Fine-grained Categorization of Fish Motion Patterns in Underwater Videos

Mohamed Amer[1], Emil Bilgazyev[2],
Sinisa Todorovic[1], Shishir Shah[2], Ioannis Kakadiaris[2], Lorenzo Ciannelli[3]
[1]School of EECS, Oregon State University, OR
[2]Department of Computer Science, University of Houston, TX
[3]College of Oceanic and Atmospheric Sciences, Oregon State University, OR
{amerm,sinisa}@onid.orst.edu, emilbek@cs.uh.edu
{sshah,ikakadia}@central.uh.edu, lciannel@coas.oregonstate.edu

Abstract

Marine biologists commonly use underwater videos for their research on studying the behaviors of sea organisms. Their video analysis, however, is typically based on visual inspection. This incurs prohibitively large user costs, and severely limits the scope of biological studies. There is a need for developing vision algorithms that can address specific needs of marine biologists, such as fine-grained categorization of fish motion patterns. This is a difficult problem, because of very small inter-class and large intra-class differences between fish motion patterns. Our approach consists of three steps. First, we apply our new fish detector to identify and localize fish occurrences in each frame, under partial occlusion, and amidst dynamic texture patterns formed by whirls of sand on the sea bed. Then, we conduct tracking-by-detection. Given the similarity between fish detections, defined in terms of fish appearance and motion properties, we formulate fish tracking as transitively linking similar detections between every two consecutive frames, so as to maintain their unique track IDs. Finally, we extract histograms of fish displacements along the estimated tracks. The histograms are classified by the Random Forest technique to recognize distinct classes of fish motion patterns. Evaluation on challenging underwater videos demonstrates that our approach outperforms the state of the art.

1. Introduction

This paper presents an approach to video categorization aimed at facilitating a particular marine biology study. With the proliferation of inexpensive and easy-to-use digital technologies for video acquisition, there is a tremendous growth of underwater video footage. This has enabled significant progress in fisheries ecology. For example, the Fishery Resource Analysis and Monitoring Division (FRAM) of the Northwest Fisheries Science Center manages West-coast Groundfish stocks and their ecosystems only based on large video collections. Also, an analysis of videos showing demersal fish off the North West Coast [1] have resulted in important biological findings on the effects of low levels of dissolved oxygen (DO) on early life stages of demersal fish [8, 11, 21]. The findings state that varying concentrations of DO in the water – ranging from lethal via sub-lethal to normal – can be quantified by observing fish swimming speed, direction, periodicity, and escape response time. These observations are important, since alternative methods of measuring DO levels have many disadvantages. One such disadvantage is that they provide only local measurements, and thus have to be deployed in large numbers over a large spatial area to acquire statistically significant data. A inexpensive and potentially more accurate estimates of DO levels can be obtained by observing fish behavior in underwater videos. In the above examples, and in general, video analysis in marine biology is typically not automated, but requires laborious visual inspection. Thus, there is a critical need for developing computer vision approaches to help marine biologists in their analysis of underwater videos.

1.1. The Problem

Motivated by the aforementioned needs of marine biology, we have developed an approach to classifying videos by fish motion patterns, also referred to as behavior classes, which occur in the videos. The classes of fish behavior are defined in terms of fish swimming speed, direction, periodicity, and escape response time. The underwater videos that we consider in this paper are publicly available [1]. The videos are captured by a camera, mounted on a moving trawl, which drags a chain along the sea bed, and thus makes move flat fish lying hidden under the sand (Fig. 1). The videos are acquired twice a day in the morning and afternoon, at sea depths 30 *m*, 40 *m*, and 50 *m*. The video acquisition is aimed at correlating fish behavior with a spe-

