

Real-Time Multi-Diver Tracking and Re-identification for Underwater Human-Robot Collaboration

Karin de Langis¹ and Junaed Sattar² *

Abstract

Autonomous underwater robots working with teams of human divers may need to distinguish between different divers, *e.g.*, to recognize a lead diver or to follow a specific team member. This paper describes a technique that enables autonomous underwater robots to track divers in real time as well as to reidentify them. The approach is an extension of Simple Online Realtime Tracking (SORT) with an appearance metric (deep SORT). Initial diver detection is performed with a custom CNN designed for realtime diver detection, and appearance features are subsequently extracted for each detected diver. Next, realtime tracking-by-detection is performed with an extension of the deep SORT algorithm. We evaluate this technique on a series of videos of divers performing human-robot collaborative tasks and show that our methods result in more divers being accurately identified during tracking. We also discuss the practical considerations of applying multi-person tracking to on-board autonomous robot operations, and we consider how failure cases can be addressed during on-board tracking.

1 INTRODUCTION

The state of the art in multi-person visual tracking has greatly improved in both speed and accuracy in recent years [1–3]. These improvements make multi-person trackers viable for use on realtime robotic platforms. However, utilizing multi-person tracking algorithms onboard autonomous robots, particularly in adverse conditions, is still an under-explored area [4]. In this paper, we propose a realtime multi-person tracker suitable for autonomous underwater robots.

This work was motivated by the need for underwater robots to distinguish between different human ‘teammates’

*The authors are with the Department of Computer Science and Engineering, Minnesota Robotics Institute, University of Minnesota Twin Cities, Minneapolis, MN, USA. {¹dento019, ²junaed} at umn.edu.

in order to improve underwater human-robot collaboration. Underwater robots are utilized for a wide range of tasks, including data collection, ecological mapping, and wreck investigations (*e.g.*, [5–7]). These tasks frequently require collaboration between robots and human divers. When the robot collaborates with a team of divers, it is highly useful for the robot to be able to identify different divers: for example, the robot may need to follow a specific diver or to recognize a lead diver from whom to take instructions.

In order to distinguish between different divers, the robot must continually detect divers who are present in images received from its cameras. The robot must also have a method to “identify” detections; that is, the robot must keep track of each person it has seen and determine whether a detection corresponds to one of those people. This is roughly equivalent to the multi-object tracking (MOT) problem, which has been extensively studied in computer vision.

Most MOT research has focused on tracking pedestrians [1–3]. In this work, we adapt these pedestrian-focused strategies for use in an underwater human-robot collaboration scenario. In particular, this involves shifting the tracker’s focus from tracking many people that come and go throughout a crowded scene, to tracking a small group of people that may leave the robot’s field of view for an arbitrary period of time, but remain in the scene indefinitely. Also, human body postures are predominantly in a horizontal orientation during the diver tracking scenario, which is not the case for pedestrians. Our tracking problem has some difficulties that are not present in the typical pedestrian tracking scenarios: there is the inherent difficulty in detecting divers (see Section 2 for a discussion), as well as the difficulty in distinguishing between two divers given the poor visibility conditions underwater and similarities in divers’ SCUBA gear (see Figure 1). However, in our problem we only need to track a few divers at a time, whereas typical MOT sequences contain dozens of people in a given frame.

Our approach is tracking-by-detection, which is the current leading MOT paradigm. In tracking-by-detection, the tracker first performs person detection on each image. The tracker aims to match each of these detections to the correct



Figure 1: Two divers collaborating with an underwater robot. It is desirable for the robot to be able to uniquely identify its human partners; however, the divers’ similar gear and the poor visibility conditions underwater make this difficult.



Figure 2: Exemplary output of our tracker on an underwater scene with multiple divers present. Two divers are detected and determined to belong to tracks “1” and “3.”

track, where each track represents a unique person. Each track typically models the person’s trajectory and/or appearance in order to assist the tracker in matching detections to tracks.

Our method is an extension of Simple Online Realtime Tracking with an appearance metric (deep SORT), a tracker that runs in realtime and performs well on standard tracking benchmarks. We extend this technique to use a custom diver appearance metric, allow tracks to persist even after arbitrarily long absences, and to recover from certain reidentification errors.

