Robust Method of Vote Aggregation and Proposition Verification for Invariant Local Features

Grzegorz Kurzejamski, Jacek Zawistowski and Grzegorz Sarwas
Lingaro Sp. z o.o., Puławska 99a, 02-595 Warsaw, Poland

Keywords: Computer Vision, Image Analysis, Multiple Object Detection, Object Localization, Pattern Matching.

Abstract: This paper presents method for analysis of the vote space created from the local features extraction process in a multi-detection system. The method is opposed to the classic clustering approach and gives a high level of control over the clusters composition for further verification steps. Proposed method comprises of the graphical vote space presentation, the proposition generation, the two-pass iterative vote aggregation and the cascade filters for verification of the propositions. Cascade filters contain all of the minor algorithms needed for effective object detection verification. The new approach does not have the drawbacks of the classic clustering approaches and gives a substantial control over process of detection. Method exhibits an exceptionally high detection rate with conjunction with a low false detection chance in comparison with alternative methods.

1 INTRODUCTION

Object detection based on local features is well known in the computer vision field. Many different researches brought about different features and methods of scene analysis in search of a particular object. Recently developed feature points are proven to be well suited for a specific object detection, rather than a generalised object's class identification. As many applications of local features has been evaluated, the ability to describe selective elements of rich graphics is the main purpose of invariant local features in many fields. Amongst the most popular local features are *e.g.* SIFT, SURF, BRISK, FREAK, MSER. These are commonly described as feature points or feature regions. They are easy to manipulate and to match. Use of such local features gives not only the ability to describe the object in many ways, but also to achieve invariance for basic object and image transformation, as skew, rotation, blur, noise. Invariant local features used in conjunction with invariant characteristic region detectors, such as Harris-Affine or SIFT detector, provide data for thorough scene-object analysis, leading to detection of an object in the scene. Giving an appropriate set of features one can determine the exact position of the object in the scene with a specified scale, rotation and even minor, linear deformations.

The straightforward approach to a detection task is to identify local features in the scene and the pattern and match them against themselves, using an appropriate metric. Common evaluations of such systems use brute KNN classification or its derivatives as FLANN KNN search or BBF search. These give good approximation of, theoretically ideal, results with substantially lower computational cost. Some of the applications make use of a LSH hashing but this approach does not provide practical distance data needed for further analysis. Matching local features provides set of correspondences, that can be filtered later. Under the assumption, that the scene contains one or no instance of the object, correspondence can be put into the object-related or noise-related class. To distinguish which class a particular correspondence belongs to, few methods have been developed. Frequently used method for such purpose is RANSAC, giving very good results, even for a high level of noise-related correspondences. After the classification of correspondences the model can be assumed and a homography can be calculated. Well designed parameters and filters can lead to high detection rate and low false detection chance. RANSAC and quantitative analysis of correspondence data become inefficient, when a contribution of noise-related correspondences in whole correspondence group grows. Because of that such approach is not sufficient for scenes, where the objects occupy small share of the image. Methods mentioned above will not work well with multiple objects present in the image as well.

Systems for multi-detection purposes incorporate divide and conquer approach. Each correspondence

Kurzejamski G., Zawistowski J. and Sarwas G..
Robust Method of Vote Aggregation and Proposition Verification for Invariant Local Features.
DOI: 10.5220/0005267002520259
In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications* (VISAPP-2015), pages 252-259
ISBN: 978-989-758-090-1

can be assigned to one of N+1 classes, where N is the number of objects in the scene. There is no known, straightforward method of assigning correspondences. Most applications use correspondences as votes in a multidimensional space. Vote space can be clustered with common clustering algorithms. Each cluster can be processed with a single-object detection algorithm. Clustering approaches can be divided into two groups. The first one consists of sparse clustering, where each cluster should contain all the needed vote data of a specific class. Such methods show low detection rate because of a far from ideal clustering process and a high level of parametrization. Second group is dense clustering, where clusters may contain only small portion of a particular correspondence class. Its analysis leads to creating or supporting hypothesis of the object's occurrence in the scene. The flagships in that matter are Hough-like methods. Such approaches show high detection rate but high false positive rate as well. There are some works that try to segment the scene with known context, as shown in the work of Iwanowski *et al.* (Iwanowski et al., 2014).

