Generic Object Trackers for Prolonged and Unusual Object Tracking

Alex Lin
Department of Electrical Engineering
Stanford University
alexlin0@stanford.edu

Abstract

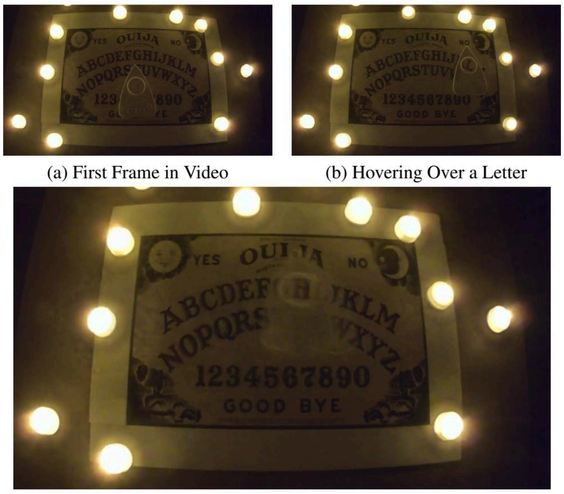
The task of generic object tracking within computer vision presents a fundamental challenge, requiring algorithms to track an object across numerous frames given only limited information in the form of an initial bounding box of the object. Previous works in generic object tracking largely focus on short-term tracking of common well-defined objects. Through a competition, we encounter a task demanding algorithms to accurately track an unusual object for two hours. To tackle this specific challenge as well as motivate more research into the task of tracking unusual objects over prolonged periods of time, for which not many datasets or work focus on, we evaluate algorithms designed using traditional computer vision techniques against state-of-the-art generic object tracking methods applied to this challenge. By designing an algorithm to extract and track these features using Canny edge detection, Hough transform, and background removal techniques, and comparing this method to four state-of-the-art object tracking methods: CSR-DCF [6], DiMP [3], ToMP [7], and TaMOs [8], we analyze the limitations of current object tracking methods applied to the task while identifying effective aspects of the methods. We find that traditional feature-based methods outperform generic object trackers in this task, though generic object trackers still achieve impressive performance of 79.7% accuracy. Further, we note the lack of datasets and works which tackle the task of prolonged and unusual object tracking, and as future work, we identify goals and assumptions of current object trackers which may be limiting the effectiveness of generic object trackers applied to this task.

1. Introduction

In a recent Capture-The-Flag (CTF) competition, a computer security competition with various challenges, one challenge required participants to track an object, a planchette, which encodes a message by moving atop a background of letters and numbers and pausing at the next character in the message. The message is encoded such that errors in object tracking or decoding are not tolerable and would lead to incorrect message decoding. The 30-fps 2-hour long video of the moving object was captured in a low-light condition with flickering lights using a poor camera, resulting in flicking effects, noise, and motion blur. We show example frames in the frame in Figure 1. This is an interesting problem because we can explore how well state-of-the-art object tracking methods apply to practical problems encountered in the real-world outside of academic datasets. Many attempts of this challenge involved manual human effort along with ad-hoc heuristics on top of traditional computer vision techniques such as Hough circle transform [5] or optical flow, due to the object being tracked being unusual as well as poor long-term accuracy of generic object tracking algorithms implemented in popular libraries. This project seeks to explore and apply state-of-the-art generic object tracking algorithms applied to this challenge and evaluate how well general state-of-the-art algorithms compare to task-specific specially designed algorithms that use traditional computer vision techniques.

Generic object tracking is a fundamental challenge in computer vision, requiring algorithms to track an object throughout many frames given only the initial bounding box of the object. The information of the bounding box, which often provides limited information about the object's appearance and motion, makes this task interesting and challenging. In real-world scenarios, objects can exhibit unpredictable behavior such as occlusions, scale variations, illumination changes, and non-rigid deformations. Moreover, there may be inperfections in the camera, such as motion blur or low resolution. Addressing these challenges while maintaining accurate tracking over extended periods is critical for various applications such as surveillance, autonomous driving, and human-computer interaction.