Specifically, this paper contributes the following:

- A multi-diver tracker that can run on a realtime robotic platform.
- An adaptation of the traditional multi-person tracking strategies, which focus on transient pedestrians in crowded scenes, to a strategy that focuses on tracking people who are always present in the scene (although they may not always be present in the robot’s field of

view).

- Evaluation of the proposed tracker on a diverse set of scenarios, including several that depict authentic underwater human-robot collaboration.

2 Related Work

Distinguishing between different divers is typically a difficult task due to low in-class feature diversity: images of different divers are often highly visually similar, both because of similar wearables (*e.g.*, SCUBA gear) and poor visibility conditions underwater. The authors’ previous work includes a first-of-its-kind method to identify divers via *k*-means clustering on hand-crafted feature vectors [8]. The current research extends this work to an online method that can track and uniquely identify divers, utilizing a MOT approach.

MOT has been extensively studied in computer vision. Most research in the area has focused on tracking pedestrians [1–3, 9], and the annual MOT challenges primarily consist of pedestrian datasets [1, 10]. We refer the interested reader to [2] for a thorough review of the field.

Many high-performing trackers process images in batches, rather than online, which makes them infeasible for realtime use [11–13]. Additionally, many of these trackers use computationally-intensive techniques such as optical flow analysis [14, 15] and Multiple Hypothesis Tracking (MHT) [12] which increase tracker accuracy at the cost of processing speed. This phenomenon is illustrated by the leaderboard for the Conference on Computer Vision and Pattern Recognition (CVPR) 2019 MOT Challenge [10]. Only one of the three most accurate submissions has a tracking component that runs at more than two frames per second (the processing time for the detection component is not reported).

In contrast, the fastest trackers rely on relatively simple yet robust heuristics, while still achieving reasonable accuracy. Many of these realtime trackers do not take people’s appearances into account, and instead match detections to tracks solely by analyzing the locations of the detections. For example, the Intersection Over Union (IOU) Tracker [16] matches a detection to a given track if there is a sufficiently high IOU between the detection bounding box and the track’s bounding box in the previous frame. Simple Online Realtime Tracking (SORT) [17] is a slightly more complex model that uses a Kalman filter [18] to model people’s motions and predict their next location. SORT matches detections to tracks if there is a sufficiently high IOU between a detection’s bounding box and the track’s predicted bounding box. While these techniques are reasonable for tracking pedestrians, they are not able to reidentify a person after he or she is temporarily occluded. This shortcoming

was addressed by deep SORT [19], which extends SORT to also use a deeply-learned appearance metric [20] to improve reidentification after occlusions. This work extends deep SORT further to improve reidentification after longer occlusions or absences, as well as customizing the appearance metric for diver reidentification.

Reidentification in visual tracking is closely tied to the more general person reidentification (*i.e.*, reID) problem. The basic reID problem can be formulated as a task to compare persons of interest appearing in ‘query’ datasets to a ‘gallery’ of potential candidate images, captured from different angles, different cameras and even different scenes. Existing work focuses almost exclusively on person retrieval on land, which are either image-based or video-based. Person reID methods often use visual cues based on the individual’s height, face, complexion, and gait [21, 22]. However, these methods are not reliable in situations where face or gait recognition is not feasible (*e.g.*, for poor image resolution, or in images captured from different angles). Gheissari *et al.* propose a novel method [23] which relies on features invariant to illumination, pose, and dynamic appearance of clothing [23]. Recent contributions increasingly rely on deep machine learning ([24–27]) for their improved accuracy in the reID task, although most are not realtime capable.

Detection is another crucial component of tracking-by-detection systems. Diver detection is a difficult problem, largely because underwater visual perception presents various challenges, including color distortion, suspended particles, and light refraction, absorption, and scattering [28, 29]. Additionally, divers have a wider range of potential positions and orientations than people on land, since divers are suspended in a 6-DOF aquatic environment. Islam *et al.* [30] design a CNN-based realtime-capable diver detection model while sacrificing relatively little accuracy. We use this model for the detection component of our tracker.

