Detection of 3D objects with a multiple-view keypoint model from item images

Domen Rački[1], **Luka Čehovin**[1], **Matej Kristan**[1]

[1] *Faculty of Computer and Information Science, University of Ljubljana*
domen.racki@*gmail.com,* {luka.cehovin, matej.kristan}@*fri.uni-lj.si*

Abstract

We present a keypoint based multiple-view model approach towards 3D object detection. Our model addresses the shortcomings of other keypoint based detection methods, since they do not take into account the 3D nature of everyday objects and thus rely on frontal object occurrences. For each 3D object, a model is learned from multiple view images of the object. The learned model is used to perform object detection from an arbitrary viewpoint of the object. The proposed model is a keypoint based detector with added filtering mechanisms to ensure a robust object detection. We evaluate our proposed method on a real world dataset containing everyday grocery items with changes in illumination, object viewpoint and position, cluttered scenes and multiple object instances.

1 Introduction

Object detection has a wide range of applications in specialized as well as everyday computer vision tasks, like image retrieval systems [2, 10], everyday object recognition [6, 4, 9] and automated pick and place systems [7]. However most approaches [6, 9] do not take into account the 3D nature of everyday objects and thus fall short of performing the detection task under perspective or even different viewpoints of the object from its different sides. We address this problem by constructing a multiple-view object model from different viewpoints of the object. In this sense, our proposed method requires a minimum number of images, enough to cover the entire object in order to build an object model. For the object detection task itself, regular household items are well suited since these objects often contain repeated patterns, logos and text which makes them ideal for keypoint based detection methods. We evaluate our method on a challenging real world dataset [4], containing everyday grocery items in scenes with strong illumination and viewpoint changes, occlusions, clutter, and examples containing multiple instances of the same object.

The remainder of the paper is structured as follows. In Section 2 we provide a short description of our proposed model, followed by the description of the object detection process in Section 3. Preliminary experiments and results are described and presented in Section 4. We conclude with the discussion in Section 5.

2 Model

Assume that a 3D object can be modelled by multiple planar models, obtained from multiple views of the object $M = \{T_j\}_{j=1:N_v}$, where N_v corresponds to the number of viewpoint images required to cover the entire object.

2.1 Encoding planar templates

A single view, i.e. object face is encoded by a set of keypoints $\{k_j^{(l)}\}_{j=1:N_k}$ and an intensity template $\mathbf{I}_j$ of size $n \times n$, where N_k corresponds to the number of extracted keypoints and $\mathbf{I}_j$ to a matrix of intensities. In our setup, the extracted keypoints are represented by position, scale and angle. Thus each keypoint can be encoded by a pair of points $[c_j^{(l)}, o_j^{(l)}]^T$, where $c_j^{(l)}$ represents the image coordinates of the keypoint and $o_j^{(l)}$ the keypoints angle and scale. Assume $p_j^{(l)}$ is the center of an object in image coordinates, which, for detection purposes, we encoded in the keypoints local coordinate system.

3 Detection

Due to unknown 3D orientations of objects, separate detections are ran for all object templates. The object of interest is detected in a cascade using the generalized Hough transform [1], MLESAC [8] and a verification step utilizing normalized cross correlation.

3.1 Generalized Hough transform

We utilize the generalized Hough transform [1] for object center detection, since it can be used to detect an arbitrary object if encoded in Hough space.

Initially keypoints are extracted from the image and each keypoint is matched to the most similar keypoint in the model. The matching is done in terms of the smallest distance, i.e. nearest neighbours. A similarity transformation matrix $\mathbf{T}$ between $[c, o]$ and matched $[\tilde{c}, \tilde{o}]$ is calculated, and a predicted object center is calculated as $\tilde{p}_j^{(l)} = \mathbf{T}p_j^{(l)}$.

The id of the predicting keypoint is recorded and the predicted object center is accumulated in an accumulator field of ρ square cells width. This is performed for all keypoints. To account for noise in keypoint location and errors in approximating a general perspective projection with similarity transform, each keypoint votes into a rectangular neighbourhood, with the size of the rectangular proportional to the detected keypoint scale. The so obtained array is post-processed by nonmaxima suppression and only the local maxima exceeding a threshold θ are considered for further processing.

