Monterey Bay Aquarium
Research Institute

Developing benchmarks for performance evaluation of tracking algorithms for use in deep-sea creature detection and tracking.

Lee, Peyton, University of Washington

Mentors: Danelle Cline & Duane Edgington

Summer 2020

Keywords: multiple object tracking, computer vision, underwater computer vision, machine learning

1. ABSTRACT

The Monterey Bay Aquarium Research Institute (MBARI) maintains a database of over 25,000 hours of deep-sea video footage, representing a considerable commitment of time, energy, and expertise in order to analyze. Advances in machine learning and computer vision have enabled the exploration of an annotation assistance workflow, utilizing a detection model and tracking algorithm in tandem to annotate and classify deep-sea creatures automatically. To determine the optimal configuration of tracking algorithms for deep-sea applications, three trackers from the OpenCV library (Median Flow, Kernelized Correlation Filters, and Tracking, Learning, and Detection) were tested on a novel video benchmark consisting of deep-sea benthic transect footage. Performance metrics were calculated using CLEAR MOT metrics. Overall, Median Flow had the most consistent performance across the tested trackers and performed positively with both varying stride and resolution. Operating Median Flow at a decreased resolution of 240 x 135 and a stride of between 1 to 10 frames yields an efficient and performant tracking system.

2. INTRODUCTION

The Monterey Bay Aquarium Research Institute (MBARI) has been actively collecting video footage of deep-sea environments for 32 years. This collection consists of over 25,000 hours of deep-sea benthic and midwater footage and is accessed and annotated through the Video Annotation Reference System (VARS) (Video Annotation and Reference System, 2015). This video footage is used to quantify species composition at a given site and estimate biodiversity and abundance. Currently, video is annotated by members of the video annotation lab at MBARI, and processing through MBARI's video archive represents an immense challenge in time investment.

The VARS Annotation Assistance (VAA) project emerged in recent years to assist the annotation process using computer vision and machine learning. Initial efforts by Yee et al. (2017) led to the use of a Faster R-CNN model for deep-sea creature detection and the development of a training dataset of benthic annotations (Ren et al. 2016).

While our detection model can effectively predict species that appear in a given video frame, it cannot count unique instances of creatures across an entire video sequence. To do so, we require a tracking algorithm that matches detections across frames to a unique ID (Fig. 1). Modern tracking algorithms are typically much faster than detection models (Luo et al. 2018). Many open-source video trackers are available through libraries like OpenCV, which provides implementations of trackers like Median Flow, Kernelized Correlation Filters, and the Tracking, Learning, and Detection tracker.

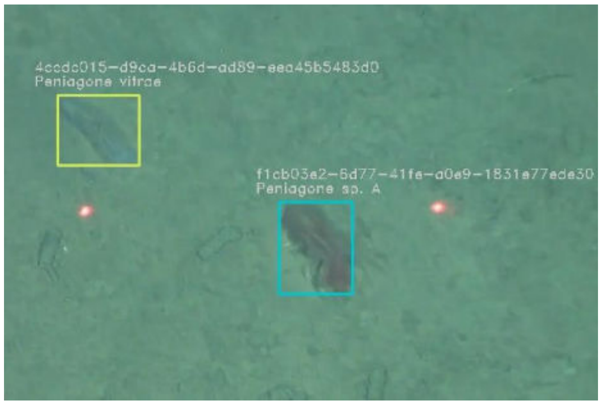

4ccdc015-d9ca-4b6d-ad89-eea45b5483d0
Peniagone vitrae
f1cb03e2-6d77-41fe-a0e9-1831e77ede30
Peniagone sp. A

Figure 1: An example frame of benthic footage with bounding boxes (localizations) drawn for detected creatures. A unique identifier is given to each object by the tracking algorithm.