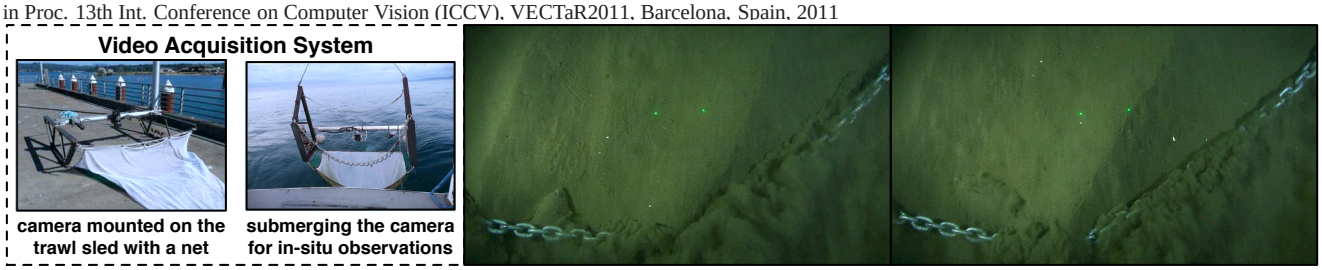

Figure 1. Our video acquisition system and challenges of underwater videos: (left) The videos are captured by a moving camera, mounted on the submerged trawl sled. A chain is dragged over the sea bed to stir flatfish lying in the sand surface. This causes whirls of sand. (middle and right) Fish are well camouflaged to have the same color and texture as surrounding sand, and may be partially occluded by the chain or partially occlude one another when swimming in a school of fish. The annotated dataset will be made publicly available at [1].

cific sea depth and time of the day, for which the biologists already know the corresponding DO levels. The three sea depths and half-a-day frequency of collecting the videos jointly define six event classes, corresponding to six behavioral patterns of fish. Given a new (previously unseen) video, the goal is to identify the sea depth and time of capturing that video (i.e., its class). This problem is difficult, because the above six event classes are characterized by very small inter-class differences and very large intra-class variations in fish motions.

1.2. Our Approach

This paper presents an efficient and robust approach to fine-grained event classification, illustrated in Fig. 2. It consists of the following three steps. First, we apply our new fish detector to identify and localize fish occurrences in each frame, under partial occlusion, and amidst dynamic texture patterns formed by whirls of sand on the sea bed. Then, we conduct tracking-by-detection. Given a similarity function between fish detections, defined in terms of fish appearance and motion properties, we formulate fish tracking as transitively linking similar detections between every two consecutive frames, so as to maintain their unique track IDs. This data association problem is specified as maximum weight clique problem, and solved using the replicator dynamics algorithm. Finally, we extract histograms of fish displacements along the estimated tracks, and then classify the histograms by the Random Forest [4] to recognize distinct classes of fish motion patterns.

1.3. Challenges and Prior Work

Fine-grained classification of underwater videos presents many challenges that are poorly addressed by existing work, primarily stemming from two variables. **Variability in Appearance:** Fish exhibit large appearance variations in terms of color, shape, size, and texture. These variations arise from the following factors: natural growth and aging of individual fish, long-term environmental factors, and temporarily present fish interactions with the environment. For example, as fish move through the water, their shape may be highly deformable. In addition, fish appearance may be affected by different imaging conditions, such as partial occlusion and self-occlusion, low resolution, reflections of light, varying camera viewpoints of fish (e.g., frontal or side view). All these factors makes the problem of automatically detecting and localizing fish in underwater videos very challenging. Current work mainly focuses on recognition of rigid, articulated object classes (e.g., cars, or people in certain postures [10, 12, 16]) which are prominently featured in the foreground. *Open question: How to specify detectors that can address a large variability of color, shape, size, and texture of fish, viewed under different imaging conditions?*