In our tests both clustering approaches lacked the ability to attain a very high detection rate with a very low false positive rate at the same time. For our test cases a processing power is not a limitation and the images are of a very high quality. Our test data contains from zero to up to 100 objects per image and presents different environmental conditions. Most of the stat-of-the-art publications do not test detection capabilities for such complex tasks. We found that current approaches cannot maintain good detection rate to false positive rate ratio on satisfactory level in many real life applications.

This paper presents the method of vote space analysis, a part of invention shown in (Kurzejamski et al., 2014). The method can be adjusted to a vast variety of object detection purposes, where the effectiveness and a low false positive rate is crucial. The method has been developed to work well with huge amount of feature data, extracted from high quality images. Most of the algorithms used in the new approach come with a logical justification. The new method uses the iterative vote aggregation, starting from proposition's positions. Propositions are generated from a graphical vote space analysis. Aggregated data undergoes analysis and filtration. Whole process has a two-pass model, that makes the method robust to some specific object positioning in the scene. Cascade of a specially selected set of filter algorithms has been utilized to reject most of the false positive detections.

2 RELATED WORK

Local features in the image can be tracked a long way in the literature. We present state-of-the-art feature points extracting and describing methods, that can be used in our method, and similar frameworks for multi-object detection purposes developed through the last years.

2.1 Feature Points

Our method should be used with conjunction with scale-invariant and rotation-invariant features for the best results. Usage of local features lacking any of this characteristics may come with a need for rejection of some parts of our method, but can be implemented nevertheless.

The best known feature points, up to this point, are SIFT points developed by Lowe (Lowe, 1999), which became a model for various local features benchmarks. The closest alternative to SIFT is SURF (Bay et al., 2008), that comes with a lower dimensionality and, in the result, a higher computing efficiency. There are also known attempts to incorporate additional enhancements into SIFT and SURF as PCA-SIFT (Ke and Sukthankar, 2004) or Affine-SIFT (Morel and Yu, 2009). SIFT and SURF and its derivatives are computationally demanding during matching process. In last years there has been big development in feature points based on binary test pairs, that can be matched and described in a very fast manner. The flagships of this approach are BRIEF (Calonder et al., 2010), ORB (Rublee et al., 2011), BRISK (Leutenegger et al., 2011) and FREAK (Alahi et al., 2012) features. Most of the cited algorithms can be used to create dense and highly discriminative voting space, which holds substantial object correspondence data needed to accomplish many of the real-world detection tasks.

2.2 Frameworks

There are few approaches to conduct multi-object multi-detection, meaning detecting multiple different objects on the scene, where any object can be visible in multiple places. Viola and Jones (Viola and Jones, 2001) developed cascade of boosted features, that can efficiently detect multiple instances of the same object in a one pass of the detection process. The method needs a time consuming, learning process with thousands of images. Method has been mostly tested on general objects, as people, cars, faces. Most straightforward method for multi-detection is using all of the sliding windows as used,