Due to the abundance of tracking algorithms, several tracking benchmarks have gained popularity in recent years. One of the most notable is the MOTChallenge, which tests trackers using pedestrian and crowd footage (Milan et al. 2016). However, marine environments offer significant challenges that are not adequately addressed by existing tracking benchmarks. Ocean environments can vary widely in lighting, background, and optical clarity. Species may be only briefly visible, partially or wholly transparent, and often deform when exposed to backwash from observation platforms. These problems are exacerbated in deep-sea environments, where light is scarce and creatures are often darkly-colored or transparent for camouflage. While there are existing underwater benchmarks, these datasets are either not publicly available, are biased towards well-lit, shallow environments, or include footage from artificial settings (such as pools or video games) (Kezebou et al. 2019).

This study developed a novel computer vision benchmark to evaluate the performance and runtime cost of three different tracking algorithms in deep-sea benthic environments. We utilized benthic transect footage to generate performance metrics on different trackers across varying tracker configurations.

3. MATERIALS AND METHODS

3.1 DETECTOR TRAINING

A Faster R-CNN detection model was chosen for the annotation process, as initially described by Ren et al. (2016). The model was trained on an AWS SageMaker instance for 50,000 epochs. The training data consisted of eight hours of pre-annotated, benthic transect frames captured by ROV at Station M off the coast of Monterey, CA, as described by Yee et al. (2017). The training dataset included annotations for 17 common benthic species (Table 1).

Table 1: The complete list of all creature concepts that were present in the training dataset. The counts of their appearances in the training dataset and in the benchmark video (by frame) are listed, as well as the number of unique individuals present in the benchmark video.

Classification	Appearances (by frame) in training set	Appearances in benchmark (by frame)	Appearances (by individual) in benchmark
Benthocodon	569	214	5
Cystechinus loveni	174	1104	8
Echinocrepis rostrata	82	452	5
Elpidia	493	224	2
Fungiacyathus (Bathyactis) marenzelleri	64	1448	7
Oneirophanta mutabilis complex	6	431	2
Peniagone papillata	14	588	4
Peniagone sp. 1	14	36	1
Peniagone sp. 2	12	108	1
Peniagone sp. A	1108	6629	48
Peniagone vitrea	360	2725	23
Scotoplanes globosa	365	193	2
Tjalfiella	230	968	9
Synallactidae	30	0	0
Hexactinellida sp. 1		0	0
Coryphaenoides	8	0	0

The trained model was then used to generate inferences for a 15-minute segment of full-resolution (1920x1080, 30fps) benthic transect footage. The model-generated inferences were processed by a Median Flow (MF) tracker and verified to ensure accuracy using the open-source Computer Vision Annotation Tool (CVAT). False positives were removed, missed objects were localized, and bounding boxes were redrawn to fit identified species' central mass. Fragmented tracks were joined, and misidentified species were corrected. This curated footage served as our ground-truth benchmark data.

3.2 TRACKER CONFIGURATION

For this experiment, we used five visual tracking algorithms that were publicly available through the OpenCV library; Median Flow (MF), Kernelized Correlation Filters (KCF), and Tracking, Learning, and Detection (TLD).

Each tracker was run times on the 15-minute, model-generated inference data with varying span (how often, in frames, the tracker was seeded with detections) and resolution (Table 2). The span was varied to simulate running the trackers on real-time detections, as the detection model can process approximately three frames per second. A stride of 10 was used to approximate real-time requirements. The time required for each tracker to complete the inference job was timed and recorded.

Table 2: The stride and resolution used for each tracker run.

Stride	Time between detections(s)
1	0.033
10	0.33
20	0.66

Resolution Factor	Video Dimensions
0.5	960 x 540
0.25	480 x 270
0.125	240 x 135

3.3 EVALUATION

The tracker's performance was calculated as multiple object tracking precision (MOTP) and accuracy (MOTA), as defined by the CLEAR MOT metrics (Milan et al. 2016; Bernardin & Stiefelhagen 2008). We leveraged the py-motmetrics library, a CLEAR MOT metrics library developed in Python, to aid in the matching and metrics-generation process (Heindl et al. 2020).