Many existing tracking algorithms perform well in short-term scenarios but struggle to maintain accuracy over prolonged periods of time. Prolonged tracking requires algorithms that can adapt to changes in appearance, scale, and occlusions while minimizing drift errors that accumulate



(a) First Frame in Video

(b) Hovering Over a Letter

(c) Blurry and Moving

Figure 1. Frames from Challenge Video

over time. Additionally, many traditional tracking algorithms depend on priors introduced during training or development of the system which assume the objects to be tracked are common objects. However, it is important to be able to track unusual objects not encountered during training for applications such as surveillance or autonomous driving. We explore the performance of various recent state-of-the-art object tracking algorithms applied to prolonged and unusual object tracking.

2. Related Work

One of the most popular methods for object tracking is the use of correlation filters due to its simplicity. These methods often use optimization methods to initialize a correlation filter. The correlation filter is convolved with future frames, with the convolution being applied in the fourier domain for performance, and bounding boxes are derived from the results of the convolution. Due to its simplicity yet effectiveness, these methods have been a popular subject of research in the object tracking community.

Early works such as MOSSE constructs a convolution filter based on optimizing the minimum sum of squared errors between a grayscale training input image and a desired 2D Gaussian shaped peak [4]. This method was able to achieve state-of-the-art results at the time while also achieving exceptional real-time performance, being able to run at 669 frames per second.

More recent methods improve upon MOSSE, acknowledging several limitations: the filter is spatially reliant on the bounding box and only supports grayscale channel data. The CSR-DCF method learns a spatial and reliability maps to learn and segment pixels of the image with more identifying features and to learn weights for each channel as well [6]. These techniques improve robustness and achieve state-of-the-art results while still having close to real-time performance.

The recent success in transformers have inspired machine learning techniques in the field of object tracking as well. Acknowledging the inductive biases introduced by previous filter optimization methods, ToMP employs a transformer-based model prediction module as well as a transformer-based model for bounding box prediction using ResNet as a backbone model for feature extraction and trained end-to-end. Although achieving state-of-the-art results, large CNN and transformer models require significantly more computation power and thus is not able to achieve real-time performance.

3. Approach

We create two heuristic based algorithms designed specifically for this task, based on the Hough transform [5], Canny edge detection and background removal techniques and we compare these traditional computer vision techniques to the following four state-of-the-art generic object tracking techniques: CSR-DCF [6], DiMP [3], ToMP [7], and TaMOs [8]. Additionally, we apply the background removal technique as a pre-processing step to each of the four generic object tracking techniques and evaluate each technique to gain insight into how the background affects the performance of the object tracker. After obtaining the predicted bounding boxes of the object for each frame, we run an algorithm to decode the encoded message in the challenge video. We also note that we crop the video to only the regions which the object moves to. We describe the two heuristic algorithms and each of the four generic object tracking techniques below.

3.1. Heuristic Algorithm 1

For the first heuristic method, we manually observe features of the object and apply Canny edge detection and the Hough transform to detect such features. The most notable feature of the object being tracked is a circle. We first convert the image into grayscale and blur the image using a 3x3 gaussian kernel. We then apply canny edge detection using manually tuned threshold values to extract distinctive edges. Finally, we apply the Hough circle transform using manually tuned parameters to detect circles in the image. Empirically, we find that circles in the background are also detected, and we use heuristics based on the location and radius of the circle to prune circles in the background in the event that multiple circles are detected.

3.2. Heuristic Algorithm 2

The second heuristic method will exploit the property that both the camera and the background is stationary. This is a reasonable assumption, since most surveillance cameras and backgrounds are fixed and stationary. We first extract an image of only the background by taking the median pixel value of N frames randomly sampled in the video. We then augment the video by subtracting the background image while clamping the pixels to the 0-255 range to, effectively, remove the background. With this augmented video, we then apply the first heuristic method again.

