

COSC 444/544 - Computer Vision

Project Part I, Research Notes

February 3, 2025

Research Papers

1. Riley Eaton

1.1. *Visual Object Tracking using Adaptive Correlation Filters*

1.2. *Efficient and Practical Correlation Filter Tracking*

2. Wanju Luo

2.1. *Object detection and tracking*

2.2. *Overview and methods of correlation filter algorithms in object tracking*

3. Dichen Feng

3.1. *Moving Object Tracking and Detection in Videos using MATLAB: A Review*

3.2. *Robust visual tracking method based on Harris Hawks algorithm*

4. Henry Pak

4.1. *Recent advances of single-object tracking methods: A brief survey*

4.2. *Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics*

4.3. *Residual Network based Single Object Tracking*

5. Santam

5.1. *Paper #1 Name*

5.2. *Paper #2 Name*

Research Summaries

The following are summaries of the papers selected by each group member.

1.1 *Visual Object Tracking using Adaptive Correlation Filters*

Reviewer: Riley Eaton

Summary: In the last decade, there have been many advances in computationally efficient SOT (Single Object Tracking) techniques, with emerging strategies becoming increasingly effective for various image features. However, classical correlation filter-based trackers often have issues when encountering abrupt lighting changes, fast motions, partial occlusions, or background clutter. To address these challenges, the authors propose techniques to improve the robustness of correlation filter-based trackers. The proposed methods include dynamic model updates, feature fusion and weighted correlation, scale estimators, multi-scale correlation search regions. They call the resultant filters *Adaptive Correlation Filters*. One of said filters the authors introduce by name is *MOSSE* which can produce stable correlation filters when operating at 669 FPS, while only needing a single frame for initialization. The authors demonstrate the effectiveness of these techniques by comparing their performance to

other state-of-the-art trackers on a variety of benchmark datasets. The results show that the proposed methods outperform other trackers in terms of accuracy and robustness.

Key Points & Concepts:

- **Template matching** is one of the simplest object tracking approaches. A single patch of an image, called a *template*, is chosen. Then, in each new frame the template is slid across the entire image and a similarity measure is computed at each location. Whatever position has the highest similarity score is the new estimated position of the object.
- **Convolution** involves flipping the kernel (or filter) and sliding it across the image. The dot product of the kernel and the image is computed at each location. This is similar to the similarity measure in template matching.
- **Correlation** by contrast does not flip the filter, and is usually used to measure similarity between images or image patches. This can be used to find how closely a test patch in a new frame matches a template patch from a previous frame.
- **Frequency Domain Computations** are used to speed up the computation of the correlation between the filter and the image. A Fast Fourier Transform (FFT) is used to convert the image and filter into the frequency domain, where some computation is performed. The inverse FFT (iFFT) is then used to convert the result back to the spatial domain. This is computationally more efficient than the spatial domain computation, and speeds up matching for high frame rate applications.
- **MOSSE** stands for the Minimum Output Sum of Squared Error. It's a method for training a filter to track an object. The filter is trained on a single frame, and then used to track the object in subsequent frames. The filter is updated at each frame to account for changes in the object's appearance. It's trained by computing the FFT of the image and the FFT of the desired output, and then dividing the two to get the filter. The filter is then updated at each frame by computing the FFT of the image, multiplying it by the filter, and then computing the iFFT to get the output. The filter is then updated by comparing the output to the desired output, and adjusting the filter to minimize the error. And as noted in the paper, "despite the simplicity of the approach, tracking based on [modifications of this filter] perform well under changes in rotation, scale, lighting, and partial occlusion."

1.2 Efficient and Practical Correlation Filter Tracking

Reviewer: Riley Eaton

Summary: This work presents a traditional correlation filter tracker that focuses on an efficient, adaptive training sample update scheme to improve robustness while keeping computational demands very low. The tracker leverages classical features (such as HOG) and performs the correlation computations in the Fourier domain—techniques that are well established in computer vision. The authors cite common issues with visual tracking techniques like current discriminative correlation filters (DCF), these include boundary effects, large amounts of calculations which suffer greatly in computationally-constrained scenarios. The authors of this work proposed a way to balance performance and efficiency for object tracking using a new color ratio feature (CR) to create a DCF-based tracker embedded with CR features, namely the CRCF tracker. This new method uses an adaptive sample update scheme in which only sufficiently novel frames, identified via a difference hashing algorithm, are added to a training set, thereby preserving valuable historical appearances and preventing overfitting to transient changes. The resulting training set integrates multiple appearance snapshots of the target instead of relying on a single, continuously updated template. As a result, the filter achieves stronger generalization and is less prone to drifting errors. Finally, to handle both short-term and long-term tracking needs, the system operates in a fast short-term mode (CRCF_ATU) that addresses partial occlusion and typical frame-to-frame variations, whereas a specific re-detection strategy is activated in long-term mode (CRCF_LCT) to recover the target when it is fully occluded or exits the field of view for an

extended period using a large-scale correlation filter that has an increased awareness of the background (BACF).