For each tracker run, the tracker's XML output for each frame was converted into a single XML document (hypothesis). It was then compared against the ground-truth tracks. In each frame, the cost to match each truth object to each hypothesis object was calculated as intersection over union, or $cost(H,T) = 1 - \frac{Area(H \cap T)}{Area(H \cup T)}$. Matches were made to minimize the total cost within the frame.

Hypothesis objects that did not have a matched truth object were considered false positives (FP), while truth objects that did not have a matched hypothesis object were considered misses. ID switches (IDSW) were counted for the sequence, incremented once each time a different matching between a truth and hypothesis object was observed than was previously made.

Multiple Object Tracking Precision (MOTP) was calculated as one minus the sum cost of all matches made, averaged by the number of matches made.

$$MOTP = 1 - \frac{cost\ of\ all\ matches\ in\ sequence}{total\ matches\ in\ sequence}$$

Multiple Object Tracking Accuracy (MOTA) was calculated as one minus the sum of the total false positives, misses, and ID switches for the entire sequence averaged over the total number of frames.

$$MOTA = 1 - \frac{FP + MISS + IDSW}{total\ frames\ in\ sequence}$$

4. RESULTS (Normal, Times New Roman, 12 pt, bold)

4.1 COMPUTATION TIME

There was some minor variation in computation time across stride for the same resolution, but across different resolutions, the change in performance was more pronounced (Fig. 2). Tracking for lower resolutions was overall much faster to compute (around 2-3 minutes per minute of footage) than tracking for half or full-resolution video footage, which took up to 23 minutes to process per minute of footage. MF took the least computation time across most trials, while KCF took slightly longer but achieved similar performance. TLD was generally slower than both MF and KCF at higher strides, which was especially visible at full resolutions.

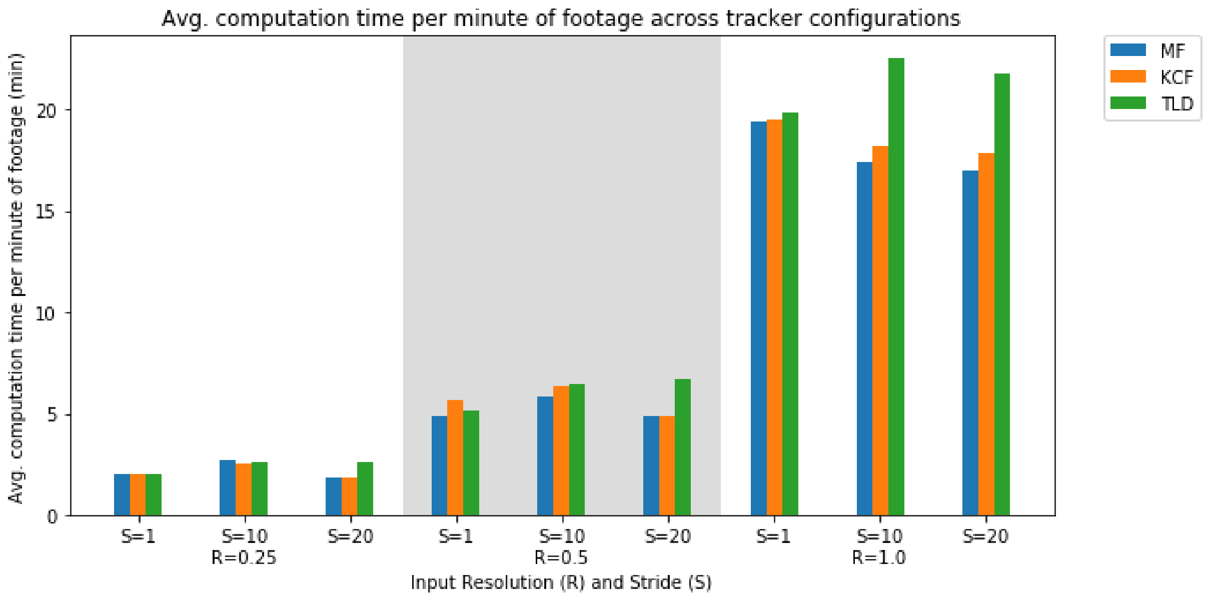


Figure 2: The average computation time for a minute of benthic transect footage for each of the three trackers. Bars are grouped by resolution and ascending stride.

