Proposal: Improved ℓ_1-tracker

Saxon Jensen
University of Western Australia
email: 20945267@student.uwa.edu.au

20/05/2015

Abstract

Object tracking is becoming increasingly important in the field of computer vision. With the increased availability of powerful computers and the lowered cost of video equipment, the ability to automatically track objects in video is even more important. Previously, object tracking algorithms were unable to track objects through occlusions and through changes of appearance, such as lighting. More recently, a group of object trackers known as ℓ_1-trackers have been proposed. While research into object representations has been conducted, relatively little work has been done in learning to track an object through various changes of pose. In this paper, a key problem with ℓ_1-tracker development is highlighted.

Keywords: tracking, computer vision, object, ℓ_1-minimisation

Background

Visual object tracking aims to estimate the state of a target (eg. size, position and pose) in a sequence of video. Object tracking has become important in the field of computer vision. Increased availability of powerful computers, cheaper video cameras, along with the increased need for automated video analysis has generated a lot of interest in object tracking. Object tracking involves detecting an object, tracking it from frame to frame and analysing the behaviour of the object. This means it has many applications such as: human computer interaction for tracking hand gestures and following gaze; object recognition from motion, such as detecting human gait; automated surveillance to detect events of interest; vehicle navigation, using motion for path finding and obstacle avoidance.

Whilst many algorithms have been proposed for object tracking, it still remains a challenging problem in real world applications. A number of factors make object tracking difficult:

- erratic or complex object motion,
- real-time tracking requirements,
- significant illumination and post changes,
- object occlusion, and
- image noise.

Recently, various algorithms have been proposed that formulate object tracking as an ℓ_1-minimization calculation. In these methods, it is assumed that the appearance of the target to be tracked can be represented as a linear combination of only a few templates in a template set. In the algorithm first proposed by Mei and Ling [4], the target is represented as a linear combination of object templates and trivial templates to handle occlusion and noise. A spare representation of the target candidate is found by minimizing the following equation:

$$\|T\boldsymbol{c} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{c}\|_1$$

Where $T \in \mathbb{R}^{n\times m}$ is the template library with each column being a template. Template vectors are formed by stacking the image columns. $\boldsymbol{y}$ is the target candidate and $\boldsymbol{c}$ is the sparse coefficient vector.

Using the ℓ_1-tracker method, it is important that the template library T is updated in such a way as to overcome the real-world problems that surround object tracking. Algorithms have been proposed that address some of these problems. To handle object occlusion, image patches have been used in an object dictionary [3, 9]. In real world applications, object trackers are often required to perform in real-time. Some more efficient optimization procedures have been proposed [2, 3, 8, 9]. However, the task of updating a template set to cope with changes to the object remains less studied.

One of the largest issues faced in object tracking is ensuring the algorithm is robust. An object must be tracked through various pose changes and environmental changes such as illumination and occlusion. For example, a car being tracked may turn 90 degrees, pass from heavy shadow to light, or pass behind another object. To track an object through these changes, the algorithm must update the template library to better represent the object as it currently looks. However, the algorithm must also avoid adding templates that do not accurately represent the object, such as when it is partially occluded.

Various approaches to the template update problem have been suggested. The algorithms proposed by Bao et al. [2] and Mei and Ling [4], Mei et al. [5] update templates based on weights assigned to each template, and replace templates with low weight values by those with high weight values. In [3], templates are given a slow update confidence. The template update problem remains less well studied, with the majority of object trackers using simple intuitive approaches.

Aim

In this project, I propose to design an object tracking algorithm, focusing specifically on the method responsible for updating the template library. The tracker must accurately track an object, whilst remaining robust to changes in illumination, occlusion and other image noise.

The algorithm proposed by Xing et al. [7] makes use of the temporal information in video sequences. This has been shown to produce better tracking results than more naive template update algorithms. I aim to design an algorithm that makes use of the temporal information in video sequences. In designing an effective algorithm, approaches taken in other algorithms are considered and intuitive ways of incorporating temporal information into these approaches are investigated.

Numerous algorithms will be investigated, including:

1. Updating a simple template library using a function of time and decaying template relevance.

2. Implementing a multi-lifespan dictionary and incorporating the time function from the last item.

Tests will include comparing the algorithm I design with each of the algorithms compared to by Xing et al. [7] using the same benchmark image sequences. Algorithms will be qualitatively compared by comparing object states overlaid on image sequences. Algorithms will be quantitatively compared by computing the difference of the calculated object state from one manually annotated.

Method

The project will comprise of the following tasks:

Background Research

1. Research into the underlying mathematics.
   Initially, research will focus on gathering information on the mathematics that underpins the sparse coding object tracking method. A detailed knowledge of the mathematical background will allow constants and numerical optimisations of other algorithms to be implemented in my own algorithm.

2. Research on Similarity Functions.
   Measuring the similarity of a target candidate to a template is central to the object tracking algorithm, and the method used is varied among object trackers.

3. Research on Dictionary Learning.
   Dictionary learning is central to the updating of the template dictionary, so literature detailing techniques used for this would be greatly beneficial.

Algorithm Implementation

1. Algorithm prototype
   As outlined in the "Aim' section, an prototype of my algorithm will be written in MATLAB, one using a single dictionary, and another using the multi-lifespan dictionary.

2. Experiment and compare
   My algorithm will be compared against algorithms also available in MATLAB, such as the ℓ_1-tracker [4], incremental visual tracking [6] and the multi-instance learning tracker [1].

Software and Hardware Requirements

The project will be multi-platform, but for the purposes of experimentation, the following platform will be targeted. Currently, it is planned to use Linux with MATLAB and OpenCV. There are no special hardware requirements, although a camera may be beneficial for custom testing video sequences.

References

[1] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, Aug 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.226.

[2] C. Bao, Y. Wu, H. Ling, and H. Ji. Real time robust l1 tracker using accelerated proximal gradient approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1830–1837, June 2012. doi: 10.1109/CVPR.2012.6247881.

[3] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1822–1829, June 2012. doi: 10.1109/CVPR.2012.6247880.

[4] X. Mei and H. Ling. Robust visual tracking using ℓ_1-minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443, Sept 2009. doi: 10.1109/ICCV.2009.5459292.

[5] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient ℓ_1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264, June 2011. doi: 10.1109/CVPR.2011.5995421.

[6] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0075-7. URL `http://dx.doi.org/10.1007/s11263-007-0075-7`.

[7] J. Xing, J. Gao, B. Li, W. Hu, and S. Yan. Robust object tracking with online multi-lifespan dictionary learning. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 665–672, Dec 2013. doi: 10.1109/ICCV.2013.88.

[8] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Low-rank sparse learning for robust visual tracking. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7577 of *Lecture Notes in Computer Science*, pages 470–484. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33782-6. doi: 10.1007/978-3-642-33783-3_34. URL `http://dx.doi.org/10.1007/978-3-642-33783-3_34`.

[9] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1838–1845, June 2012. doi: 10.1109/CVPR.2012.6247882.