the corresponding scene key-point to cast a similarity invariant vote by applying to each vector the transformation of the corresponding key-point. The considered transformation is only a scale based on the characteristic scale of the key-points, ignoring the rotation because the model will always appear upright in the scenes under analysis. Applying also rotation however proved to be counterproductive as even the models with the best results had their votes more scattered in the accumulation array, probably due to almost perfect matches between key-points in similar positions but not exactly the same which however are not a problem for RANSAC when finding the homography.

The obtained AA is then processed by NMS and finally thresholded before looking for possible reference points. To obtain a quite detailed AA the quantization of the image space cannot be too coarse so even if the votes are consistent they are not aggregated in a single point, so votes are aggregated a second time based on distance between the hypotheses for the reference points with an algorithm similar to DBSCAN (a density based clustering algorithm).

Each of the clusters thus found is a possible instance and if the overall vote count for the cluster is above a certain threshold it is considered good and the matches are used to compute an homography and size and position of the instance are reported like in the single instance case.

RANSAC finds the homography by choosing only the most fitting matches and some are discarded. Because is possible that votes for different instances are clustered together if they are too spread apart, if the amount of matches ignored by RANSAC is above a certain threshold the process is repeated with only these points to find a second, third... homography and so more instances.

3 Considerations About Errors

In both single and multiple instance detection some products are missed and not detected even if they are there. This can be certainly improved if the scene images were at a higher resolution and not slightly blurred, which combined with the fact that some model provided for the products are not the same as those found in the scenes (i.e. containing or not an advertisement in a corner) results in few good matches found.