3 Methodology

Our algorithm keeps a set of all known tracks, denoted as \mathcal{T} . After finding detections with the network described in [30] for a given frame, the algorithm attempts to match each detection to an existing track $t \in \mathcal{T}$. Below we describe how the algorithm makes these matches.

3.1 Intersection Over Union

Our algorithm first checks to see if any detections can be matched to tracks via intersection over union (IOU). We begin with this strategy because simple IOU trackers can be very effective [16], and IOU can be computed quickly. Therefore, we take the IOU “shortcut” if possible before doing any more intensive computations.

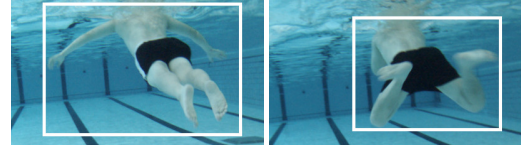


Figure 3: An illustration of how the area and aspect ratios of bounding boxes can oscillate rapidly during swimming.

We use the following definition for the IOU between bounding boxes A and B :

$$IOU(A, B) = \frac{A \cap B}{A \cup B}$$

Our IOU-assignment procedure is as follows: We have a set of detections \mathcal{D}_t that contains all detections found at time t . We then calculate the IOU between each $d_i \in \mathcal{D}_t$ and each $d_j \in \mathcal{D}_{t-1}$. If $IOU(d_i, d_j) \geq 0.75$, d_i is assigned to the same track as d_j . This is a more conservative threshold than the 0.5 threshold used by the IOU tracker [16], since the IOU tracker uses batch processing to eliminate erroneous associations and we process each frame sequentially.

3.2 Appearance and Location Metrics

To match the remaining detections to tracks, we utilize two metrics: an appearance metric that describes the similarity of a detection’s appearance and a track’s appearance, and a location metric that describes the similarity between a detection’s location and a track’s location.

The **location metric** is largely identical to the approaches in [20] and [17]. We use a simple Kalman filter to model the motion of each track. Our tracking scenario is defined on the four-dimensional state space (x, y, \dot{x}, \dot{y}) where (x, y) is the bounding box’s center position in image coordinates and \dot{x} and \dot{y} are the respective velocities of x and y . We do not include the area or aspect ratio of the bounding box in the state space because due to rapid arm and leg motion from swimming strokes, the aspect ratio and area of a diver’s bounding box can oscillate rapidly (see Figure 3 for an illustration of this effect). The location metric between detection d_i and track t_j is then defined as the squared Mahalanobis distance between the Kalman filter’s predicted location of t_j and the actual location of d_i .

The **appearance metric** differs from the approach used by other trackers. We calculate a series of hand-crafted features to find a feature vector that describes the appearance of the detected diver. In contrast, most trackers use a deep neural network trained on a person reidentification dataset to generate feature vectors, *e.g.*, [20, 31, 32]. We do not take this approach for two reasons: (a) networks trained on person reidentification datasets are not well-suited to reidentifying divers [33], and (b) diver-specific reidentification

datasets do not exist and data scarcity prevents us from creating one.

The appearance features we extract and the reasoning behind their inclusion are fully described in [8]. In summary, we use features that can satisfactorily differentiate between divers, but are also relatively robust to changes in lighting, and diver position and orientation:

- Average color distribution in the LAB color space
- Amplitude of the spatial frequency distribution
- Shape approximation through image contours
- Shape approximation through convex hull
- Hu image moment invariants [34]

These features have been shown to be sufficient for the k-means algorithm to effectively cluster images of divers according to their identities [8].

We take the same approach as [19] for calculating appearance similarities between tracks and detections. For each track, we store the normalized feature vectors of the 100 most recent detections that have been matched to that track. To measure the appearance similarity between a detection and a track, we find the cosine similarities between each of the track’s stored feature vectors and that detection’s normalized feature vector. The smallest of these cosine similarities is then the appearance similarity between the track and detection. Concretely, if detection d_i ’s normalized feature vector is \mathbf{f}_i , and \mathcal{F}_j is the set of stored normalized feature vectors for track t_j , then the appearance similarity between d_i and t_j is calculated with:

$$\text{sim}_{\text{appearance}}(d_i, t_j) = \min(1 - \mathbf{f}_i^T \mathbf{f}_j \mid \mathbf{f}_j \in \mathcal{F}_j)$$