Variability in Motion: Motions and actions of individual fish (e.g., turning, foraging, evading obstacles), and interactions with the other fish (e.g., swimming in a school of fish, escaping from predators) are highly variable. Existing work mainly focuses on tracking rigid objects which move smoothly, and thus is not appropriate for analyzing highly erratic fish motions and their complex interactions [5, 9, 13, 17, 22, 23, 25]. An alternative is to be use an object detector to generate target hypotheses in each frame, and then transitively link the detections so as to maintain their unique identities. Transitive linking is very difficult in underwater videos, due to dynamic textures in the background, and severe partial occlusions. Similar challenges are usually addressed by learning an affinity model between detections in terms of color and speed [14, 17–19], spatiotemporal context [19], and occluder map [13]. Given affinities between detections, the aforementioned work formulates tracking as a data association problem. This is typically posed as bipartite matching, and solved by either the greedy Hungarian algorithm, or the more sophisticated network flow algorithms [26]. Tracking-by-detection approaches may perform poorly in the presence of long gaps in a sequence of object detections (e.g., due to occlusions). This challenge can be addressed by a hierarchical association of detections [13]. *Open question: How to conduct tracking of highly deformable non-rigid objects, like fish, with erratic, non-smooth motion trajectories, under sever occlusion, and*

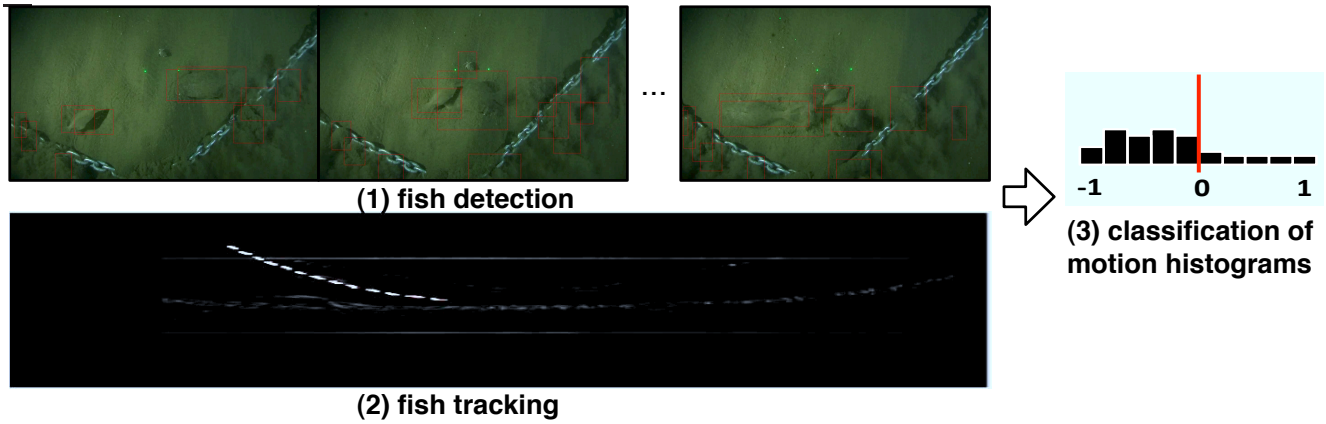

Figure 2. Main steps of our approach: (1) Detection and localization of all fish occurrences in every video frame (red bounding boxes); (2) Tracking-by-detection to transitively link similar fish detections across the frames; (3) Extraction of motion features along the track estimates, and their classification using a Random Forest classifier [4]. The videos in our dataset are captured with a moving camera, and have low contrast, low resolution, motion blur, and dynamic texture in the background from whirls of sand on the sea bed. Also, fish may partially occlude one another, and they have erratic motion patterns.

Figure 3. Flowchart of the proposed fish detector.

amidst dynamic texture distractors?

To the best of our knowledge, there is no vision system aimed at categorizing fine-grained motion classes of fish in underwater videos. In the following three sections, we describe the three steps of our approach.

2. The Proposed Fish Detector

Our work is based on the use of optical flow for accurate segmentation and background/foreground separation. The complexity of the motion model assumed in detecting the dominant change in a frame and the degree to which a foreground image should be segmented are independent. We consider a motion model that is based on the notion of dominant change, so as to independently treat changes that adhere to the assumed model and the ones that are non-adherent. In particular, we use a simple translation motion model as the dominant change between consecutive frames. Outliers to this model are processed further to differentiate between fish and noise.