for example, in Sarwas' and Skoneczny's work (Sarwas and Skoneczny, 2015). Most of them are unfortunately computationally expensive. High effectiveness can be achieved with Histogram of Oriented Gradients (Dalal and Triggs, 2005) and Deformable Part Models (Felzenszwalb et al., 2010). The biggest drawbacks for our application is that Deformable Part Models needs learning stage and Histogram of Oriented Gradients is not rotation invariant. Blaschko and Lampert in (Blaschko and Lampert, 2008) uses SVM to enhance the sliding window process. Efficient subwindows search has been used in (Lampert et al., 2008). In addition, branch-and-bound approaches, as in (Yeh et al., 2009), are promising for multi-detection purposes with conjunction with Bag-of-words descriptors. Lowe (Lowe, 2004) proposed generalized Hough Transform for clustering vote space with SIFT correspondence data. Authors of (Azad et al., 2009) created a 4D voting space and used combination of Hough, RANSAC and Least Squares Homography Estimation in order to detect and accept potential object instances. Zickler in *et al.* (Zickler and Efros, 2007) used angle differences criterion in addition to RANSAC mechanisms and vote number threshold. Zickler *et al.* in (Zickler and Veloso, 2006) used custom probabilistic model in addition to Hough algorithm.

In our system's application we could use only one generic pattern image per object so we rejected most of the learning-based global descriptors.

3 ALGORITHM

The algorithm presented by authors is built upon two mechanisms: the vote spaces creation and a vote aggregation for each of the vote spaces created. The vote space is created for each pattern. Its adjacency data is projected onto the (X, Y) plane, creating vote images (one for each vote space). The vote images are analysed in search for object's position propositions. This mechanism is shown in the Algorithm 1. The aggregation process is performed for each vote space and for its each proposition, starting from the proposition with the highest adjacency value. Aggregation consists of two passes with slightly different vote gathering approaches. The first pass is needed to estimate the detected object's area in the scene, so the second aggregation pass would gather only votes considered to be from that particular object's instance. The structure of each pass is presented in Algorithm 2.

Algorithm 1: Vote Data, Vote Image and propositions creation.

```
Data: Original Patterns (OPT), Scene Image (SCN)
Result: Vote Data and propositions for object's centres for each pattern.
1  Feature points extraction on OPT and SCN;
2  foreach pattern in OPT do
3      Find correspondences (COR) between pattern and SCN feature points;
4      foreach correspondence in COR do
5          Reject if has low distance value;
6          Reject if has high hue difference value;
7          Calculate adjacency value;
8      end
9      Creation of vote space (VS) from COR;
10     Creation of vote image (VI) from VS;
11     Search for propositions (PR) in VI;
12     Sort PR list;
13 end
```

Algorithm 2: Vote aggregation and detection acceptance.

```
Data: VI, VS, PR
Result: Occurrences (OCR) in the SCN for a particular pattern
1  foreach Proposition in PR do
2      Gather all votes in local area from VS;
3      Unique filtering for gathered votes (V);
4      Cascade filtering for V;
5      if not rejected by cascade filtering then
6          Estimate object's area;
7          Gather all votes with a Flood Fill algorithm;
8          Unique filtering for new V;
9          Cascade filtering for new V;
10         if not rejected by cascade filtering then
11             Calculate object's area;
12             Create occurrence entry in OCR;
13             Erase all vote data in occurrence's area in VS and VI;
14         else
15             reject proposition
16         end
17     else
18         reject proposition
19     end
20 end
```

3.1 Vote Image Creation

First part of our method is the vote space and vote image creation (lines 9 and 10 of Algorithm 1). Vote

space consists of multiple dimensions: X, Y, Scale, Rotation and Distance. Each vote contains specific X and Y position of the center of the object. The Distance may be the result of using specific metric for particular feature points. For SIFT the standard procedure is to use L2 distance for its feature vector, which contains gradient data in the area around the characteristic point. One may use some additional information in distance calculation, as color difference. Someone can use ranking method as the LSH hashing instead of the L2 metric as well.

Vote image is the projection of the adjacency data available in vote space onto X and Y dimensions. Vote image has one intensity channel, created by normalizing adjacency sum cue. Another approach would be to use the distance value instead of adjacency as the main cue. We found L2 metric, as well as many other distance-based approaches, as insufficient.