3.3 Matching Detections to Tracks

Next, we must use the location and appearance metrics to match the remaining detections to tracks. We use the typical strategy of formulating an assignment problem that can be solved with the Hungarian algorithm [35]. This is done by finding a cost, c_{ij} for matching detection d_i to track t_j . We let $c_{ij} = \text{sim}_{\text{appearance}}(d_i, t_j)$. Because of poor visibility underwater and strong resemblance between divers’ SCUBA gear, sometimes a detection is highly visually similar to more than one track. In this case, we introduce the location similarity as a tie-breaker and have $c_{ij} = \text{sim}_{\text{appearance}}(d_i, t_j) + \text{sim}_{\text{location}}(d_i, t_j)$ for all costs associated with that detection. The Hungarian algorithm then finds the optimal matches between detections and tracks such that the costs are minimized.

We do not rely heavily on the location similarity to contribute to the cost of a match. This is because the location

similarity is derived through a Kalman filter, which is designed to model linear systems [18]. Since divers’ movements are not consistently linear, we cannot highly depend on Kalman filter predictions. In addition, the robot’s exact motions (and by extension the camera’s exact motions) are unknown, which also negatively affects the Kalman filter’s predictive power.

However, the Kalman filter can be useful for determining which matches between detections and tracks are unacceptable. We consider matches between detections and tracks to be unacceptable if the detection’s location is too far away from the track’s predicted location (*i.e.*, if the location metric is above a certain threshold), or if the detection’s appearance is too dissimilar from the track’s appearance (*i.e.*, if the appearance metric is above a certain threshold). The thresholds used for unacceptable matches were found empirically by testing our algorithm on a validation dataset. We used 25 for the location metric threshold and $1e-4$ for the appearance metric threshold.

We indicate an unacceptable match by setting $c_{ij} = \infty$. If a detection cannot be matched to a track with a cost $c < \infty$, we create a new track for that detection. To account for spurious detections, new tracks are not officially included in \mathcal{T} until they have been matched with a detection for three consecutive frames.

3.4 Short Term vs. Long Term Reidentification

When a detection is not matched to a track for a frame, the track is no longer active. If a person belonging to an inactive track is detected, the person will need to be reidentified, *i.e.*, matched to their existing track.

Our algorithm as described above can accomplish reidentification if the detection’s appearance is similar enough to the inactive track and its location is similar enough to the inactive track’s predicted location. However, our algorithm changes its approach for tracks that have been inactive for a longer period of time (*i.e.*, more than five frames). In this case, we no longer calculate a predicted track location, because the track’s Kalman filter will have too much uncertainty and propagated error. We also increase the appearance similarity threshold slightly, to $5e-4$. This is because when a person is absent from the scene for a longer period of time, there may be significant changes to their position and orientation, as well as the scene’s lighting, so we adopt a more forgiving threshold. The threshold was also obtained empirically by testing our algorithm on a validation dataset.

3.5 Identity Recovery

One potential problem with our reidentification technique is that we must correctly reidentify a diver on the first frame in which he or she reenters the robot’s field of view.