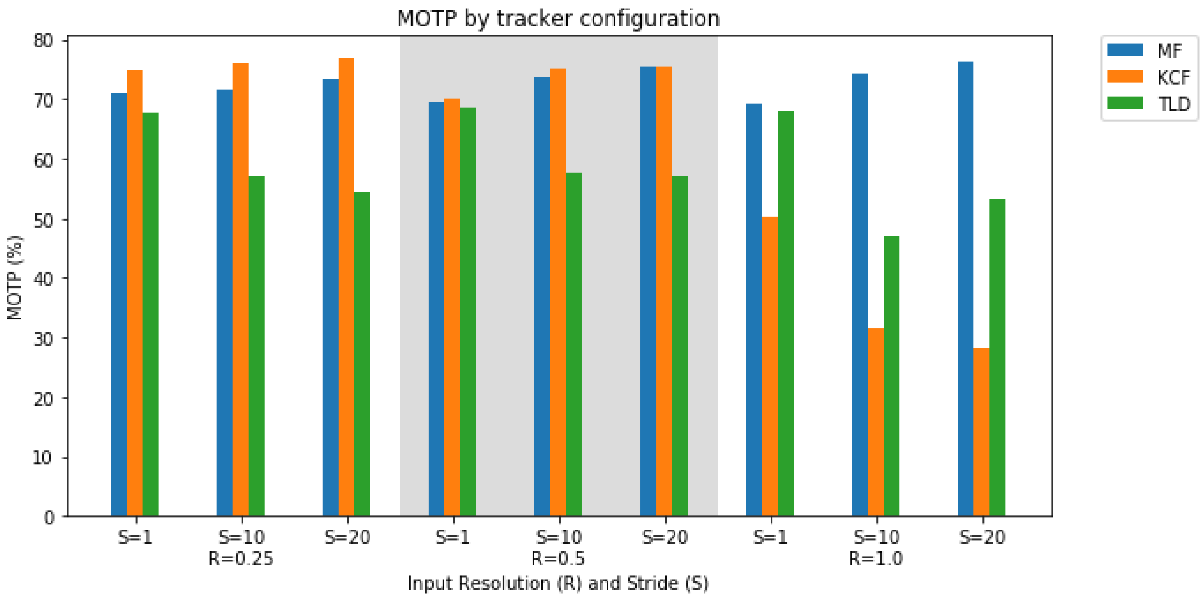


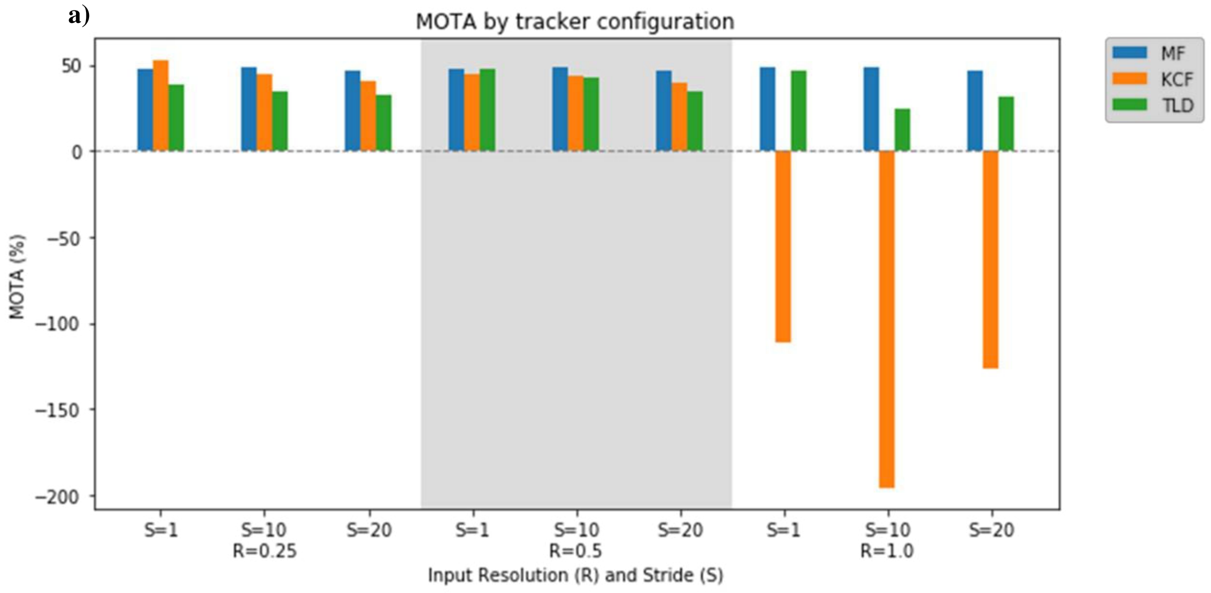
Figure 3: The calculated multiple object tracking precision (MOTP) as a percentage. Runs are grouped by resolution and ordered by ascending stride.

4.2 MULTIPLE OBJECT TRACKING PRECISION & ACCURACY

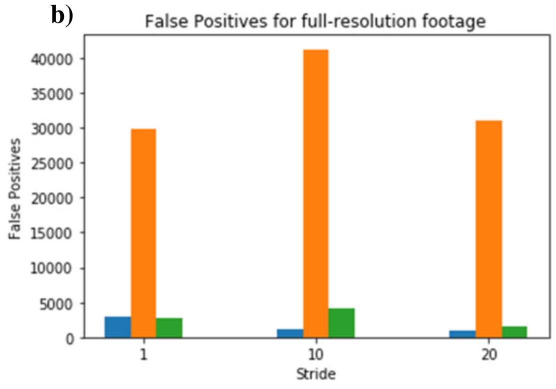
For lower resolutions, both MF and KCF improved in precision with a longer stride (Fig. 3). However, KCF decreased in precision at higher (full) resolutions, underperforming when compared to both TLD and MF. TLD generally decreased in precision as stride increased. Overall, MF seemed to be the most versatile, maintaining a consistent precision across resolutions.

