Enhancing LINE-MOD Object Recognition with Winner Take All and Fast Approximate Nearest Neighbor Search

Abi Raja
Stanford University
abii@stanford.edu

Ivan Zhang
Stanford University
zhifanz@stanford.edu

Abstract

We explore several extension to the LINE-MOD detection and object recognition algorithm [1] to further improve upon its speed, scalability and invariance properties. More specifically, we use the Winner Take All (WTA) algorithm [2] combined with Fast Library for Approximate Nearest Neighbors (FLANN) to speed up the template matching process by transforming the template feature vectors into lower dimensional hashes and by using an approximate nearest neighbor approach to limit our template search space. In addition, we explore the use of depth image to make LINE-Mod detection scale invariant. We analyse the speed improvement as well as performance effects WTA has on the original LINE-Mod algorithm.

1. Introduction

Real-time object detection attempts to learn new objects in real-time as well as detect the existing objects in its model.

This is an interesting area of study because of the various applications in robotics. A good real-time object recognition algorithm would enable robots to perform complex tasks such as identifying mugs in the close vicinity in heavily occluded scenarios and fetching it for the human user. In particular, our goal with this project is to make an existing template matching algorithm, LINE-MOD, faster and more scalable.

Many approaches have been tried for real-time object detection. A class of approaches that have been popular in recent times are template matching-based approaches. Template matching is preferred to statistical techniques because new templates can be learned without modifying the existing model extensively. In our research, we base our work off the LINE-MOD algorithm [1] and attempt to make its detection phase much faster by reducing the dimensionality of the templates and by doing fewer comparisons with templates. The LINE-MOD algorithm has good precision recall performance and computes feature vectors very fast. However, it scales linearly with the number of objects and number of templates per object.

Winner take all, a hashing algorithm that has proven quite useful in generating better results for a wide variety of similarity searches in higher dimensions including matching local feature descriptors [2]. We combined WTA hashing with FLANN to improve the computational complexity of the original LINE-MOD algorithm.

The interesting challenges in this problem stem from trying to balance improved speed with the heuristic nature of the speed-up. Hence, the goal is to preserve recall that is comparable to brute force matching. In the Experiment section, we describe the effectiveness of the various approaches implemented by us and what we learnt from these different approaches.

2. Background/Related Work

As mentioned before, template matching has been used for various applications in robotics and other fields, such as facial recognition and medical image processing. The field can roughly be subdivided into two approaches: feature-based and template-based matching. In a typical feature-based matching technique, strong features such as corners or edges are present to aid the search for potential matching locations for the template in a test image [6]. However, when handling low-textured objects these methods tend to encounter difficulties. A typical template-based matching algorithm considers the template images in their entirety but tends to suffer performance issues since the matching process may require searching through a large amount of pixels to find potential matches for the template. We have found several works that attempted to reduce this computational complexity.

Some work focuses on reducing the complexity of computing similarity measures. [7] suggests relying on dot products to measure the similarity between template gradients and those of the test image. Though this similarity measure can be computed efficiently, it decreases rapidly

if center at which potential matches are evaluated deviates from the center of the true match. Hence potential match locations on test images must be evaluated densely to handle appearance variation, making the algorithm computationally costly.

Others explore the potential of limiting the amount of pixels one needs to search through in a given test image. [8] suggests a branch-and-bound scheme to drastically reduce this amount. As a high-level summary, it hierarchically splits the set of all possible sliding windows into disjoint subsets, while keeping bounds on the maximum similarity of a sliding window within each disjoint subsets. In this fashion, one can target the search at areas with the highest potential similarity scores and discard the rest of search space when possible.

2.1. LINE-MOD

The algorithm we are basing the paper on is LINE-MOD, which proposes a similary measure that, for each gradient orientation on the object, searches in a neighborhood of the associated gradient location for the most similar orientation in the test image. This can be summarized in the following mathematical formula:

$$F(I,T,c) = \sum_{p \in P} \left(\max_{c' \in R(c+p)} |\cos(grad(O,p) - grad(I,c')| \right),$$

where I is the given test image, $T = (O, P)$ is a template consisting of a training image O as well as a list P of locations of discriminative gradient features. Also, we have $R(c+p) = [c+p-\frac{s}{2}, c+p+\frac{s}{2}] \times [c+p-\frac{s}{2}, c+p+\frac{s}{2}]$ defining a square neighborhood centered around location $c+p$ on the test image. This expression then defines the similarity between test image I and template T at a given location c. This improves robustness of the matching algorithm regarding to small shifts and deformations, and it works particularly well when combined with a sliding window template matching algorithm with a particular stride (a number of pixels to skip in each iteration) in both vertical and horizontal directions.