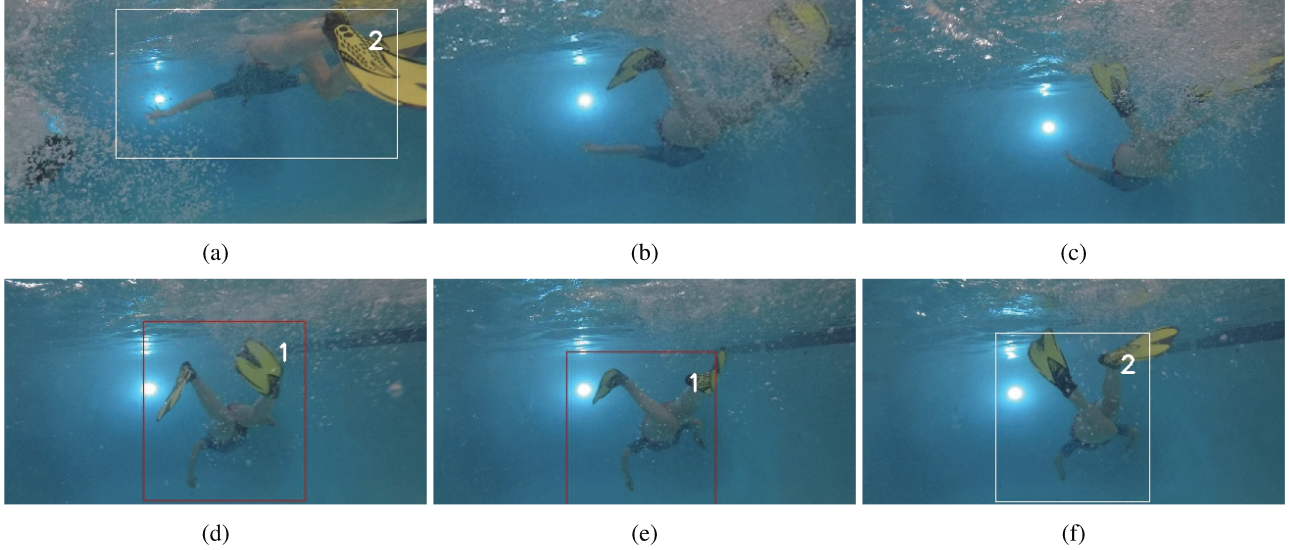


Figure 4: An example of our identity recovery technique. (4a) This swimmer is being tracked with identity “2.” (4b-4c) When the swimmer descends, bubbles partially occlude the body and the detector fails. (4d) Once the bubbles begin to clear, the change in body position means that the tracker does not find enough appearance similarity to correctly match this detection to track “2.” Instead a new track “1” is created. (4e) Since “1” is a new track, the algorithm continually checks for sufficient similarities between “1” and tracks that have not been seen since “1” appeared. (The red colored bounding box indicated that the tracker is uncertain about this track’s identity.) (4f) The algorithm finds a sufficient amount of similarity between new track “1” and old track “2.” Track “1” is merged into “2”, and the swimmer’s appropriate identity is recovered.

If, in that initial frame, the diver’s position or a temporary partial occlusion leads to a high dissimilarity between the detection’s appearance metric and the diver’s true track, the diver will not be assigned to their true track. We refer to this scenario as a missed reidentification.

Missed reidentifications are not generally a large concern in the MOT community, and the standard MOT evaluation metrics do not heavily penalize missed reidentifications. However, for our use case, missed reidentifications are hugely problematic since they lead to the robot being mistaken about a diver’s identity. To address this problem, we have a procedure to correct missed reidentifications.

The process is as follows: after the algorithm matches detections to tracks, it examines the set of new tracks. We consider a track to be “new” if it has existed for 15 or fewer frames. For each new track t_i , the algorithm finds the set of tracks that have not been seen since t_i was created. We consider this set to contain all tracks that could possibly belong to the same person represented by t_i .

Next, the algorithm checks to see if t_i does in fact represent the same person as another track t_j . To do this, we find the cosine similarity between each feature vector stored for t_i and each feature vector stored for t_j . If more than 25% of these cosine similarities are below our acceptable appearance similarity matching threshold, the algorithm decides that t_i and t_j share an identity. (Again, this threshold was established empirically through testing on a validation dataset.) Then track t_i is merged into t_j . See Figure 4 for

an illustration of this technique in practice.

4 Experiments

We evaluate the performance of the proposed tracker by testing it on eight videos, six of which take place in open-water (*i.e.*, ocean) environments, and two of which take place in closed-water (*i.e.*, pool) environments (see Table 2). The six ocean videos were all recorded during field trials and thus represent authentic underwater human-robot collaboration scenarios. The two pool videos were recorded to simulate human-robot collaboration scenarios. Each video contains between two and four divers, and the videos are about 10 minutes long in total. The ground truth identities and bounding boxes for each video were annotated by hand.

We compare our tracker’s performance to SORT and deep SORT to ensure that our modifications result in better tracking for underwater human-robot collaboration scenarios. The SORT method does not incorporate any appearance information, whereas deep SORT uses a CNN trained on the MARS person reidentification dataset [36]. Table 2 contains a summary of the three trackers’ performances on our eight videos. All trackers used our custom diver detector to generate detections (note that the detector’s performance is also included Table 2). We then used standard MOT metrics [37] to evaluate how well the different trackers identify divers; see Table 1 for a brief description of the metrics.