3.3. CSR-DCF

CSR-DCF [6] utilizes per-color channel correlation filters. The correlation filters are learned through minimizing a cost function, which minimizes the error between the output convolution and a desired 2D Gaussian distribution. CSR-DCF considers each color channel separately with weights per-channel to avoid the issue of color channels having different scales. CSR-DCF also learns a spatial segmentation map to identify features in the image which represent the object being tracked well, through color histograms and probability estimates. During online tracking, the bounding box detected in the current frame is used for the optimization of the weights and correlation filter.

3.4. DiMP

DiMP [3] combines multiple neural networks which predicts a convolution correlation filter and separately predicts a bounding box. The neural networks consist of a pretrained backbone feature extracting network, along with model initializer and model optimizer sub-networks which are optimized during online inference time to adapt to the object being tracked. We use the ResNet-50 backbone.

3.5. ToMP

ToMP [7] utilizes a transformer architecture based neural network to predict a correlation filter. Similar to DiMP, ToMP uses a feature extracting backbone. One critical difference in this technique, however, is that the model predicts the weights used for the correlation filter (object localization), and bounding box detection, framed as a regression problem, jointly. Another key difference is that during online tracking, the model keeps the annotated bounding box of the first frame in its training set.

3.6. TaMOs

TaMOS [8] extends ToMP to generic multi-object tracking. TaMOs is able to track multiple objects in one pass of the model. TaMOs possesses many similarities as ToMP, including jointly predicting and regressing the bounding box, as well as predicting the weights for a correlation filter. The key difference in the architecture that allows multi-object detection is that TaMOs changes the input embedding representation of the transformer such that it encodes multiple object embeddings as well as the bounding box representation. Although this technique is similar to ToMP and our task only involves tracking a single object, we hypothesize that a model which trains end-to-end on a more diverse dataset and trains for the more difficult task of multiple object tracking allows the model to be more robust and experience less overfitting, which is a trend that has been seen in large language transformers models.

4. Experiments and Results

We evaluate each method on the challenge video. The metrics we use to evaluate each method is the decoding accuracy in terms of the number of letters in the ground truth message the method was able to successfully decode and the runtime performance of the method. To compute the decoding accuracy, we use longest substring matching between the ground truth message and the decoded message to avoid skipped or extra letters impacting accuracy. We also report correct letters detected, total number of letters detected, and the precision of the method, which is the number of correct letters out of the total number of detected letters. Additionally, we qualitatively access the erroneous frames for each method and analyze the root cause for the errors.

We run a total of ten experiments, two for each of the heuristic methods, and two experiments for each of the four generic object tracking methods, one using the original video and one with the background removed from the video.

All methods were tested with an Intel(R) Xeon(R) CPU E5-2640 v3 and an NVIDIA Tesla K80 GPU. Experiment results are shown in Figure 2.

5. Analysis and Discussion

In the following sections, for each method, we analyze the performance and assess when the methods fail.

5.1. Heuristic Methods 1 and 2

The two heuristic methods perform very similarly. Examples of successful circle detection from methods 1 and 2 can be seen in Figures 3 and 4 respectively. We observe that these methods detect the object very accurately when the object is moving slowly or not moving. However, these methods fail to detect the object when there is motion blur. This is due to the edges of the circle on the object being blurred which causes canny edge detection and Hough transform to fail to detect the circle. However, we note that due to the decoding procedure not needing to detect letters while the object is moving, the extremely high object location detection of these two methods result in nearly perfect