To detect moving fish, optical flow is computed between every two consecutive frames. Optical flow vectors are then clustered to identify different motion groups. A background/foreground analysis is used to remove background segments, and shape analysis is used to remove false positives (Fig. 3). For optical flow estimation we use the algorithm proposed by Brox *et al.* [7], and for clustering we use agglomerative hierarchical cluster tree with inner squared distance metric (minimum variance) to compute the distance between the clusters [24]. In the following, we explain these steps in more detail.

Let us denote the i^{th} frame of the video as I_i, and the displacement vectors as $(\vec{u}, \vec{v})$ (Figs. 4(c), 4(d)). A feature vector is built as $\vec{V} = [\vec{u}\ \vec{v}\ \vec{\alpha}\ \vec{\beta}]$ for each pixel in frame I_i, where $\vec{\alpha} = arctan(\vec{u}, \vec{v})$ is the angle of the displacement (Fig. 4(e)), and $\vec{\beta} = \sqrt{\vec{u}^2 + \vec{v}^2}$ is the amplitude of the displacement (Fig. 4(f)). Clustering is used to build a regional grouping of similar features (Fig. 4(g)). Assuming a constant background motion, clustering results in one cluster for the background and n clusters for the foreground, where n is the number of moving objects in the scene.

Due to deviations from our assumption, and the complexities stemming from the camera motion and the imaged scene, optical flow computation does not necessarily return a constant displacement vector for the background (Fig. 4(g)). In this case, to remove false positive results, foreground/background analysis is performed. Clusters with image regions which are bigger than the mean fish size are assumed to belong to the background, and those smaller sizes than the mean fish size are assumed to be noise and ignored from further processing (Fig. 4(h)). In addition, we compute the mean of feature vector for all remaining clusters and remove those clusters which are similar to the background feature vector mean (Fig. 4(i)). Finally, we compute the gradient changes within the remaining clusters. Assuming that the fish undergo displacement in each frame, the motion would introduce significant gradient changes. Thus, all clusters with small gradient distribution are also removed from further processing (orange color in Fig. 4(i)). Then, for each of the remaining clusters, x_i, y_i, w_i and h_i are computed (bounding box), where i is the cluster ID (Fig. 4(j)). Algorithm 1 summarizes our fish detector.