Votes in the vote spaces are built upon filtered correspondence sets. The distance threshold used in line 5 of Algorithm 1 was calculated as:

$$thr = \frac{MIN(V) + MAX(V)}{2}, \tag{1}$$

where V is the votes group and the MIN and MAX operators return the value of a vote with minimal and maximal distance value from the group. The rejection function D is presented in equation 2.

$$D(v) = \begin{cases} accept, & dist(v) \leq thr \\ reject, & dist(v) > thr \end{cases}. \tag{2}$$

We transform distance value into normalized adjacency value in range from 0 to 1 (line 7 of Alg. 1). 1 indicates perfect match. 0 indicates near to rejection difference between feature points. We transformed distance values into adjacency (*adj*) values with a specific function:

$$adj(v) = 1 - \left(\frac{dist(v)}{thr}\right)^2. \tag{3}$$

Adjacency values are gathered in a single channel, gray vote image. Vote image can be optionally normalized for visualization purposes. Such normalized image has been shown in Figure 1. If the feature extraction and matching process are highly discriminative, the object instances in the scene should be recognizable by a human. Manual verification of vote image gives some level of valuable insight into votes intersperse in the image and a level of a false votes groupings recognizable by a human.

Last step of Algorithm 1 is the search for propositions in a vote image. Proposition is a point in the vote image and corresponding part of vote space, where the potential object's center is located. We used Good Features To Track by Tomasi and Shi (Shi and Tomasi, 1994) to detect multiple local maximas in the vote image and used them as the propositions. The number of propositions should be much higher than a number of objects in the image. It is trivial to set the Good Features To Track to find all the important points in the image, but it leads to generation of thousands of propositions. Number of propositions will significantly impact the algorithm's processing time, so it is not possible to ignore the need for a trade-off in detector's parameter adjustment. For each proposition's X and Y position the adjacency sum for corresponding votes in vote space is calculated and used for sorting purposes in line 12. The highest adjacency value proposition should be the first taken later into a vote aggregation process. As the adjacency sum is proportional to the channel value in the vote image, the cue for sorting stage is easy to compute. Sorting the propositions ensures, that the strongest vote grouping will be processed first. In case of the positive object recognition, the vote data corresponding to object's detection area will be erased from vote space and image.

3.2 Vote Aggregation

Second part of our approach contains a vote aggregation mechanism. Vote aggregation starts from a proposition's position, which should be the center of a local vote grouping in the vote image. Data of the vote groupings can significantly vary for different object instances in the scene. The best instances can be represented by hundreds of votes, when the weakest positive object response can be connected with only a few. Generic clustering may ignore such clusters and merge it with the bigger ones. Generic clustering algorithms has generic parameters, that are hard to adjust with object-oriented logic or even intuition. Some clustering approaches tend to cluster all the available vote data, even if the noise (false correspondences) fills most of the vote space.

We propose an iterative 2-pass vote aggregation process for selective clustering purposes. In each pass the unique filtering and the cascade filtering take place, which reject false positive detections. Two pass design prevents situations in which aggregation area contains multiple objects. Pass one of the aggregation collects all the votes in local area of proposition's position (line 2 of Alg. 2). The size of a local area may be a function of a corresponding pattern size. After gathering of all the votes in the local area, the unique filtering is performed and the resulting group of votes

(a) Vote image (b) Blurred and normalized vote image (c) Scene with tea detections

Figure 1: Sample of vote image generated while localizing red, herbal tea casing.

is tested with a cascade of filters (lines 3 and 4 of Alg. 2).

In the second pass of the process the aggregation is conducted with Flood Fill algorithm, starting from the proposition's position (line 7 of Alg. 2). The Flood Fill range is limited to a scaled down object's area. Such limitation can be constructed with a scale and rotation estimation from the first pass of the aggregation. The limitation ensures, that the aggregation process will not collect the votes from neighbouring object instances. Second pass of the algorithm contains unique filtering and cascade filtering as well, as the vote collection may be different in this pass.