In terms of MOTA, MF scored consistently across varying strides and varying resolutions, while KCF's performance varied widely across resolutions (Fig. 4). At lower resolutions, its performance exceeded that of MF, but at higher resolutions, it scored a negative MOTA.

a)



b)



c)

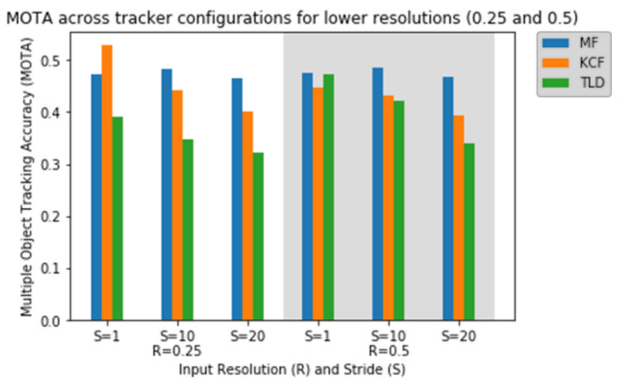


Figure 4: The calculated multiple object tracking accuracy (MOTA) as a percentage for **a)** all resolutions and **c)** lower resolutions (0.25 and 0.5) in order to better show differences in accuracy. False positives for run configurations where resolution was 1.0 are shown by **b)**. Runs are grouped by resolution and ordered by ascending stride.