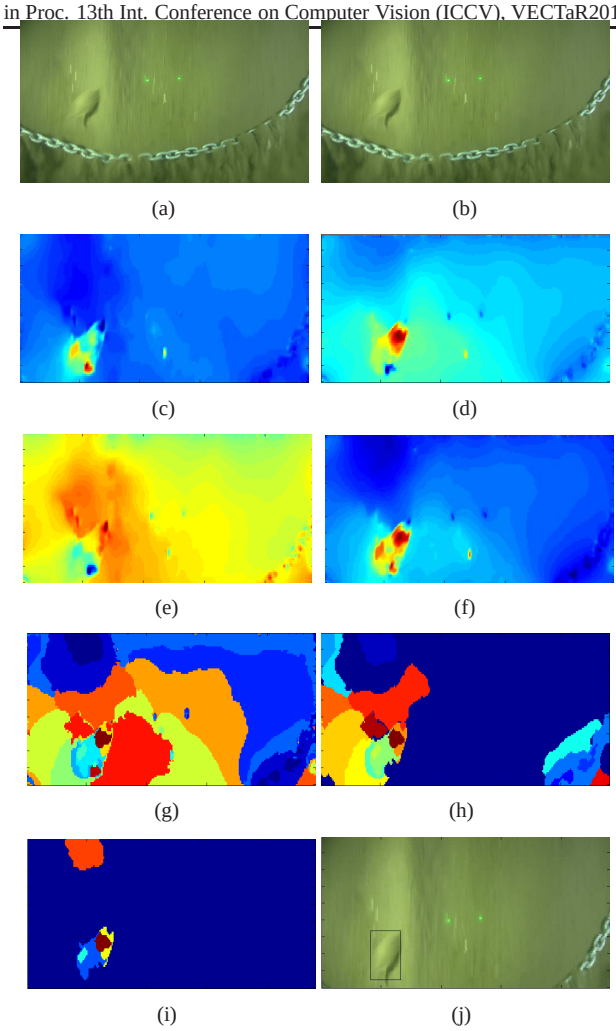

Figure 4. Output of our fish detector. (a,b) Input frames, (c,d) $\vec{u}$, $\vec{v}$, (e,f) $\vec{\alpha}$, $\vec{\beta}$, (g) Output of the clustering algorithm, (h) Output after background estimation based on the mean fish size, (i) Output after removing clusters that are similar to the background, (j) Output bounding box. Cluster in orange is a false positive which is removed, since there are no gradient changes inside the region.

3. Fish Tracking

This section presents our approach to tracking all fish occurrences in the video. The total number of fish, their motions, and spatiotemporal configurations are unknown. We use the fish detector presented in Sec. 2 to scan each video frame, and detect likely locations of fish occurrences. As an input parameter, the detector has a threshold which controls the decision making as to which locations in a frame to declare as fish occurrences against the background. We purposefully select this threshold to be low, so as to generate many hypotheses of fish occurrences, and thus does not miss the true ones. More formally, the threshold is set such that the detector has high recall, and, consequently, low precision. The obtained false positives are filtered out by transitively linking similar detections across the frames, as illustrated in Fig. 5. In the sequel, we explain how to transitively link similar fish detections.

Algorithm 1: Fish detection

Input: frame I_i, I_{i+1}
Output: x, y, w, h

1: Compute optical flow $[\vec{u}\ \ \vec{v}] \leftarrow (I_i, I_{i+1})$
2: $\vec{V} = [\vec{u}\ \ \vec{v}\ \ arctan(\vec{u}, \vec{v})\ \ \sqrt{\vec{u}^2 + \vec{v}^2}]$
3: $c =$ cluster$(\vec{V})$
4: Background $c_b \leftarrow size(c) > size(fish)$
5: Mean feature vector of background $\vec{V}_b^*$=mean$(\vec{V}(c_b))$
6: $\vec{V}_i^*$=mean$(V(c_i))$, where c_i is the i^{th} cluster
7: Remove $\left\|\vec{V}_i^* - \vec{V}_b^*\right\| < \xi$
8: Compute x_i, y_i, w_i, and h_i for each cluster

Each detected bounding box, z, is characterized by a descriptor, $\boldsymbol{z}$, whose elements include: (a) location and size of the bounding box, and (b) a PCA projected vector at 5% reconstruction error of the following features: (b.i) HOG descriptor of size 81×1, (b.ii) HSV color histogram of size 256×3, and (b.iii) two 10-bin histograms of optical flow along x and y directions within the box.

Given two detections z and z', and their descriptors $\boldsymbol{z}$ and $\boldsymbol{z}'$, similarity between them is defined as:

$$w = \exp(-(\boldsymbol{z} - \boldsymbol{z}')^{\mathrm{T}}\mathbf{M}(\boldsymbol{z} - \boldsymbol{z}')), \qquad (1)$$

where $\mathbf{M}$ is a distance metric. $\mathbf{M}$ is a diagonal matrix, whose diagonal elements are inversely proportional to the variance of each feature in the bounding box descriptor $\boldsymbol{z}$. These variances are estimated on training examples.

We formulate tracking as finding the maximum weight clique of a graph whose nodes are pairs of detections and edges encode their similarity, given by (1).

More formally, let $Z^{(t)} = \{z_1^{(t)}, z_2^{(t)}, \dots\}$ denote the set of object detections at time t, and $Z = \cup_{t=1,\dots,T} Z^{(t)}$ be the set of all detections. A track is an ordered set of detections $\mathcal{T} = \{z_a^{(t_1)}, z_b^{(t_2)}, \dots\}$, such that $\forall t, |\mathcal{T} \cap Z^{(t)}| \leq 1$. It follows that tracking can be defined as the problem of finding a subset of all detections whose time sequences form a set of non-overlapping tracks, $\Sigma = \{\mathcal{T}_k : \mathcal{T}_k \cap \mathcal{T}_l = \emptyset,\ k \neq l,\ k, l = 1, 2, \dots\}$, $\Sigma \subseteq Z$, such that each $\mathcal{T}_k \in \Sigma$ is a set of all detections of a unique target.