Since we can identify keypoints that voted for the object center, we first apply a filtering step to remove keypoints that are outliers. This is achieved by the analysis of the scale of the voting keypoints. The assumption is that correctly detected keypoints should be predicting approximately equal scale of the object. To remove the outliers the following technique is applied. Let $\{S_i\}_{i=1:N_c}$ be the set of object scales that keypoints voted for, where N_c is the number of keypoints that voted for object center and S_i is computed as the Euclidean norm $||S_i|| = \sqrt{\mathbf{T}_i(1,1) + \mathbf{T}_i(2,2)}$. This set is pruned by removing ten percent of the keypoints with lowest and highest predictions of object scale. This yields a set of keypoints predicting a more refined scale of the object, with matches in the object model.

3.2 Iterative bounding box estimation

On the obtained set of keypoints, MLESAC is applied in order to further omit possible outlier interference, and fit a homography to the best keypoint subset. This yields a prediction of the objects location in the image to which we apply additional filtering.

The predicted object location, encoded as the objects bounding box and the fitted homography are verified in order to detect matrix singularity and strong bounding box skewing which imply false homography estimation. If the verification is passed, all the keypoints that lie outside of the estimated object bounding box are removed, and MLESAC is applied on the set of remaining keypoints, yielding a new best fitting homography. This process is repeated until the set of selected keypoints stops changing. The filtering process is illustrated in Figure 1.

3.3 Normalized cross correlation verification

The final verification step requires the stored intensity template $\mathbf{I}_j$, of the corresponding face of the object. The detected bounding box is transformed to a $n \times n$ intensity matrix $\tilde{\mathbf{I}}_j$ which is verified against the stored intensity template. In order to address partial object occlusions both intensity templates are divided into a two by two grid as shown in Figure 2. Normalized cross correlation is computed for each cell, as shown in Figure 3, and the mean of two highest responses is taken as the final matching score. The bounding box hypothesis passes the test if this score exceeds a threshold γ, otherwise the hypothesis is rejected.

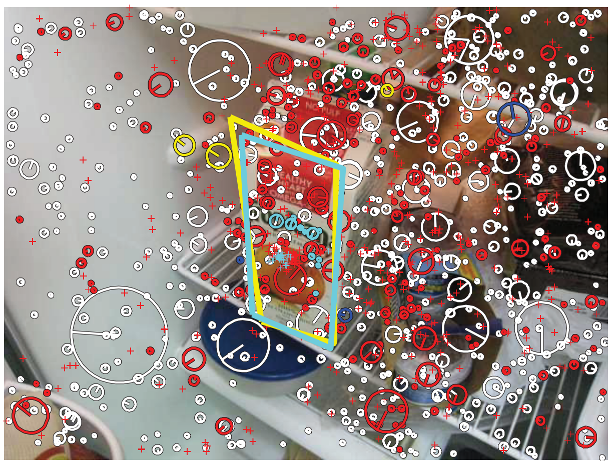

Figure 1: Example of iterative bounding box estimation. The blue colour indicates keypoints that voted for object center but were removed by scale filtering. The yellow colour bounding box is the initial estimation and the yellow keypoints are the keypoints that were removed, since they lie outside of the yellow bounding box. The cyan keypoints are the final keypoints used for the final cyan bounding box estimation.

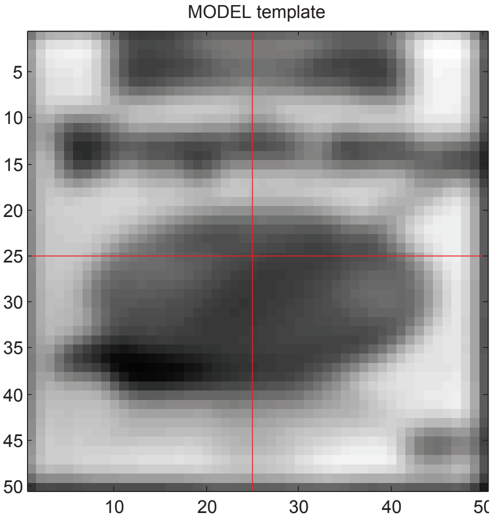

Figure 2: Intensity template division. Before computing normalized cross correlation intensity templates are divided into a two by two grid. The normalized cross correlation is then computed for each corresponding cell and the mean of two highest responses is taken as the final matching score.