For each group of votes, a unique filtering should be performed in each pass (lines 3 and 8 of Alg. 2). Unique filtering preserves only one vote with the highest adjacency corresponding to the same feature point in the pattern. We can do so, because we want the aggregated votes to be connected with only one object. If multiple votes are connected with one specific feature in the pattern we can assume that only the strongest vote isn't the noise.

Most of the feature point detectors incorporate mechanisms of rejecting the points located along the edges. Unfortunately, this mechanisms work only in a micro scale. In high resolution some graphical structures, that for human seem as a straight edge, has a very complicated, uneven shape for characteristic points detector. Characteristic points located along the edges have similar features, so may be matched with the same feature in the pattern. It leads to generation of many false propositions, which can sometimes be accepted by a cascade filters.

Some of the false positive detections in our experiments were initiated as a bunch of feature points placed along a simple, steep gradients and edges. For instance, when the scene presented product shelves, more than a half of false detections contained edge of the shelf near the center and its vote data present mostly along the shelf's edge.

3.3 Cascade Filtering

Cascade filtering (lines 4 and 9 of Alg. 2) is a process of validating vote group with a cascade of filters. Each filter can accept aggregated votes or reject them. Any rejection will result in dropping the aggregation process and removing the processed proposition from the propositions sorted queue. No vote data is removed from vote space or vote image in that situation. If all the filters in first pass accepts the vote group, process may estimate the size and rotation of the object represented by the majority of votes (line 11 of Alg. 2).

Cascade filters comprise of: (1) vote count thresholding, (2) adjacency sum thresholding, (3) scale variance thresholding, (4) rotation variance thresholding, (5) feature points binary test, (6) global normalised luminance cross correlation thresholding. First pass of the vote aggregation uses filters: (1), (2), (3) and (4). Second pass of the process uses filters: (3), (4), (5) and (6).

Vote count thresholding is a simple filter, thresholding number of votes in the aggregated group. Lowe in his work (Lowe, 2004) proposed generalised Hough transform for object detection. In this method he has assumed that only three votes are enough to identify the object. Unfortunately, such assumption

leads to many false positive detections. Three local features are not enough to describe complex, generic graphics. We tested vote count thresholding for values from 3 up to 20. We found 6 as the optimal value for filtering out too weak responses. If the vote grouping represents real object instance and has less than 6 votes, it means that the prior algorithm processes has too low effectiveness.

Adjacency sum thresholding rejects all the groups of votes with sum of adjacency values less than a threshold value. This filter in certain circumstances can be used instead of the vote count thresholding. Nevertheless the rejection data from these two filters may give an insight into vote certainty levels of the detection. Even huge vote groupings with more than 100 votes may have a very low adjacency sum value.

Scale variance thresholding rejects all the groups of votes with a scale value variance higher than the threshold value. One may rebuild this filter into mechanism separating noise signal from positive detection signal with a Gaussian model. For our purposes such method is computationally too expensive. Simple variance thresholding rejects many false detections and is easy to compute.

Rotation variance thresholding rejects all the groups of votes with rotation value's variance higher than the threshold value. Rotation variance thresholding works analogically to scale variance thresholding but using the rotation values. A rotation variance is not straightforward to compute. We set twelve buckets for rotation values and choose the three buckets with the highest count number. Its resultant was taken as an average rotation. All the values has been rotated so the average rotation was assigned to 180 degrees. Then the variation in regards to 180 degrees has been computed and used for thresholding.