Closer investigation revealed that the low MOTA score was caused by the tens of thousands of false positives generated by the KCF tracker when running at higher resolutions (Fig. 4b). These false positives were correlated disproportionately with the classifications for *Peniagone vitrea*. For example, *P. vitrea* represented approximately 18.02% of the detections in the benchmark but represented 65.66% of false positives for the full resolution, stride-10 KCF tracker output. TLD achieved a similar or worse score than MF, which became more pronounced as stride increased.

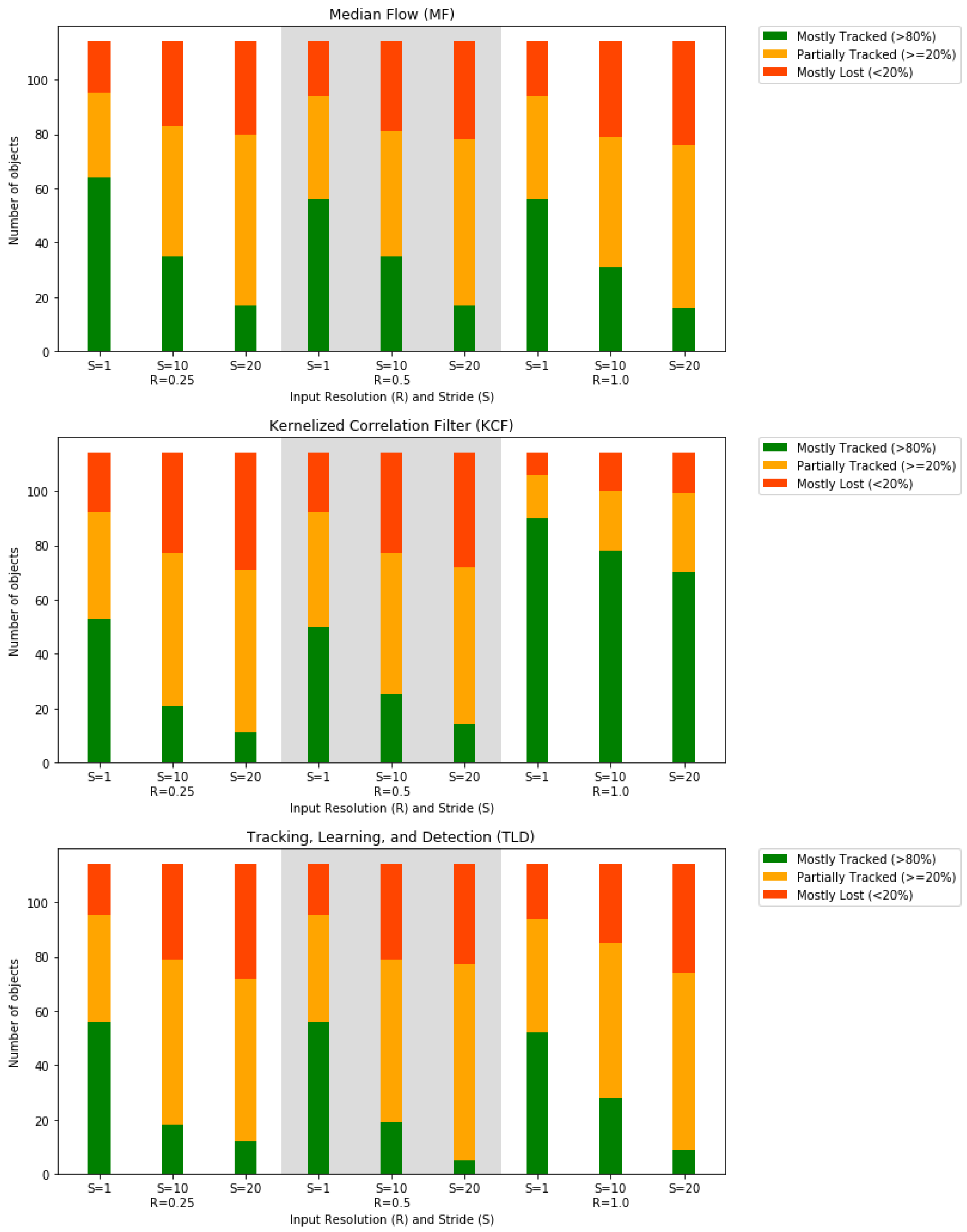


Figure 5: The tracking coverage for the three trackers. Objects that were tracked for more than 80% of the sequence were considered Mostly Tracked, between 20-80% were considered Partially Tracked, and less than 20% were considered Mostly Lost.

4.3 OBJECT TRACKING COVERAGE

An additional metric was calculated for each tracker configuration, measuring how long the tracker tracked a given truth object. Objects tracked for over 80% of their lifetime were labeled mostly tracked, while objects tracked less than 20% of their lifetime were labeled mostly lost. Any objects that fell between 20% and 80% coverage were labeled partially tracked (Fig. 5). All three trackers had a higher proportion of lost objects as stride increased, and a corresponding decrease in mostly and partially tracked objects. However, TLD and MF were reasonably consistent across resolutions, while both KCF had a higher proportion of mostly tracked objects when processing full-resolution footage.

5. DISCUSSION

The difference in runtime between resolutions is consistent with the amount of additional data that the trackers must process, which increases exponentially with resolution. The high time cost required to process high-resolution footage suggests that it would not be cost-effective to process footage at full resolution. While KCF had a greater coverage at higher resolutions, the significant computation time associated with that performance renders it impractical for use. Thus, trackers should be run at a lower resolution whenever possible. Additionally, the decrease in tracking coverage with increasing stride suggests that trackers should be given the smallest stride feasible for the application. Operating at a lower stride should also not impact runtime, as at lower resolutions, there was no significant difference in runtime across strides.