Key Points & Concepts:

- **Correlation Filter:** A technique where learning the object appearance is framed as a convolution problem in the frequency domain. By using fast Fourier transforms (FFT), correlation-based tracking can be performed efficiently, usually in real time.
- **Discriminative Correlation Filters** treat correlation filtering as a supervised learning problem, training on both positive and negative samples to maintain high precision in challenging environments (occlusions, distractors, appearance variations), especially in real-time tracking. This means the filter is trained to minimize error across multiple labeled examples, rather than simply produce a peak for one template.
- **Training Sample Set:** A collection of cropped image patches (frames) where the tracker has high confidence in the object location. The filter is trained on this entire set, rather than just the most recent frame.
- **Color Ratio (CR) Features:** These are features in each frame that are represented by the ratio between the object's and background's respective color histograms. In addition to HOG and grayscale, these features help distinguish between the object and the background in many real-world scenarios. Each pixel channel is weighted by $\sqrt{p[\text{target}]/p[\text{background}]}$ where p is the color histogram.
- **Transient Changes:** These are temporary changes in the object's appearance that are not representative of the object's overall appearance.
- **Difference Hashing Algorithm (DHA):** An image-similarity measure that converts an image patch into a small binary hash matrix. The similarity of two patches is computed via the Hamming distance. Here it decides whether a new frame's appearance is different enough from stored samples to include in the training set.
- **Hamming Distance:** The measure used to compare two image patches after they are converted into binary hash matrices, where each patch is represented with an 8x8 binary matrix. The Hamming Distance between two such matrices is the count of positions at which the corresponding bits differ. This indicates how different two patches are at the bit level.
- **Short-Term tracking** assumes the target remains in view with limited full occlusion.
- **Long-Term tracking** assumes disappearances, out-of-view scenarios, or severe occlusions for any period of time.
- **Re-Detection strategy** discussed in the paper is a method to recover the target when it is fully occluded or exits the field of view for an extended period. This is done using a Background-Aware Correlation Filter (BACF).
- **Background-Aware Correlation Filter (BACF):** During re-detection, a Background-Aware Correlation Filter is trained for wider search regions using the representative samples stored in the training set. It then performs a global (or near-global) search over an expanding image window, guided by a random-walk model that increases the search region over time. If the filter's response meets a reliability threshold at some candidate location, that position is treated as the newly found target, and normal (short-term) tracking is resumed from there.

2.1 Paper Name

Reviewer: Wanjun Luo

Summary:

Key Points & Concepts:

- **Point Description**

2.2 Paper Name

Reviewer: Wanju Luo

Summary:

Key Points & Concepts:

- **Point Description**

3.1 Moving Object Tracking and Detection in Videos using MATLAB: A Review

Reviewer: Dichen Feng

Summary: This paper implemented a single object tracking system using MATLAB which focusing on handling the poor lighting conditions and occlusions.

Key Points & Concepts:

- **Color Recognition:** Use RGB and HSV color space to process video frames. HSV model is very effective on handling lighting variations.
- **Noise Reduction:** Uses a median filter to remove noise from images.
- **Approximate Median Filter:** Extract moving object from video.
- **Kalman Filter:** Predicts and refines the position of the moving object in frames.
- **Dynamic Template Matching:** Adjust the object template if its shape changes during tracking.

3.2 Robust visual tracking method based on Harris Hawks algorithm

Reviewer: Dichen Feng

Summary: The Harris Hawks Optimization (HHO) algorithm is a nature-inspired metaheuristic that mimics the cooperative hunting strategy of Harris hawks to solve optimization problems by dynamically switching between exploration and exploitation phases.