Figure 2. Decoding Performance Metrics

Method	Decoding Accuracy	Correct Letters	Total Detected Letters	Runtime per frame (s)	Precision
Heuristic 1	**0.9984**	**5579**	**5580**	**0.00478**	0.9998
Heuristic 2	0.9873	5517	5518	0.00586	0.9998
CSR-DCF	0.08536	477	477	**0.0301**	1
CSR-DCF RM BG	0.01306	73	73	0.0302	1
DiMP	0.03221	180	180	0.0621	1
DiMP RM BG	0.3300	1844	1861	0.0622	0.9909
Tomp	0.7092	3963	3978	0.1038	0.9962
Tomp RM BG	0.4608	2575	2629	0.1038	0.9795
TaMOs	**0.7974**	**4456**	**4481**	0.4864	0.9944
TaMOs RM BG	0.3631	2029	2047	0.4864	0.9912

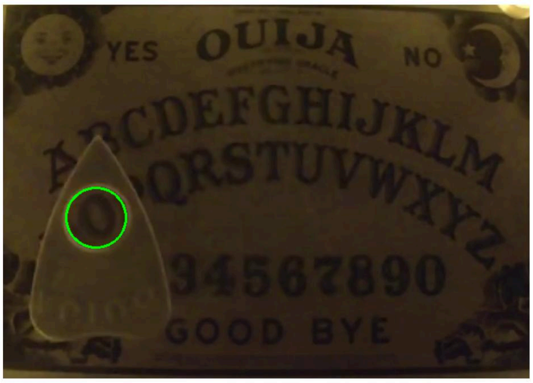


Figure 3. Heuristic Method 1 Circle Detection

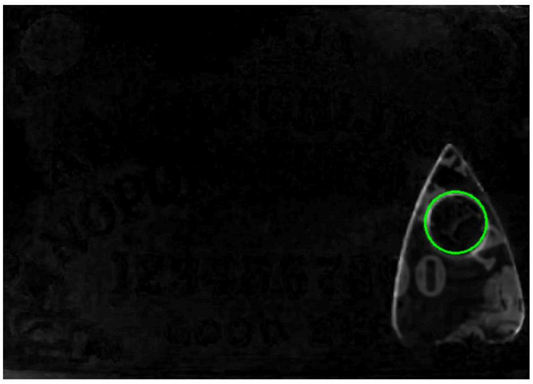


Figure 4. Heuristic Method 2 Circle Detection

decoding accuracy.

We also observe that heuristic method 2 detects the object in approximately 5% less frames compared to the first method. Observing the differing frames, we note that the removal of the background impacts the clarity of the edges on the object, thus causing Canny edge detection and Hough transform to fail more often.

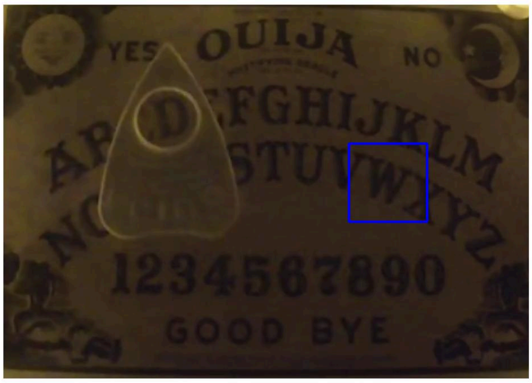


Figure 5. CSR-DCF with Bounding Box Locked on Background

5.2. CSR-DCF

For CSR-DCF without background removal, we observe two behaviors: the bounding box locks onto the background eventually, and the bounding box drifts from the object slightly. This may be due to a fundamental issue of correlation filters, which is that correlation filters can result in false positives with the background in certain cases. This effect is exacerbated by the method CSR-DCF uses for online tracking, which primarily uses the results of the current frame to track future frames. So, once a single false positive correlation error occurs, the tracker fixates on the erroneous object. In this case, the tracker fixates on the static background, and thus remains fixated there forever. This can be seen in Figure 5 where the bounding box remains fixated on the background, which causes the low accuracy from this