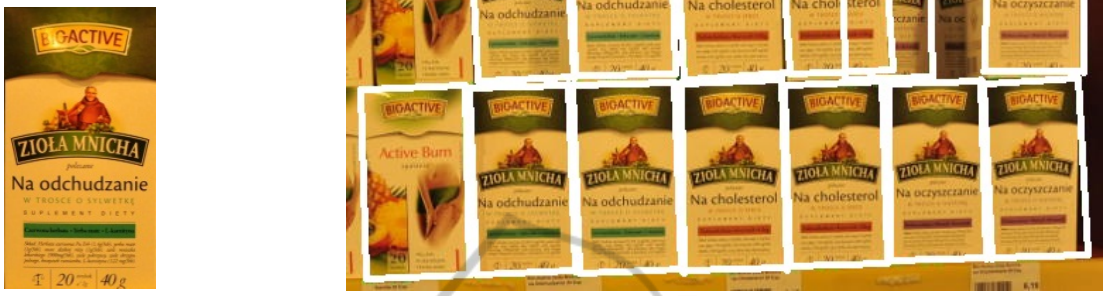
Feature binary test uses feature points correspondence data preserved in each vote. We created multiple luminance binary tests for random feature pairs in the scene, which are represented by votes in aggregated votes group. We created identical tests for corresponding feature points on the pattern side. Each set of binary tests provided a binary string that can be compared with a hamming distance. The normalised distance can be thresholded.

Normalised luminance cross correlation is used as a last filter. It needs the exact object's graphics patch extracted from the scene. It's computationally expensive, but can filter out many false positive detections, that cannot be filtered by previous filters. The images are resized to the size of 50x50 pixels before the calculation of the cross correlation. The filtering is conducted only in the second pass of the aggregation process, where the theoretical object's frame can be calculated from the data from the first pass.

4 EXPERIMENTS

Our testing platform, incorporating method described in this paper, has been developed to search product logos and casings on scenes presenting market shelves and displays. The database used for the test for this paper consists of 120 shelf photos taken in 12MPx resolution and scaled down to 3MPx for testing purposes. The pattern group consist of 60 generic patterns of logos and product wrappings. Each shelf photo was tested with each one of the patterns, giving 7200 detection processes. The photos contained usually three classes of products so most of the patterns could generate only false positive detections. Average number of products presented in the scenes was emph23.6. Patterns were scaled to have the bigger size between 512 and 256 pixels. In the application of product search on the market shelves we describe high quality images as photos bigger than 2MPx, with a minimum of ten thousands pixels for the smallest searched object and all of the logos text readable for a human. Our aggregation approach bases its effectiveness upon chosen local features. We used SIFT implementation for main experiments. Main advantage of our method lays in filtering out false detections and processing all possible occurrences. SIFT is a state-of-the-art features detector and descriptor. Our test showed that 100% of the actual object instances were processed through our cascade filtering with a proper proposition's location. That's thanks to dense proposition detections and a straightforward vote image creation.

Detection effectiveness lays in proper vote group filtering. Amount of positive detections rejected during cascade filtering results from all the computer vision algorithms incorporated into detection system and can be hardly used to measure aggregation effectiveness without proper comparisons with similar methods in the same application field. False detection rate yields more analytical data. We found no false positive detections during our tests, that were fault of not sufficient description capability of feature descriptor. All of false detections were the result of too loose parameters, that were needed for very high positive detection rate. Nevertheless we came across 203 false detections in 129 of 7200 detection processes, resulting in more than 1% (Table 1) false detection chance per detection process. This result seems low, but at the same time means emph66.2% chance, that the false detection will take place when looking for any product instance from our patterns database.



(a) Pattern

(b) Scene

Figure 2: Sample of detection results.

Our method has been compared to the method using the HOG descriptor. For the training stage we generated set of 60 derivative images for each pattern through small affine transformations. We used all other patterns as a negative images. We used implementation of HOG method, called Classifier Tool For OpenCV and FANN (HOG, 2014). Our method achieved only slightly better detection rate, but significantly lower chance for false detections. The average number of false detections were almost two times higher for the HOG approach (Table 2).

Table 1: Detection rate and false detection chance for our tests.

Method	Detection Rate	False Detection Chance
Ours	81.3%	1.79%
HOG	73.6%	21.42%

Table 2: Average number of false detections for process, where the false detection occurred.