Tracking can be formalized by constructing a graph, $G = (V, E, w)$. V is the set of nodes representing pairs of object detections from every two consecutive frames, called tracklets, $V = \{i^{(t)} : i^{(t)}{=}(z_a^{(t)}, z_b^{(t+1)}), z_a^{(t)}{\in}Z^{(t)}, z_b^{(t+1)}{\in}Z^{(t+1)}, t{=}1, ..., T\}$, with cardinality $|V|{=}n$. E is the set of undirected edges connecting only those tracklets $i^{(t)}{\in}V$

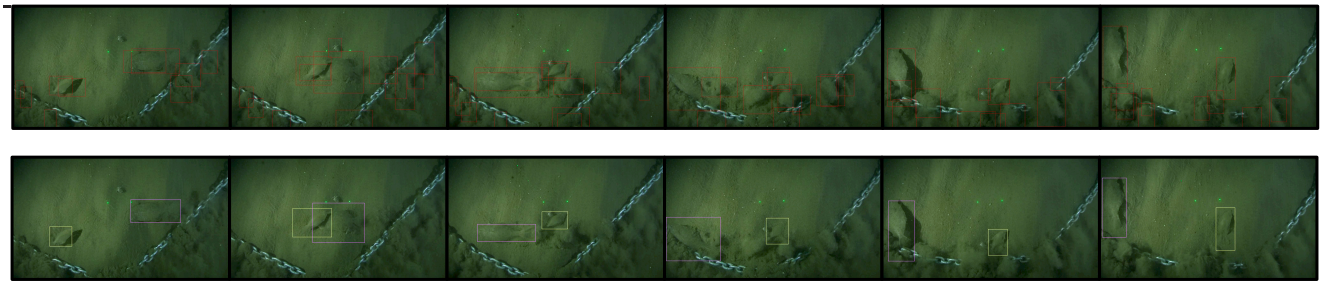

Figure 5. An example video sequence from our large dataset [1]: (top row) Our detections of fish occurrences (red bounding boxes), where the detection threshold is set such that the recall is high. (bottom row) Our multi-target tracking results. As can be seen, our tracker is able to address: rapid changes of fish motion directions, distractors such as whirls of sand and moving chain, and relatively low contrasts between the fish and the background. We also successfully maintain the track IDs (marked with unique colors) even in the case of occlusion.

and $j^{(t)} \in V$ that do not share the same detection, $E = \{(i^{(t)}, j^{(t)}) : i^{(t)} \cap j^{(t)} = \emptyset,\ t = 1, ..., T\}$. Finally, $w : V \rightarrow \mathbb{R}^+$ associates positive weights w_i with every node $i \in V$, defined as similarity by Eq. (1).

Previous work has shown that the tracking problem is equivalent to enumerating maximum weight cliques (MWC) of G [6, 20]. The MWC problem is to find a subset of mutually adjacent vertices (i.e., a clique) having the largest total weight. For tracking, we sequentially enumerate MWCs of G. We first identify the largest MWC of G. Then we eliminate its nodes and edges, and recompute the next MWC. This sequential enumeration is repeated until the resulting MWC becomes prohibitively small (less than 10 video frames). Then, following the above definitions, we link tracklets into distinct tracks, such that a track, $\mathcal{T}$, may contain only one tracklet from each $\Sigma^{(t)}$, $t = 1, ..., T$, and $\mathcal{T}$ may contain two consecutive tracklets $i^{(t)} \in \Sigma^{(t)}$ and $j^{(t+1)} \in \Sigma^{(t+1)}$ only if $i^{(t)}$ ends and $j^{(t+1)}$ starts with the same object detection. In the following, we present a formulation of the MWC problem, and specify a MWC algorithm.