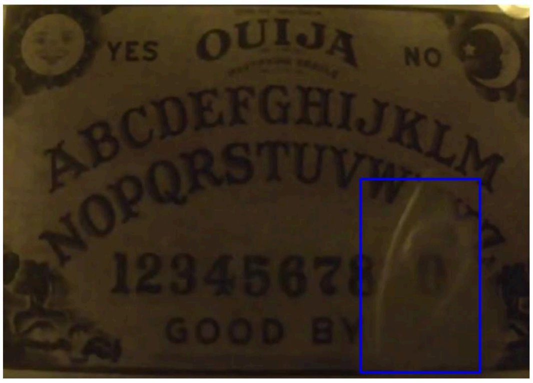


Figure 6. DiMP with Deforming Bounding Box

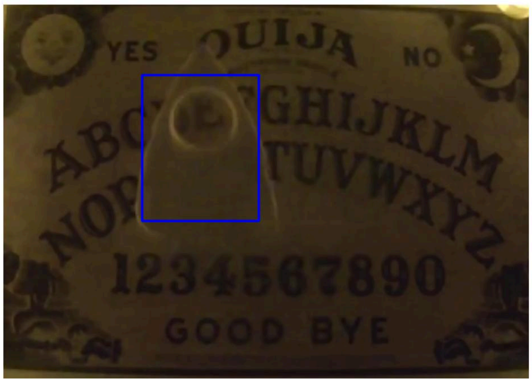


Figure 7. ToMP with Deforming Bounding Box

method.

For CSR-DCF with background removal, we observe only the drifting behavior of the bounding box. This makes sense, since the background is almost entirely dark, which causes the correlation filter to rarely mistake the background for the object. However, we note that the bounding box drifts away from the center of the object rapidly due to the contrast between the object and the background being lower and thus the correlation filter is unable to find clear peaks and drifts away from the center of the object. The low accuracy of this method is caused by this rapid drift.

5.3. DiMP

For DiMP without background removal, we observe three behaviors: the bounding box changes shape throughout a few frames, the bounding box changes position slightly even between frames which are nearly identical, and that after a few minutes of footage, the bounding box locks onto the entire frame. We hypothesize that this is due to the bounding box prediction model being a separate model in DiMP, and that the bounding box prediction model is not robust enough. This behavior of the bounding box changing shapes can be seen in Figure 6. However, once the bounding box expands to a certain extent, it locks onto the entire background, thus causing the low accuracy obtained by this method.

For DiMP with background removal, we observe the same first two behaviors as DiMP. However, the bounding box does not ever lock onto the entire frame, resulting in much higher accuracy. Because the background is removed, the bounding box prediction model is able to work much better and we observed drastically improved accuracy, with the method able to recover approximately one-third of the letters.

5.4. ToMP

For ToMP without background removal, we observe the bounding box changes shape and moves around slightly during frames where the object is rapidly moving, however generally the bounding box still remains centered on the object. During frames which do not change and the object is stationary, the bounding box remains the same. However, we also observe that sometimes the bounding box expands and shrinks to adapt its shape to encompass other features of the object, which significantly changes the aspect ratio of the bounding box, as demonstrated in Figure 7. This method was able to achieve great accuracy of 70.9%, with most of the errors due to the bounding box suddenly changing shapes.

For ToMP with background removal, we observe the same trends, except that the bounding box expands and contracts more chaotically and rapidly, leading to the significantly lower accuracy.

5.5. TaMOs

For TaMOs without background removal, we observe similar trends as ToMP. The bounding box changes shape during moving frames, but generally remains centered around the object. In frames where the object is moving, we do observe the bounding box drifting slightly, as demonstrated in Figure 8. However, we observe that unlike ToMP, the aspect ratio of the bounding box remains constant, save for a few outlier frames, and that in most cases, the bounding box does not expand to capture other features. Indeed, due to the higher stability and robustness of the bounding box prediction, this method obtains the highest accuracy out of all the generic object tracker methods, with nearly 80% accuracy.