Method	Average Number of False Detections
Ours	1.57
HOG	3.18

During experiments with product casings we encountered number of problems with association of detections to a specific result group. Some products are very similar, with only slight local graphical differences. This is particularly true for the same brand with different aromas or casing sizes. Figure 2 presents one of such cases, where tea casing has identical logo for its few variations with one being visually very different from the others. We decided to interpret only the visually off tea as a false detection. In retail field the rest of the detections should be processed further to discriminate different variations of the products. One can use partial patterns with a bag-of-words approach on top of our aggregation method to do so.

5 CONCLUSIONS

In this paper the method of vote aggregation designed for use in multi-object multi-detection systems has been introduced. Aggregation process yields promising results in tests, leading to analysis of each potential object in the image. The unique filtering leaves out many false object occurrence propositions and the cascade filtering rejects most of the false positive detections, that is crucial for presented application. System built upon the aggregation method can achieve more than 80% detection rate with the false detection chance below 2%. It is still far from industrial standards, but there are many places for improvement as well.

Presented method is designed to analyze very high quality images. Images processed in tests were taken by a hand, resulting in high amount of blurred and skewed visual data. The method of image acquisition should be analyzed further. In future work we will incorporate estimates of the best parameters for presented method as well as solve simple parametrization dependencies. We are going to test the system with a two-phase approach, where second phase of the detection would use the patterns extracted directly from the scene. The pattern size has too much impact on the detection rate, as the feature points approach works the best, when the objects in the scene and in pattern images have the same size. We are going to evaluate resizing options for better detection results.

ACKNOWLEDGEMENTS

This work was co-financed by the European Union within the European Regional Development Fund.

REFERENCES

(2014). Classifier tool for opencv and fann v. 4.11.8. http://classifieropencv.codeplex.com/. Accessed: 2014-12-18.

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). FREAK: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517.

Azad, P., Asfour, T., and Dillmann, R. (2009). Combining Harris interest points and the SIFT descriptor for fast scale-invariant object recognition. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4275–4280.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346 – 359. Similarity Matching in Computer Vision and Multimedia.

Blaschko, M. and Lampert, C. (2008). Learning to localize objects with structured output regression. In Forsyth, D., Torr, P., and Zisserman, A., editors, *Computer Vision ECCV 2008*, volume 5302 of *Lecture Notes in Computer Science*, pages 2–15. Springer Berlin Heidelberg.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary robust independent elementary features. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Computer Vision ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 778–792. Springer Berlin Heidelberg.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1.

Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645.

Iwanowski, M., Zieliński, B., Sarwas, G., and Stygar, S. (2014). Identification of products on shop-racks by morphological preprocessing and feature-based detection. In Chmielewski, L., Kozera, R., Shin, B.-S., and Wojciechowski, K., editors, *Computer Vision and Graphics*, volume 8671 of *Lecture Notes in Computer Science*, pages 286–293. Springer International Publishing.

Ke, Y. and Sukthankar, R. (2004). PCA-SIFT: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506–II–513 Vol.2.

Kurzejamski, G., Zawistowski, J., and Sarwas, G. (2014). Apparatus and method for multi-object detection in a digital image. EU Patent 14461566.3.

Lampert, C. H., Blaschko, M., and Hofmann, T. (2008). Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

Leutenegger, S., Chli, M., and Siegwart, R. (2011). BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. volume 60, pages 91–110. Kluwer Academic Publishers.

Morel, J.-M. and Yu, G. (2009). ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571.

Sarwas, G. and Skoneczny, S. (2015). Object localization and detection using variance filter. In Choraś, R. S., editor, *Image Processing & Communications Challenges 6*, volume 313 of *Advances in Intelligent Systems and Computing*, pages 195–202. Springer International Publishing.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 593–600.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages 511–518.

Yeh, T., Lee, J., and Darrell, T. (2009). Fast concurrent object localization and recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 280–287.

Zickler, S. and Efros, A. (2007). Detection of multiple deformable objects using PCA-SIFT. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1127–1132. AAAI Press.

Zickler, S. and Veloso, M. M. (2006). Detection and localization of multiple objects. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 20–25.