The Effects of Different Descriptors and Detectors on Consensus-Based Matching and Tracking

Qasim Ahmed, Andrés Solis Montero, Dr. Jochen Lang
School of Electrical Engineering and Computer Science, University of Ottawa

Introduction

Many computers today need to interact with their environments to accomplish their tasks. But this interaction has its limitations. There is a huge body of research around the world, which is currently being done to slowly chip away at these limitations; one of those limitations being sight. From robotics, to quality insurance, security and more the applications for computer vision and more specifically, long-term tracking and detecting of objects are truly endless.

Analogous to how sight in humans involves a lot more than just the eyes, sight to computers involves a lot more than just the camera. In fact cameras are doing an ever better task of capturing the world. The drawbacks of this are more data, and more work on behalf of the computer. The increase in computation has allowed computers to process more data than ever. But with more power comes more data, since we now have an ever greater need to process data. Thus algorithms need to be made not only to accomplish the task of long-term tracking and detecting, but to optimize it. This way more can be done with our always finite computational resources. Many algorithms have already been proposed, but progress is still needed because they are either unreliable or too computationally demanding.

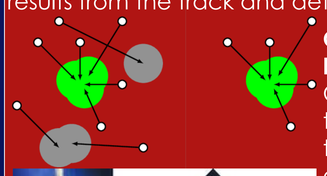
The first part of the research investigates and compares two different novel approaches to long term visual tracking called Track-Learn-Detect (TLD)[1] and Consensus-based matching and Tracking of key points (CMT)[2]. Comparison is done by using implementations[3] for both algorithms in C++. The second part of the research focuses on the effects of different feature detectors in the C++ implementation of CMT.

A benchmark analysis of TLD, CMT and different versions of CMT each with a different feature detector, are then done to quantify the experimental results.

Background

Tracking-Learning-Detecting

TLD is a novel approach to long term tracking and detection of objects. TLD first needs to be initialized. The algorithms then detects and saves Key points of that object. After initialisation the algorithm simultaneously Tracks and detects each frame. The algorithm also "learns" online by changing the information on the objects appearance depending on the results from the track and detect step.

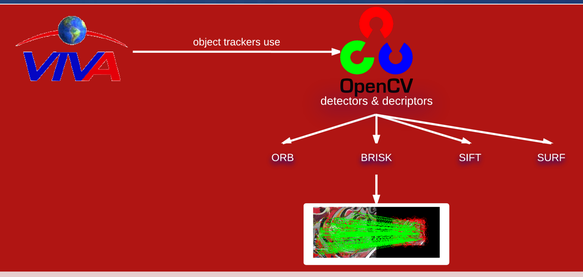


Consensus-based matching and tracking of key points

CMT is another novel approach to long-term tracking. Like TLD, the CMT algorithm needs to be initialised. After initialisation key points are made to track the object. Each key point votes for where the middle of an object may be; this is done with each passing frame. CMT does not update the information of the objects appearance; in contrast to TLD.

Methodology

Two implementation in C++ are used in this research; one for TLD and one for CMT. OpenCv is used for feature detecting and description which are two processes used in the CMT algorithm.
The four different feature detectors and descriptors used for this research are: Brisk[3], Orb[4], Sift[5], and Surf[6].
From the initial implementation of CMT three new implementations are made. These implementations are identical to the original Viva Labs implementation, but instead of using the Brisk detector and descriptors they use different ones form the OpenCV library.
The next step is to test the performance of the different CMT implementations and the TLD implementation.