3.1. The Maximum Weight Clique Problem

A subset of V can be represented by an indicator vector $\boldsymbol{x} = (x_i) \in \{0, 1\}^n$, where $x_i = 1$ means that node i is in the subset, and $x_i = 0$ otherwise. Let $\boldsymbol{A}$ denote the weighted adjacency matrix of G, where $\boldsymbol{A}_{ij} = 1 - w_{ij}$, and w_{ij} is the similarity between tracklets i and j, given by (1). Then, from the well-known extension of Motzkin-Straus theorem to weighted graphs, proposed by Pelillo and his collaborators [3], the MWC of G, denoted as $\boldsymbol{x}^*$, can be specified by the following quadratic integer program:

$$\begin{aligned} &\boldsymbol{x}^* = \operatorname{argmax}_{\boldsymbol{x}} \boldsymbol{x}^{\mathrm{T}} A\, \boldsymbol{x}, \\ &\text{s.t. } \forall i \in V, x_i \in \{0, 1\}, \text{ and } \textstyle\sum_i x_i = 1, \end{aligned} \tag{2}$$

where the constraints in (2) mean that $\boldsymbol{x}$ belongs to the standard simplex.

The MWC problem in (2) is known to be NP-hard for arbitrary graphs, and, according to recent theoretical results, so is the problem of approximating it within a constant factor. In the next subsection, we present a parallel, distributed heuristic for approximating the MWC problem based on dynamics principles developed and studied in various branches of mathematical biology. The continuous formulation of the MWC problem naturally maps onto a parallel, distributed computational network whose dynamical behavior is governed by the so-called replicator equations. These are dynamical systems introduced in evolutionary game theory and population genetics to model evolutionary processes on a macroscopic scale. The theoretical guarantees that the solutions provided by our replicator dynamics algorithm are actually the ones being sought are presented in [3].

3.2. The Replicator Dynamics

We relax the integer quadratic problem in (2) to a continuous problem, such that x_i may take real values, $\forall i \in V, x_i \in [0, 1]$. Then, we iteratively seek a solution of the relaxed MWC problem by considering the following dynamical system:

$$x_i(t) = x_i(t) \frac{(\boldsymbol{A}\, \boldsymbol{x}(t))_i}{\boldsymbol{x}(t)^{\mathrm{T}} \boldsymbol{A}\, \boldsymbol{x}(t)},\ \forall i \in V. \tag{3}$$

It is readily seen that the simplex constraints of (2), i.e., $\Delta = \{\forall i \in V, x_i \in [0, 1], \text{ and } \sum_i x_i = 1\}$, remain invariant under the dynamic in (3). That is, every trajectory starting in Δ will remain in Δ for all iterations $t = 1, 2, ...$ Moreover, the stationary points of the dynamic in (3), i.e., the points satisfying $x_i(t + 1) = x_i(t)$, coincide and are the solutions of the equations

$$x_i[(\boldsymbol{A}\, \boldsymbol{x})_i - \boldsymbol{x}^{\mathrm{T}} \boldsymbol{A}\, \boldsymbol{x}] = 0,\ \forall i \in V. \tag{4}$$

A stationary point $\boldsymbol{x}$ is said to be asymptotically stable if every trajectory converges to $\boldsymbol{x}$ as $t \rightarrow \infty$.

Both (3) and (4) are called replicator equations in theoretical biology, since they are used to model evolution over time of relative frequencies of interacting, self-replicating

entities. The replicator dynamics can also be interpreted as a gradual and adaptive equilibrium selection process [3].

For tracking, i.e., identifying the MWC of G, we use the replicator dynamics, given by (3). When the algorithm reaches an asymptotically stable stationary point, the resulting solution is taken as the MWC $\boldsymbol{x}^*$. The corresponding tracklets transitively linked within the obtained $\boldsymbol{x}^*$ represent our track estimate. To find another track, we eliminate from G all nodes in the MWC whose corresponding indicators $x_i^* = 1$, and repeat the replicator dynamics, given by (3), on the remaining graph. The iterations stop when the resulting MWC becomes prohibitively small.