LINE-MOD also uses a parcular gradient feature for matching. At each location of a given image, LINE-MOD computes the gradient orientation of each color channel and picks the orientation with the greatest gradient magnitude. Then, the algorithm quantizes the gradient orientations into eight equal spacings in order to represent each gradient orientation in 8 bits.

LINE-MOD speeds up this similarity calculation by spreading gradient orientations and precomputing response maps. More specifically, each gradient orientation at a particular location in effect "spreads" to a neighborhood of $s \times s$. This allows the gradient orientation at a particular location to be represented by an 8 bit vector with possible multiple bits turned on. This enables LINE-MOD to precompute a cosine similarity matrix of dimension 256×256 so that later queries can be answered efficiently.

As for general implementation, a LINE-MOD model first learns a list of templates for each object. Then, when a test image is encountered, the algorithm searches through the test image with a sliding window skipping 7 pixels each time horizontally or vertically. A sliding window contains a matching object if the similarity score between the gradient feature matrix of this window and that of a template is above a certain threshold. The preliminary bounding box is then determined by using the center of sliding window and the bounding box of the template. At the end of this search, non-max suppression is performed on all potential bounding boxes with an overlap threshold of 0.5. The remaining bounding boxes are the locations of the predicted objects.

Although LINE-MOD is able to compute similarity functions and gradient orientations efficiently, it does not address several problems. Firstly, the running time scales linearly with respect to the number of objects and templates learned. Secondly, the algorithm is not invariant under scaling. Thirdly, it does not address the problem of having to search through a large amount of pixels in a given test image. Our following approach addresses the first and second problem.

3. Approach

3.1. WTA Hash

The Winner Take All (WTA) hash is a sparse embedding method that transforms the input feature space into a much lower dimensional space while preserving the dimensionality orderings.

More precisely, the WTA hashing technique works as follows: for each hash value we first permute the input high dimensional vector with a permutation Θ, and then take the first K components of this permuted vector. The hash value is then the index of the biggest component of the first K components. We can then combine different hash values generated from different permutations Θ and combine them into a single hash vector. This hash vector can be of a much lower dimension than that of the original input vector.

The rationality of using WTA hash is as a potential solution to reducing the computational complexity of matching against many templates of the same object. Given the gradient feature matrix of a test image around a particular location, it is expensive to compute its cosine similarity with respect to every single template of the object. Instead, using WTA hash, we can reduce the dimensionality of feature vectors and limit our search scope within the space of possible matching templates. We discuss how this can be achieved below.

3.2. FLANN

To take advantage of WTA, we use the Fast Library for Approximate Nearest Neighbors (FLANN). After all templates are loaded for a particular object of interest, we transform every template's gradient feature vector using WTA hash, and build a FLANN index with this set of low dimensional vectors. Later, when a gradient feature vector of a test image comes around, we perform the same WTA hash to the vector, and perform K-nearest-neighbor search using the FLANN index. According to [Yagnik], using WTA hash significantly improves the performance of approximate nearest neighbor search, hence we have reason to believe that the K nearest neighbors returned in this way are close to the original test image's gradient feature vector in the original high dimensional vector space.

Before experimenting with WTA hashing, we also experimented with using FLANN without hashing. In other words, we construct a FLANN index with the original gradient feature vectors and later query approximate K nearest neighbors using the original gradient feature vector of a test image at a particular location. The results are shown in the Experiments section.

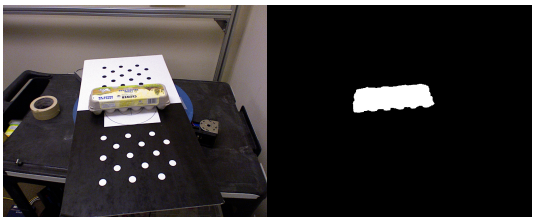
3.3. Utilizing Depth Information

The other problem with the original LINE-MOD method is that although it is invariant to small distortions of images, it is not inherently scale invariant. For instance, if the object in the training image is fairly far away from the camera, while in a testing image it is much closer to the camera, then with even if the object in the testing image has the same orientation as that in the training image, its gradient features around a certain location will be far more spread out than those in the training image. This causes difficulties for the template matching problem.