4 Experiments and results

We evaluated our proposed model on the challenging real world dataset [4], containing everyday grocery items in scenes with strong illumination and viewpoint changes, occlusions, clutter, and examples containing multiple instances of the

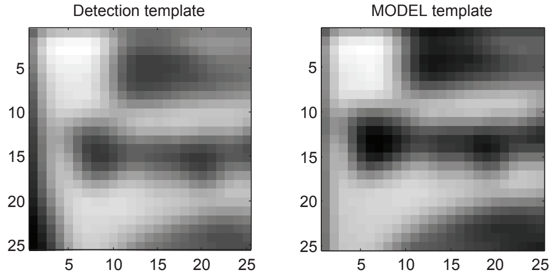

Figure 3: Normalized cross correlation. The upper left cell of the detected object intensity template is compared to the upper left cell of the model intensity template.

same object. We took seven out of ten dataset grocery items, displayed in Figure 4, for our experiment to estimate preliminary model performance. Items ”ricepilaf”, ”rice tuscan” and ”diet coke” where not used in our preliminary experiment since we used planar viewpoint images of objects to construct viewpoint consistent objects models and no planar images where available for these items. The detection of each of the seven items was evaluated on 50 images per item, containing one or more instances of the same item. The total number of processed images was 350.

The experiment was executed as follows. For each of the seven items an object model M_i is learned using a set of different viewpoint images T_j of the object. For each object a set of test images containing one or multiple instances of the object is processed. A detection is valid if the detection bounding box passes the filtering step and if the overlap with the provided ground truth bounding box is over 0.3 in terms of the Pascal VOC overlap measure [3], which is defined as the fraction of the intersection between two bounding boxes and their union.

The generalized Hough accumulator filed is set to $\rho = 50$ square cells width and thresholded with $\theta = 10$. The threshold value for normalized cross correlation is set to $\gamma = 0.5$. Keypoints are extracted using the SIFT [5] keypoint detection algorithm. In our experiment, we use the intensity template size of $n = 50$ pixels.

4.1 Results

The results of the experiment are presented in Table 1. For each item we compute precision, recall and the F-measure, i.e. the harmonic mean of precision and recall. Figure 5 shows an example of detecting multiple instances of the same objects from different object viewpoints. Figure 6 shows an example of partly occluded object detection.

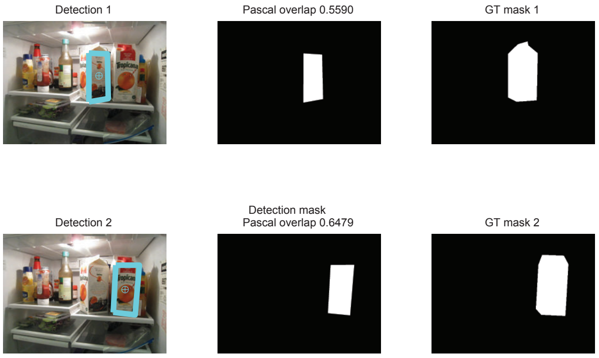

Figure 5: Object detection example. Our proposed method is able to detect an object from multiple views as well as detect multiple instances of the same object.

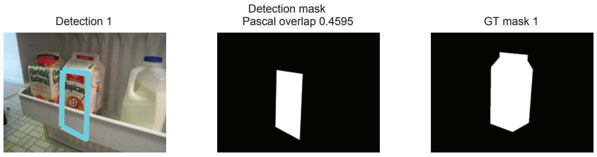

Figure 6: Occluded object detection. Detection is robust even though almost the whole bottom half of the object is occluded.

We can observe that for most items, precision, i.e. the fraction of retrieved instances that are relevant is quite high, especially in the case of cuboid items like ”Carton oj”, ”Carton soymilk” and ”Juicebox”. In the case of Cylindrical items like ”Can chowder” and ”Tomatosoup” a drop in precision performance is noticeable. A substantial precision performance drop is noticeable in the case of low textured or ovoid items as in the case of ”Can soymilk” and ”Potroastsoup”. Recall, the fraction of relevant instances that are retrieved, however is high only for ”Carton oj”. Only approximately half of items were detected for ”Can chowder”, ”Carton soymilk” and ”Juicebox”. The lowest recall values are reported for ”Can soymilk”, ”Tomatosoup” and ”Potroastsoup”.