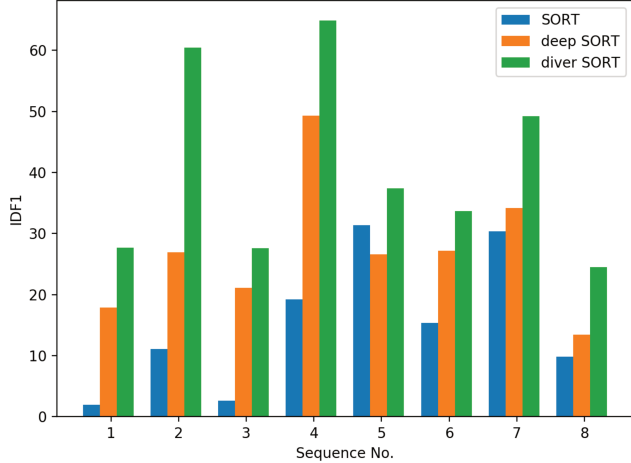


Figure 5: A comparison of the IDF1 metric across all eight scenarios for the three trackers tested.

Across all eight videos, our tracker performed best in correctly identifying detected divers as measured by IDF1 (Figure 5), IDP, and IDR. This indicates that our reidentification and identification recovery methods are effective. Since our tracker specifically outperforms deep SORT, the results also indicate that deeply learned person reidentification appearance metrics fall short of the hand-crafted features used by our tracker. However, deep SORT does consistently outperform SORT on identification, so the deeply learned appearance metric is not ineffective.

Our tracker had no improvement over the others on identity switches and fragmentations: at best, our tracker had marginally fewer identity switches (*e.g.*, sequences 2 and 5 in Table 2); otherwise, it was not a top performer. Our tracker also consistently had the most fragmentations, likely because our tracker persists track identities even through long occlusions or periods of absence. In some cases, such as Scenario 1, the baseline SORT tracker had extremely low IDS and FM scores. This was due to SORT only matching a few detections to tracks, and each track lasting for only 1-2 frames. Such tracks are too short to experience identity switches and fragmentations.

It is also important to note that there was a wide range for IDF1, IDP, and IDR values across the eight videos. Additionally, there is a strong relationship between those values and the detector’s performance. For example, video 7 resulted in the best detector performance, as well as our tracker’s best IDF1 score (64.9). On the other hand, video 6 resulted in our detector’s worst performance and also our tracker’s worst IDF1 score (24.5). We found a positive correlation between detection precision and identity precision for both our tracker (Pearson $r \approx 0.727$, $p < 0.05$) and deep SORT (Pearson $r \approx 0.850$, $p < 0.05$). This is unsurprising since detection quality is known to have a high impact on a tracker’s performance [2, 17, 38].

Metric	Description
DP	Detection Precision. Defined as $TP/(TP + FP)$ where TP is a true positive (<i>i.e.</i> , a detection that closely matches a ground truth bounding box) and FP is a false positive (<i>i.e.</i> , a detection that does not closely match a ground truth bounding box). All trackers used the same detector.
DR	Detection Recall. Defined as $TP/(TP + FN)$ where TP is a true positive detection and FN is a false negative detection (<i>i.e.</i> , a ground truth bounding box that does not closely match any detection). All trackers used the same detector
IDF1	Identity F1. Harmonic mean of IDR and IDP.
IDP	Identity Precision. Defined as $(IDTP)/(IDTP + IDFP)$ where IDTP is the number of true positive identities (<i>i.e.</i> , identities output by the tracker that match ground truth identities) and IDFP is the number of false positive identities (<i>i.e.</i> , identities output by the tracker that do not match ground truth identities) in the tracker’s output.
IDR	Identity Recall. Defined as $(IDTP)/(IDTP + IDFN)$ where IDTP is the number of true positive identities, IDFP is the number of false positive identities, and IDFN is the number of false negative identities.
IDS	Identity Switches. The total of number of times that a tracked trajectory changes its matched ground truth identity.
FM	Fragmentations. The total number of times a tracked trajectory is interrupted (<i>i.e.</i> , frames are dropped).