For TaMOS with background removal, we observe chaotic results where the bounding box deforms signifi-

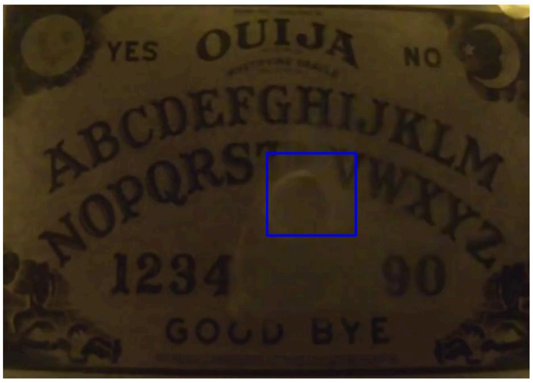


Figure 8. TaMOs with Drifting Bounding Box

cantly during moving frames and does not remain centered around the object, but also suddenly locks onto the background in many frames. We hypothesize that this is due to the nature of the TaMOs model being able to detect multiple objects because given that the contrast between the background and the object decreased significantly when the background is removed, the model mistakes the background for the object more often.

6. Conclusion

We evaluate algorithms designed using traditional computer vision techniques against state-of-the-art generic object tracking methods to a task which requires the tracking of an unusual object not seen during offline training time for a prolonged amount of time. We find that exploiting human observation and specially designing an algorithm for the task using traditional techniques still outperforms generic object trackers. Most notably, hand-designed features and feature detection is still an effective method to detect and track an object. However, we do see incredible performance from generic object trackers, with TaMOs being able to achieve nearly 80% accuracy on a task it was not trained for and data it had never seen before.

Even though the performance of different generic object trackers greatly vary depending on assumptions made in the design of the method or consequences of the algorithm and architecture the tracker is based off of, amongst the generic object tracking methods we evaluated, we observe three conclusions: jointly predicting a correlation filter and bounding box makes bounding box predictions much more robust, transformers based architectures trained offline on larger amounts of data may lead to more robustness, and online training should maintain larger amounts of historical information to avoid one-time mistakes propagating indefinitely to future frames.

7. Future Work and Limitations

Future work include designing object tracker models for the specific task of prolonged and unusual object tracking. Specifically, we note three goals which conflict with the quality that may be achieved on this task: current generic object tracking methods aim for real-time tracking, methods tend to focus on short-term tracking, and methods assume objects can disappear and reappear. By relaxing the real-time constraint and assuming the object tracker will be applied to long-term tracking, we can conceive methods which run more training optimization steps at inference time using frames it has already successfully tracked. By assuming long-term tracking as opposed to short-term tracking, we can utilize longer histories of data during inference. By assuming objects cannot disappear and reappear, the model is able to make a best-guess attempt, as well as model object motion using methods such as SORT [2].

One limitation of hand-crafted feature detection is that in cases of object deformation, such as in the case of motion blur, the method is not able to robustly detect features. Future work would include creating datasets for the task of prolonged and unusual object dtectin and design metrics which can evaluate these properties.

Code is provided in [1].

References

[1] https://github.com/voidmercy/cs231a-final-project. 6

[2] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. pages 3464–3468, 09 2016. 6

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. pages 6181–6190, 10 2019. 1, 2, 3

[4] David Bolme, J. Beveridge, Bruce Draper, and Yui Lui. Visual object tracking using adaptive correlation filters. pages 2544–2550, 06 2010. 2

[5] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, jan 1972. 1, 2

[6] Alan Lukežič, Tomáš Vojíř, Luka Čehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. *International Journal of Computer Vision*, 126, 07 2018. 1, 2, 3

[7] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Paudel, Fisher Yu, and Luc Gool. Transforming model prediction for tracking. pages 8721–8730, 06 2022. 1, 2, 3

[8] Christoph Mayer, Martin Danelljan, Ming-Hsuan Yang, Vittorio Ferrari, Luc Gool, and Alina Kuznetsova. Beyond sot: Tracking multiple generic objects at once. pages 6812–6822, 01 2024. 1, 2, 3