During the experiment we noticed examples of false detections, i.e. detected locations that don’t contain the object in question as shown in Figure 7. This is due to the fact, that the current proposed method relies on a single keypoint to cast a vote. This presents a problem since arbitrary keypoints, not located on the object in question, can be matched to model keypoints and thus introduce noise into the detection process. Even the last verification step utilizing normalized cross correlation cannot prevent a false detection if the detected regions intensities are, to some extent, similar to the model intensities. In some cases we noticed mismatched bounding box estimations. Again, the reason for this could lie in the already mentioned fact that we rely only on one keypoint to cast a vote. If enough keypoints, that don’t line on the object in question, vote for the same center there still might be outliers, voting for a different object scale, thus interfering in the bounding box estimation process. Another occurring problem is that some objects don’t get detected. The reason behind this probably lies in the tuning of the keypoint detection algorithm’s parameters in order to extract a sufficient amount of keypoints even on

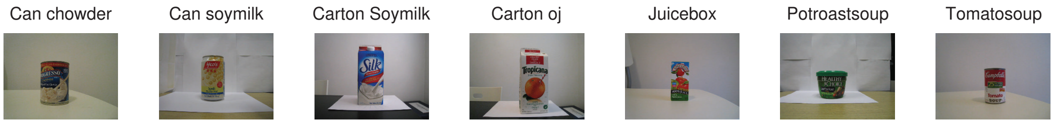

Figure 4: Dataset grocery items used in our experiment. Cuboid items are ”Carton oj”, ”Carton soymilk” and ”Juicebox”. Cylindrical items are ”Can chowder”, ”Tomatosoup” and ”Can soymilk”. ”Potroastsoup” is an ovoid item.

Object	Precision	Recall	F-score
Can chowder	0.5781	0.4933	0.5324
Can soymilk	0.3548	0.2716	0.3077
Carton oj	0.7397	0.7500	0.7448
Carton soymilk	0.7241	0.4516	0.5563
Juicebox	0.6143	0.5119	0.5584
Potroastsoup	0.2586	0.3750	0.3061
Tomatosoup	0.5079	0.3019	0.3787

Table 1: Evaluation results. For each object, the proposed model detection performance is expressed in precision, recall and F-score.

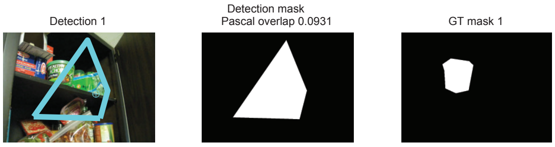

Figure 7: False detection. An object is falsely detected due to single keypoint voting. Since a large green area is found in the scene, which is the same colour as the object in question ”potroastsoup”, arbitrary keypoint detections appear which are matched to the object model.

smaller object occurrences, i.e. objects that appear farther in the background.

5 Conclusion

The proposed keypoint based detector demonstrates a solid performance on the challenging real world dataset. A 3D object is modelled by multiple planar models, obtained from multiple views of the object. Each view, i.e. face of the object is encoded by a set of keypoints and an intensity template. Object detection is performed for all object templates. Estimated object locations are verified in a cascade of verification steps to ensure a robust object detection. There are still examples of false detections, i.e. detected locations that don’t contain the object in question. As argumented, this is due to the fact, that the current proposed method relies on a single keypoint to cast a vote, which potentially introduces noise into the detection process, as keypoints can emerge on arbitrary locations. In our future work we plan to research local keypoint grouping as this could reduce noise, introduced by single keypoint voting. Furthermore a local group of keypoint should improve detection performance, as specific groups of keypoints are not likely to appear at arbitrary locations.

References

[1] D. H. Ballard. Readings in computer vision: Issues, problems, principles, and paradigms. chapter Generalizing the Hough Transform to Detect Arbitrary Shapes, pages 714–725. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007.

[3] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[4] E. Hsiao, A Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2653–2660, June 2010.

[5] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV ’99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[6] M. Merler, C. Galleguillos, and S. Belongie. Recognizing groceries in situ using in vitro training data. In *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, pages 1–8, June 2007.

[7] Paolo Piccinini, Andrea Prati, and Rita Cucchiara. Real-time object detection and localization with sift-based clustering. *Image Vision Comput.*, 30(8):573–587, August 2012.

[8] P. H. S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.*, 78(1):138–156, April 2000.

[9] T. Winlock, E. Christiansen, and S. Belongie. Toward real-time grocery detection for the visually impaired. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49–56, June 2010.

[10] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR ’11, pages 809–816, Washington, DC, USA, 2011. IEEE Computer Society.