Table 1: Description of MOT metrics used in evaluation. See [37, 38] for more detailed discussions.

Our tracker’s end-to-end image processing rate is 9.8 frames per second on a machine with an AMD Ryzen 5 2600 3.9GHz processor and 16GB of RAM, making it real-time capable for AUV deployment.

5 Conclusion

In this paper, we propose a method to allow underwater robots to track and identify people in real time. Our method combines techniques from multi-object tracking and underwater diver detection and identification. Results show that our custom tracker has more correct identifications than baseline realtime trackers on underwater datasets. How-




Scen.	Location	Exemplar	DP \uparrow	DR \uparrow	Tracker	IDF1 \uparrow	IDP \uparrow	IDR \uparrow	IDS \downarrow	FM \downarrow
1	ocean		49.7	38.7	SORT	2.0	28.6	1.0	0	0
					deep SORT	17.9	20.5	15.8	22	25
					diver SORT	27.7	39.3	21.4	7	19
2	ocean		59	35.9	SORT	11.1	57.1	6.2	1	1
					deep SORT	26.9	35.9	21.5	3	7
					diver SORT	60.4	78.0	49.2	0	8
3	ocean		64.5	49.1	SORT	2.6	42.9	1.3	0	0
					deep SORT	21.1	24.4	18.5	24	29
					diver SORT	27.6	41.6	20.7	15	27
4	ocean		62.7	37.2	SORT	15.4	25.3	11.1	11	25
					deep SORT	27.2	36.6	21.7	16	26
					diver SORT	33.6	46	26.5	19	46
5	ocean		70.7	54.7	SORT	30.4	42.2	23.8	23	63
					deep SORT	34.1	39.2	30.3	30	80
					diver SORT	49.2	58.6	42.4	19	92
6	ocean		30.5	25	SORT	9.8	12.9	7.9	15	37
					deep SORT	13.4	15.4	11.9	9	37
					diver SORT	24.5	28.4	21.5	10	53
7	pool		81.6	60.3	SORT	19.2	27.7	14.7	25	43
					deep SORT	49.3	58.1	42.8	14	53
					diver SORT	64.9	80.0	54.6	17	65
8	pool		63.2	25.8	SORT	31.4	68.6	20.3	2	13
					deep SORT	26.6	46.3	18.6	4	15
					diver SORT	37.4	51.9	29.2	12	26
ALL					SORT	18.6	29.6	13.6	77	182
					deep SORT	30.0	36.3	25.5	122	272
					diver SORT	42.1	53.1	34.9	99	336

Table 2: A comparison of our algorithm’s performance (diver SORT) and two other realtime trackers’ performance on several videos of divers. See Table 1 for a brief description of the metrics used.

ever, in situations where the detector produces highly inaccurate detections, all tested trackers perform poorly. This

has two implications: (1) improving our detector can result in highly significant tracking improvements, and (2) the

tracker has the potential to be highly unreliable in the field when adverse conditions reduce detection accuracy. In such situations, the erroneous tracker output could lead to erratic and unpredictable robot behavior, which may jeopardize the mission. Future work therefore involves improving real-time diver detection and developing a system that can flag poor detector performance (*e.g.*, dropping many frames, inconsistent numbers of detections between frames). When poor detection conditions are identified, the humans working with the robot can be made aware that the robot's tracking module should not be relied upon.

