Computer Vision Project (Part 1)

COMP9517 - Semester 1, 2015

Louis Tiao (3390558)
Edward Lee (3376371)

April 14, 2015

1 Introduction

In Part 1 of the project, we are tasked with implementing software to track a number of (planar) objects in sequential video frames. In other words, given a number of reference images containing planar objects, we are required to estimate the motion models that describe how the objects are changing in the video sequence over time, in terms of translation, rotation, scale and other homographic transformations. We present a high-level overview and discussion of our approach in section 2. Results are displays in section 3.

2 Approach

We use the conventional feature-based methods to address this task. Namely, we find the correspondences between distinctive features in the video frames and the reference images, such as corners, edges, blobs, etc. and thereby fit a homographic transformation to the corresponding points in the video frame and reference image. This process is broken down into the following subtasks, and we describe our approach for each one.

2.1 Detection and Description

To detect distinctive features in images and compute their description vectors (descriptors), we first experimented with the class of algorithms based on scale-space extrema detection: *Scale-Invariant Feature Transform* [Lowe, 2004] (SIFT) and *Speeded-Up Robust Features* [Bay et al., 2008] (SURF), the latter being an improvement and optimization of the former.

These algorithms encompass the tasks of detecting features and computing descriptors in a way that is both scale and rotation invariant. In practice, we found SIFT offered very good performance in terms of accuracy, but was impractically slow for real-time application. Although SURF offered better performance in terms of speed, it did so at the cost of some accuracy and was ultimately still too slow for real-time application.

Next, we experimented with (a combination of) other feature detectors and descriptor extractors, namely: *Features from Accelerated Segment Test* [Rosten and Drummond, 2006] (FAST), which is solely a feature detector, *Binary Robust Independent Elementary Features* [Calonder et al., 2010] (BRIEF), which is solely a descriptor extractor, and finally, *Oriented FAST and Rotated BRIEF* [Rublee et al., 2011] (ORB), which (as the name suggests) is an amalgamation of FAST and BRIEF with modifications to support rotation invariance.

As expected, the combination of FAST and BRIEF performed much faster than SIFT/SURF, and was of comparable accuracy to SIFT/SURF when there was little to no rotation, but of practically no accuracy in the presence of rotation.

To our dismay, we found that ORB+ORB (the combination of ORB as both a feature detector and descriptor extractor), while still very fast, was not more accurate than FAST+BRIEF in the presence of rotation, let alone without it.

In the end, we concluded that SURF+SURF was most suitable for real-time applications with rotation, and otherwise FAST+BRIEF for those without it. For offline applications, SIFT+SIFT remains the most suitable. We designed our software to allow for different combinations of feature detector and descriptor extractor to be specified at runtime, to account for the different possible kinds of applications.

Refer to section 3 for a visualisation of results.

2.2 Matching

Once we detect the features (keypoint) and compute their descriptor vectors in both the video frame and the reference image(s), we match them by identifying the nearest neighbor of the keypoint in a video frame in the set of keypoints of the reference images. The distance metric, and by extension, the definition of nearest neighbor is dependent upon the descriptor extractor, since the descriptor computed under SIFT/SURF is a 128-element real-valued vector, while it is a binary string under ORB/BRIEF.

For descriptor extractors that use real-valued vectors, we experimented with the exhaustive search matcher, implemented with `BFMatcher` in OpenCV, with the L_2-norm (Euclidean distance) as the distance metric. We also tried an approximate nearest neighbor matcher, specifically using k-d trees and implemented with `FlannBasedMatcher` in OpenCV and its interface to FLANN (Fast Library for Approximate Nearest Neighbors). Consistent with Lowe's observation in [Lowe, 2004], this did not provide a significant speed-up over exhaustive search.

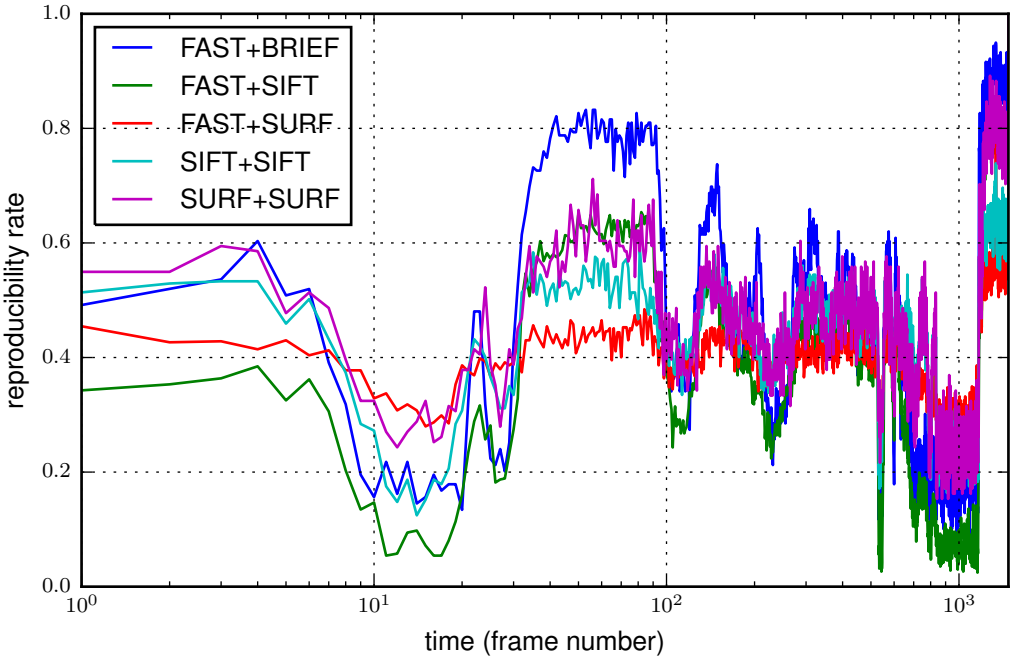
For descriptor extractors that use binary strings, we likewise use an exhaustive search, but with Hamming distance as the distance metric and later also tried an approximate nearest neighbor matcher (using Locality-sensitive hashing (LSH) instead of k-d trees). Again, this did not provide a significant speed-up over exhaustive search.

We perform the simple outlier rejection procedure suggested by Lowe after the above matches have been found by comparing the distance of the closest neighbor to that of the second-closest neighbor and discarding the match if the closest neighbor not much closer than the closest incorrect match. The rationale is that correct matches need to have the the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching.

2.3 Transformation

Once set of correspondences have been obtained, we use the iterative Random sample consensus (RANSAC) method to fit a homographic transformation to the pairs of points in the correspondences, which is almost guaranteed to contains outliers. This is implemented using the `findHomography` function in OpenCV with `CV_RANSAC` specified as the method. With the transformation matrix H, we visualize the located object in the video frame by applying H to the four corner points of the reference image using `perspectiveTransform` and drawing a closed polygon around those points on the video frame. This serves to highlight the object and its transformation with respect to the reference image.

3 Results



References

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. ISSN 10773142. doi: 10.1016/j.cviu.2007.09.014.

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6314 LNCS, pages 778–792, 2010. ISBN 364215560X. doi: 10.1007/978-3-642-15561-1_56.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 09205691. doi: 10.1023/B:VISI.0000029664.99615.94.

Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pages 430–443, 2006. ISBN 3540338322. doi: 10.1007/11744023_34.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571, 2011. ISBN 9781457711015. doi: 10.1109/ICCV.2011.6126544.