We can partially solve this problem by incorporating a depth image, and scale the computed gradient feature matrix by the average depth of the template area. We then store the scaled gradient feature matrices in each template, and when a gradient feature matrix is computed around a location in a specific test file, we also scale that feature matrix by the average depth of image within the bounding box of the template before we perform similarity matching. The limitation of this method is that it still does not solve the problem of different resolutions of images. For instance, two images may contain the same mug held at the same distance away from the camera but if the two pictures are of two resolutions then the mug will be smaller in the lower resolution (smaller) image.

4. Experiment

We tested the performance of using purely FLANN and WTA plus FLANN against the original brute force way of matching.



(a) Sample image file (b) Sample mask file

Figure 1: Example data files

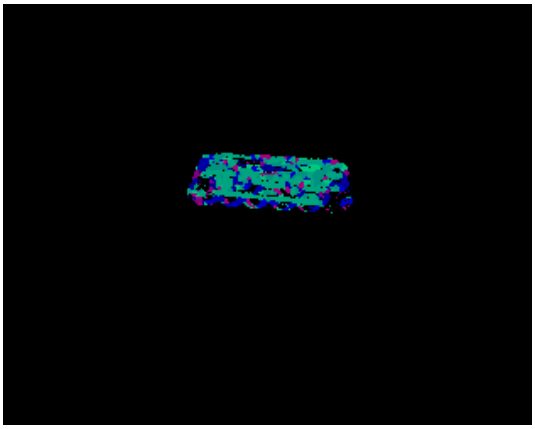


Figure 2: Example gradient feature teamplate

The image suite is provided kindly by Dr. Bradski from Willow Garage, and it consists of images of 20 different objects with around 70 images and corresponding mask files for each object. For a particular object, the 70 images all have different rotations and we separated these images into training sets and testing sets. See an example image file and an example mask file (used to determine where the object is during training phase) as well as a visualization of a gradient feature template below.

The first experiment we implemented was one to measure the performance of the WTA, i.e. we were trying to determine whether low dimensional vectors obtained after WTA hashing have their neighbors preserved compared to their original high dimensional corresponding vector before WTA hashing. To do this, for the particular object "Tea Box", we separate the 71 images into two groups: 70 training images and 1 test image. We train a LINE-MOD model using the 70 training images as wel as their corresponding mask files, and we test the single test image using the trained model. Since we know the structure of the single test file, we can determine when the sliding window reaches where the tea box is in the test file during testing phase based on the specific location of the tea box within the test file. At that point, we find out about the indices of

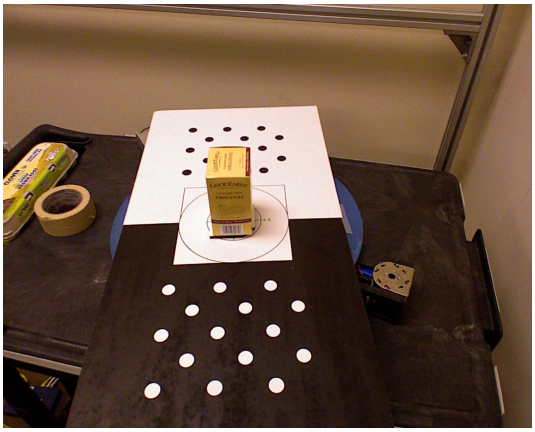
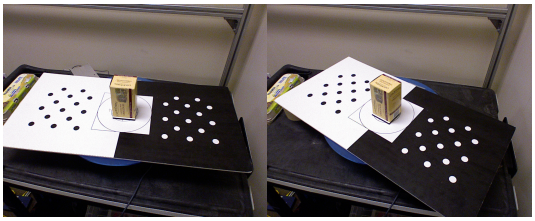


Figure 3: Test file



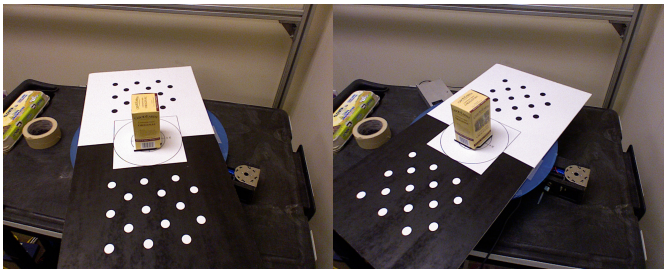


Figure 4: Nearest neighbors returned by FLANN using WTA

the templates that are returned as the K nearest neighbors of the current sliding window, and then based on those indices, we can calculate what are the corresponding original training files. The results are shown below:

The second experiment we implemented was to test the speed improvement during training and testing phase when using brute force method, purely FLANN, and WTA plus FLANN. We selected 2 objects and divided each object's image sets in half between training and testing images (35 images in each group). Then we ran the LINE-MOD algorithm with the 3 different methods while varying the number of approximate nearest neighbors we query for the last two methods.

Lastly we implemented an experiment to test how performance suffers when we purely use the FLANN library or if we use WTA combined with FLANN. Notice that the brute force method will have strictly more matches identified since it conducts an exhaustive search through all learned templates at any location on the test image. Given that the baseline LINE-MOD's precision is very high (95%), it is not likely for our proposed approaches to perform as well as the brute-force method in the strict sense of precision / recall curve. Also, we definitely see a trade-off between the speed improvement and matches missed due to WTA, and this shows that WTA does not fully captures the relative distance of the original high dimensional vectors in this particular object recognition problem. Here we follow the convention that a bounding box is correct if the area of the intersection with the "ground truth" rectangle over the area of union with the "ground truth" rectangle is over 0.6. The results of the two experiments above are summarized in the table below:

As we can see from the table, using FLANN by itself performs poorly even with relatively big number of nearest neighbors returned. This is possibly due to the fact that the library uses Euclidean distance as the distance measure when constructing hierarchical trees, while LINE-MOD uses cosine similarity during template matching. With WTA, however, we obtain a considerable speed up of the algorithm, although recall is affected. This is likely due to the features vectors losing discriminitivity during the WTA hashing procedure. However, the speed up is notable, so WTA hashing can potentially be used in combination with other clustering techniques in improving upon the speed of template matching.

5. Conclusion

We explored the possibility of enhancing LINE-MOD, a state-of-the-art template matching algorithm, using WTA hashing and FLANN library. Although using WTA hashing does reduce the discriminitivity of the features, we did achieve a significant speed boost using our approach. Future work includes implementing approximate nearest neighbor search using cosine similarity instead of the Euclidean distance built into the FLANN library, and also using other clustering algorithms to reduce the search space for the templates.

This project gave us the opportunity to read and implement algorithms published in very recent journals and trained our ability to work with and modify moderate-sized libraries.

6. References

[1] S. Hinterstoisser, and C. Cagniart. Gradient Response Maps for Real-Time Detection of Texture-Less Objects. *IEEE International Conference on Computer Vision (ICCV)*, 2011.

	Time (s)	Recall (Tea)	Precision (Tea)	Recall (Egg)	Precision (Egg)
LINEAR-SEARCH	286.977	1.00	0.85	0.94	1.00
WTA (knn = num / 2)	213.456	0.64	1.00	0.62	1.00
WTA (knn = num / 4)	173.893	0.48	1.00	0.44	1.00
FLANN (knn = num / 2)	240.912	0.21	0.92	0.33	0.90
FLANN (knn = num / 4)	203.122	0.11	0.90	0.18	0.90

Figure 5: Performance and speed analysis using 3 methods discussed

[2] J. Yagnik, D. Strelow, D. Strelow, and R. Lin. The Power of Comparative Reasoning. *International Conference on Computer Vision (ICCV)*, 2011.

[3] S. Lao, Y. Sumi, M. Kawade, and F. Tomita. 3D Template Matching for Pose Invariant Face Recognition Using 3D Facial Model Built with Isoluminance Line Based Stereo Vision, *Proceedings of 15th International Conference on Pattern Recognition*, 2000.

[4] L. Cole, D. Austin, and L. Cole. Visual Object Recognition using Template Matching. 2004.

[5] E. Adelson, C. Anderson, J. Bergen, P. Burt, and J. OgdenPyramid Methods in Image Processing. 1984.

[6] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 60(2), 91110, 2004.

[7] C. Steger. Occlusion, Clutter, and Illumination Invariant Object Recognition. 2002

[8] C. Lampert, M. Blaschko, and T. Hofmann. Beyond Sliding Windows: Object Localization by *Efcient Subwindow Search*. 2007.