Overall, MF performed relatively consistently across various resolutions, varying only slightly in MOTP and MOTA. While other trackers (KCF) outperformed it in certain situations, Median Flow features a slightly faster computation time and similar MOTA, MOTP, and tracking coverage across resolution variance. This result diverges from the findings of a study by Lehtola et al. (2017) on the performance of visual tracking algorithms on embedded systems. This study did confirm that the KCF tracker was marginally slower than MF, but KCF outperformed MF in terms of the Jaccard Index, which is analogous to MOTP. However, the substantial improvement in precision KCF gave over MF that Lehtola et al. (2017) described was not present in our analysis. This may be due to our dataset, where objects move linearly with the transect, a task which Median Flow is designed for (Kalal et al. 2010).

Our study also suggests that MOTP is a less useful metric than MOTA, as it only describes how well objects were localized when matched. Especially for MBARI's deep-sea footage analysis, accurately counting and tracking creatures is more critical than the system's ability to localize creatures precisely.

However, MOTA also suffers from a lack of transparency; because it is the aggregate of several other metrics, it may hide other patterns in tracker performance. The MOTA stayed relatively constant for MF over different strides (Fig. 4c), but a closer analysis of tracking coverage shows that MF had a decreased ability to track objects at higher strides (Fig. 5). However, a decreasing number of false positives counterbalanced the missed objects at higher strides (Fig. 4b). Thus, MOTA cannot fully capture the nuances of a tracker's behavior, and other base metrics (such as coverage and false positives) may yield more detailed comparisons.

6. RECOMMENDATIONS

For analyzing benthic transect footage, Median Flow seems to be the strongest candidate, as has a fast runtime and generally stable and positive performance in both MOTP and MOTA across varying strides and resolutions. Its flexibility means that it can be deployed in a broader range of configurations than KCF or TLD, and can be run at lower resolutions and longer strides while still achieving similar results to processing full-resolution footage. However, as Median Flow still takes roughly 2 minutes to compute per minute of footage, it may be advisable to test the tracker at lower resolutions (such as 1/8) if applied to real-time applications.