References

- [1] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A Benchmark for Multi-Object Tracking," *arXiv:1603.00831 [cs]*, Mar. 2016. arXiv: 1603.00831.
- [2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple Object Tracking: A Literature Review," *arXiv:1409.7618 [cs]*, Sept. 2014. arXiv: 1409.7618.
- [3] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: an analysis of the state of the art in multiple object tracking," *arXiv preprint arXiv:1704.02781*, 2017.
- [4] M. J. Islam, J. Hong, and J. Sattar, "Person Following by Autonomous Robots: A Categorical Overview," *International Journal of Robotics Research*; *accepted for publication*, Sept. 2019. preprint available: arXiv: 1803.08202.
- [5] O. Hoegh-Guldberg, P. J. Mumby, A. J. Hooten, R. S. Steneck, P. Greenfield, E. Gomez, *et al.*, "Coral Reefs under Rapid Climate Change and Ocean Acidification," *Science*, vol. 318, no. 5857, pp. 1737–1742, 2007.
- [6] F. Shkurti, A. Xu, M. Meghjani, J. C. G. Higuera, Y. Girdhar, P. Giguere, B. B. Dey, *et al.*, "Multi-domain Monitoring of Marine Environments using a Heterogeneous Robot Team," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1747–1753, IEEE, 2012.
- [7] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, *et al.*, "Robotic Tools for Deep Water Archaeology: Surveying an Ancient Shipwreck with an Autonomous Underwater Vehicle," *Journal of Field Robotics (JFR)*, vol. 27, no. 6, pp. 702–717, 2010.
- [8] Y. Xia and J. Sattar, "Visual diver recognition for underwater human-robot collaboration," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6839–6845, IEEE, 2019.
- [9] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the CLEAR MOT metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.
- [10] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and

- L. Leal-Taixe, "CVPR19 tracking and detection challenge: How crowded can it get?," *arXiv preprint arXiv:1906.04567*, 2019.
- [11] L. Fagot-Bouquet, R. Audigier, Y. Dhome, and F. Lerasle, "Improving multi-frame data association with sparse representations for robust near-online multi-object tracking," in *European Conference on Computer Vision*, pp. 774–790, Springer, 2016.
- [12] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4696–4704, 2015.
- [13] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple People Tracking by Lifted Multicut and Person Re-identification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Honolulu, HI), pp. 3701–3710, IEEE, July 2017.
- [14] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, pp. 4705–4713, 2015.
- [15] S. Shantaiya, K. Verma, and K. Mehta, "Multiple object tracking using Kalman filter and optical flow," *European Journal of Advances in Engineering and Technology*, vol. 2, no. 2, pp. 34–39, 2015.
- [16] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.
- [17] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, IEEE, 2016.
- [18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [19] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, IEEE, 2017.
- [20] N. Wojke and A. Bewley, "Deep Cosine Metric Learning for Person Re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 748–756, Mar. 2018.
- [21] W. Zajdel, Z. Zivkovic, and B. Krose, "Keeping track of humans: Have I seen this person before?," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 2081–2086, IEEE, 2005.
- [22] T. Huang and S. Russell, "Object identification in a Bayesian context," in *IJCAI*, vol. 97, pp. 1276–1282, 1997.
- [23] N. Gheissari, T. B. Sebastian, and R. Hartley, "Person reidentification using spatiotemporal appearance," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1528–1535, IEEE, 2006.
- [24] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: Deep Filter Pairing Neural Network for Person Re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 152–159, 2014.
- [25] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 3376–3385, IEEE, 2017.
- [26] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by GAN improve the person re-identification baseline in vitro," *arXiv preprint arXiv:1701.07717*, vol. 3, 2017.
- [27] N. McLaughlin, J. Martinez del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1325–1334, 2016.
- [28] J. Sattar and G. Dudek, "On the performance of color tracking algorithms for underwater robots under varying lighting and visibility," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 3550–3555, IEEE, 2006.
- [29] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing underwater imagery using generative adversarial networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7159–7165, IEEE, 2018.
- [30] M. J. Islam, M. Fulton, and J. Sattar, "Toward a generic diver-following algorithm: Balancing robustness and efficiency in deep visual detection," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 113–120, 2018.

- [31] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *European conference on computer vision*, pp. 475–491, Springer, 2016.
- [32] C. Long, A. Haizhou, Z. Zijie, and S. Chong, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *ICME*, vol. 5, p. 8, 2018.
- [33] Y. Xia, S. Xu, and J. Sattar, "On the Land and Underwater: Person Re-Identification for Human-Robot Collaboration." Unpublished, In preparation: draft available at <http://irvlab.cs.umn.edu/files/personreid>, August 2019.
- [34] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.
- [35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [36] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *European Conference on Computer Vision*, pp. 868–884, Springer, 2016.
- [37] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking," in *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.
- [38] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.