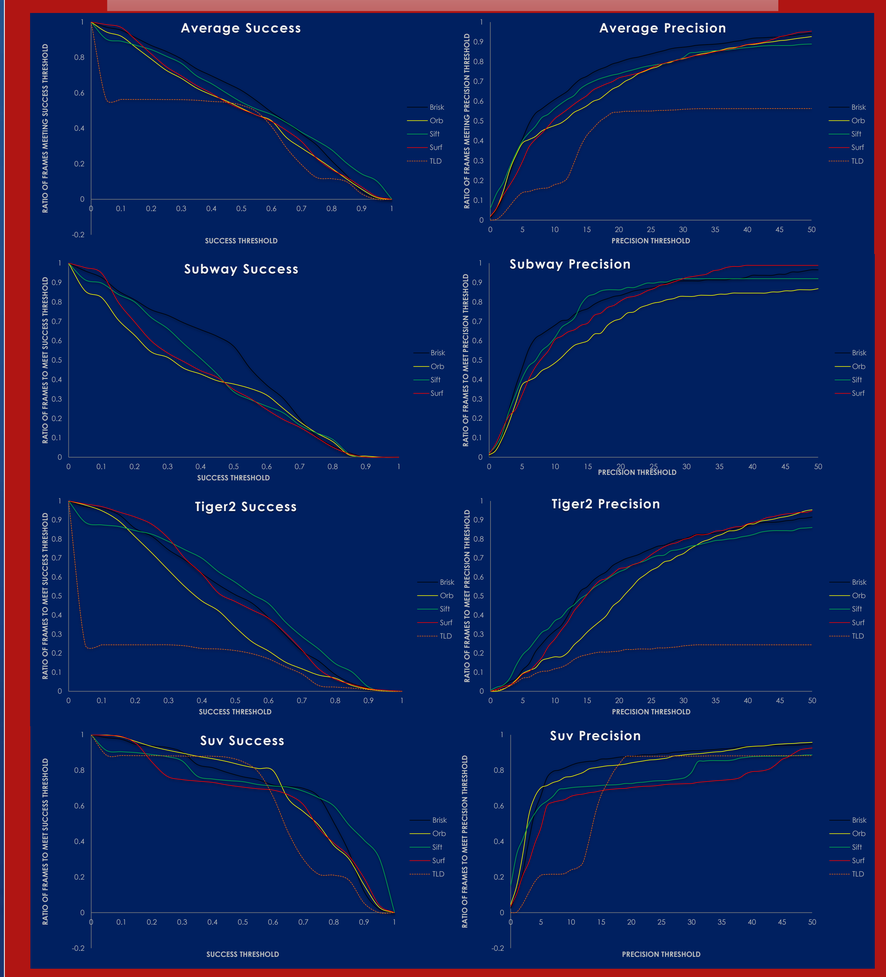
Testing

Testing the five trackers is done using a testing class. The test class must make use of a testing framework developed at Viva labs. Tests were done after each frame to determine the success and precision of a tracker. Success is defined as a trackers ability to overlap with the ground truth. Precision is defined as a trackers ability to be near the center of the ground truth.

The testing process made use of three sample benchmark video's from Y.Wu et al's paper[7] for online object tracking. The samples include ground truth information; used by the test class for comparison against the trackers results.

Occlusion Deformation Background Clutter	Illumination variation Deformation Fast Motion Out-of-Plane Rotation	Occlusion Motion Blur In-Plane Rotation Out-of-View	Occlusion In-Plane rotation Out-of-View

Results



Results Format

Given a set of threshold values for testing precision and success, the test's output is a set of ratios; one ratio for each threshold given.

$$ratio(threshold_x) = \frac{\#frame\ meeting\ threshold}{total\ \#\ of\ frames}$$

The set of ratios and thresholds are then plotted in the results figures.

Conclusions

The differences between the different versions of the CMT algorithm were detectable but not large.

- On average TLD had less success and precision than every version of CMT
- On average the version of CMT using Brisk is slightly more successful than other versions; except at lower and higher success thresholds.
- On average the version of CMT using Brisk is slightly more precise than other versions; except at very low and very high precision thresholds.
- On average the version of CMT using Sift is more successful at higher success thresholds than other versions.
- Different videos which emphasize different issues in tracking give quite different test results. Different versions of the implementation performed best or worst in different videos; relatively to other versions. For example, while the version of CMT using Brisk is better on average, it is not better than the other versions in the Tiger2 and SUV videos .

Acknowledgements

I would like to thank the University of Ottawa for making it possible to work on this project through the Undergraduate Research Opportunity program.

References

1. Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-Learning-Detection. TPAMI, 34(7), 2012.
2. G. Nehebay, R. Pflugfelder. Consensus-based Matching and Tracking of Keypoints for Object Tracking. In Winter Conference on Applications of Computer Vision, 2014.
3. S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In ICCV, 2011.
4. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski: ORB: An efficient alternative to SIFT or SURF. In ICCV, 2011.
5. David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.
6. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008
7. Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In CVPR, 2013.