However, to confirm these results, expanding the scale of the benchmark and running multiple trials is necessary. Increasing the number of benthic transect benchmark videos available to test with would decrease the chance that random factors in the single video influenced our results. Using more videos would also allow us to test our trackers in a broader range of scenarios. Additionally, due to the shared nature of the systems used to run the trackers, the time cost may have been artificially increased or decreased due to system load. Running additional trials for each benchmark video would decrease the associated random error. However, this would also take a great deal of computation time.

Additional trackers could be viable candidates for further exploration. Two more trackers that are provided by OpenCV are the Minimum Output Sum of Squared Error (MOSSE) and the Discriminative Correlation Filter with Channel and Spatial Reliability (CSRT) trackers.

Finally, we expect that the trackers that performed well on the benthic transect will differ from the trackers that perform well on midwater footage. This is due to several factors— shorter intervals of appearance for creatures, limited or inconsistent lighting, the large number of particles present in the water column, and transparent, darkly-colored, or minute creatures—all of which act as additional challenges to tracking and detection systems aimed at midwater environments.

7. ACKNOWLEDGEMENTS

I would like to thank my mentors, Danelle Cline and Duane Edgington, for their incredible support and encouragement during this internship. Thank you to George Matsumoto, Megan Bassett, and Tatjana Ellis for their flexibility and mentorship within the internship program, and Larissa Lemon and the MBARI Video Lab for their assistance in the annotation process.

8. REFERENCES

Bernardin, K., & Stiefelhagen, R. (2008). Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. EURASIP Journal on Image and Video Processing, 2008, 1–10. https://doi.org/10.1155/2008/246309

Heindl, C., Valmadre, J., & Toka. (2020). Cheind/py-motmetrics [Python]. https://github.com/cheind/py-motmetrics

Kalal, Z., Mikolajczyk, K., & Matas, J. (2010). Forward-Backward Error: Automatic Detection of Tracking Failures. 2010 20th International Conference on Pattern Recognition, 2756–2759. https://doi.org/10.1109/ICPR.2010.675

Kezebou, L., Oludare, V., Panetta, K., & Agaian, S. S. (2019). Underwater Object Tracking Benchmark and Dataset. 2019 IEEE International Symposium on Technologies for Homeland Security (HST), **1–6**. https://doi.org/10.1109/HST47167.2019.9032954

Lehtola, V., Huttunen, H., Christophe, F., & Mikkonen, T. (2017). Evaluation of Visual Tracking Algorithms for Embedded Devices. 88–97. https://doi.org/10.1007/978-3-319-59126-1_8

Luo, H., Xie, W., Wang, X., & Zeng, W. (2018). Detect or Track: Towards Cost-Effective Video Object Detection/Tracking. ArXiv:1811.05340 [Cs]. http://arxiv.org/abs/1811.05340

Milan, A., Leal-Taixe, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A Benchmark for Multi-Object Tracking. ArXiv:1603.00831 [Cs]. http://arxiv.org/abs/1603.00831

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. ArXiv:1506.01497 [Cs]. http://arxiv.org/abs/1506.01497

Video Annotation and Reference System. (2015, April 24). MBARI. https://www.mbari.org/products/research-software/video-annotation-and-reference-system-vars/

Yee, N., Cline, D., & Edgington, D. (2017). Workflows for Automated Detection and Classification of Unlabeled Deep Sea Imagery. https://www.mbari.org/2017-intern-papers/

9. ADDITIONAL RESOURCES

https://github.com/plee-mbari/cv-evaluator

Code written to convert and evaluate tracker output using the cheind/py-motmetrics library.

https://bitbucket.org/plee-mbari/deepsea-track-notebook/src/master/
A series of Python notebooks that run the trackers and displays the metrics evaluation.