Key Points & Concepts:

- **Harris Hawks Optimization (HHO):** Inspired by the cooperative hunting strategies of Harris hawks.
 - **Exploration Phase:** Hawks search randomly for the target.
 - **Transition Phase:** Adjusts strategy based on the target's energy and location.
 - **Exploitation Phase:** Hawks converge on the prey's position through coordinated movements.
- **Object Representation:** Uses color-weighted histogram in the HSV color space to describe the object's appearance. Also uses the Bhattacharyya distance for matching the object between frames.
- **Tracking Process:** Target initialized in the first frame using a bounding box. Then the color-weighted histogram of the object is computed as a feature descriptor in the HSV color space to make the algorithm less sensitive to lighting changes.
- **Pseudo Code:**
 1. Initialize the position of N Hawks.
 2. While $i < IterMax$, do:

3. Evaluate fitness values of N Hawks.
4. Set P_{rabbit} as the best position of the rabbit from the best fitness value.
5. For $j = 1$ to N , do:
6. Update E , E_0 , and J using Eq. (2).
7. Update E using Eq. (3).
8. If $|E| \geq 1$, update Hawks position using Eq. (1).
9. If $|E| < 1$, then:
10. If $|E| \geq 0.5$ and $r \geq 0.5$, update Hawks position using Eq. (3).
11. Else if $|E| < 0.5$ and $r \geq 0.5$, update Hawks position using Eq. (5).
12. Else if $|E| \geq 0.5$ and $r < 0.5$, update Hawks position using Eq. (8).
13. Else if $|E| < 0.5$ and $r < 0.5$, update Hawks position using Eq. (9).
14. Return P_{rabbit} .

4.1 Recent advances of single-object tracking methods: A brief survey

Reviewer: Henry Pak

Summary: Published in 2021, this paper is a survey providing an overview of the advancements in single-object tracking methods. It focuses on the past decade’s progress, especially in algorithms based on **correlation filters** and **deep learning**, which have significantly improved the performance of object trackers.

Key Points & Concepts:

- **Categorization of Methods:** It categorizes single-object tracking algorithms into correlation filters-based and deep learning-based methods, also discussing earlier methods such as optical flow, filter methods, and kernel-based methods.
- **Performance Comparison:** It compares the performance of algorithms on OTB2015, VOT2016, and LaSOT datasets based on speed, accuracy, and robustness.
- **Classification of Deep Learning Methods:** It classifies deep learning methods into those based on feature extraction and those that are end-to-end methods.

4.2 Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics

Reviewer: Henry Pak

Summary: Published in 2021, this survey paper offers a comprehensive overview of single-object tracking methodologies, common datasets, and performance evaluation metrics. It categorizes tracking approaches and discusses learning-based techniques.

Key Points & Concepts:

- **Classification of Methods:** It classifies object tracking methods into four main categories: "feature-based", "segmentation-based", "estimation-based", and "learning-based methods".
- **Focus on Learning-Based Methods:** It focuses on learning-based methods, which are further divided into generative, discriminative, and reinforcement learning approaches. It also notes that deep learning is a sub-category of discriminative learning.
- **Feature-Based Methods:** It discusses feature-based methods, like those using color, texture, and optical flow.
- **Review of Datasets:** It reviews various datasets, including OTB100, OTB50, VOT (various years), ALOV300++, TempleColor128, NUS-PRO, DTB70, Nfs, UAV123, GOT-10k, LaSOT, OxUvA, TrackingNet, and YouTube-BoundingBoxes.

- **Evaluation Metrics:** It explains common evaluation metrics like center error and region overlap.
- **Challenges in Object Tracking:** It discusses the various challenges in object tracking such as illumination variations, background clutter, occlusion, and deformation.

4.3 *Residual Network based Single Object Tracking*

Reviewer: Henry Pak

Summary: Published in 2022, this paper introduces a new approach to single-object tracking using a Residual Network (RESNET-101) architecture combined with a Region Based Convolutional Neural Network (RCNN) object detector. The method, called R-SOT, is designed to predict bounding boxes around a target object in video frames using supervised learning.

Key Points & Concepts:

- **R-SOT Tracker:** The proposed R-SOT tracker uses RESNET-101 for value prediction and training.
- **RCNN Object Detector:** It employs an RCNN object detector for identifying bounding boxes.
- **Performance Evaluation:** The tracker's performance is evaluated on the **Online Object Tracking Benchmark (OTB) dataset**.
- **Bounding Box Prediction:** The bounding box prediction relies on the sequence of frames used.

5.1 *Paper Name*

Reviewer: Santam

Summary:

Key Points & Concepts:

- **Point Description**

5.2 *Paper Name*

Reviewer: Santam

Summary:

Key Points & Concepts:

- **Point Description**