From (3), it is easy to show that the complexity of our tracking algorithm is $O(n^2)$, where n is the number of pairs of fish detections across the video frames.

4. Video classification

This section presents our final third step in which we extract motion features from the estimated fish tracks, and then classify them using the Random Forest(RF) [4].

For each fish track in the video, obtained in the previous step, we compute a fish motion descriptor. The descriptor vector is defined as a log-polar histogram of fish displacements between consecutive frames along the track. Similar to the shape context descriptor, our histogram has 24 bins. A particular displacement is counted in a bin which has the magnitude and angle of the displacement. The histogram thus collects a statistical evidence of fish motions along the corresponding track in the video.

For classifying a new video, we "drop" each motion histogram descriptor through the RF [4]. A majority vote decides the class of the video. Given a training set of labeled examples of motion histograms, the RF grows many decision trees. We view the decision trees as a way of discriminatively structuring evidence about the class distributions in the training set. In particular, each leaf of each tree in RF stores a number of training examples from each class that reached that leaf. When a new motion descriptor is encountered, it is "dropped" down each of the trees in the RF, until it reaches a leaf in every decision tree. The class of the new descriptor is determined as a majority class of the training examples stored in the reached leaf. Finally, the class of the entire new video is determined as a majority class of the training examples stored in all leaves reached by the motion descriptors.

Training: In our implementation, we use five training videos per each of the six classes, where the videos may contain on an average more than 100 fish tracks. The fish tracks are assigned motion histograms, which are then used as training examples for constructing the RF, as described in [4]. In particular, we use the standard random splits of training data to train ten decision trees of the RF, constructed in the top-down way. The growth of each tree is constrained so its depth is less than twenty, and each of its leaf nodes contain at least ten training examples.

5. Results

This section presents our evaluation on 60 underwater videos, each lasting about 30 seconds. The videos are captured by a camera mounted on an underwater trawl sledge, as depicted in Fig. 1. The sledge has a chain that is dragged over the sea bed to stir fish lying on the sand surface. As can be observed in a few example frames in Fig. 1, the videos present many challenges, including dynamic textured background, low contrasts between fish and background, motion blur and occlusion. The dataset is split into 30 training and 30 test videos, where each of the six event classes is represented by five training and five test videos.

As a baseline, we have implemented the state-of-the-art approach, presented in [15], that does not explicitly extract motion tracks of targets, but reasons about the global statistics of all the low-level features extracted from the video. This approach achieves the state-of-the-art results in recognition of complex human activities in movies, and thus we expect that it will also work well on our videos. The approach uses space-time interest points as features, organizes the points into space-time pyramids, and classifies them with multi-channel non-linear SVMs. However, in our implementation, we obtained a relatively poor accuracy rate of 42.3%, averaged over 30 test videos. This suggests that our six event classes require an approach to fine-grained classification, which will explicitly account for particular motion patterns along estimated fish tracks.

Given a video, we first run our fish detector (Sec. 2), then, track the detections (Sec. 3), and finally classify the extracted motion histograms (Sec. 4). For comparison, we also consider state-of-the-art detectors to identify fish occurrences, including: (i) Implicit Shape Model (ISM) [16], (ii) HOG detector [10], and (iii) Deformable part-based model [12] with detection threshold set to -2 for high recall. The same detectors have been also used with success in more standard videos (e.g., [5]). For detection evaluation we use the following metric: ratio of intersection and union of ground-truth and detected bounding boxes around fish, $\rho = \frac{D \cap GT}{D \cup GT}$, where D denotes the area of the detection bounding box, and GT denotes the area of the ground truth bounding box. For $\rho \geq 0.5$ we have true positive (TP), and for $\rho < 0.5$ we have false positive (FP). Table 1 presents our recall and precision results, and the comparison with the three state-of-the-art detectors. As can be seen, the proposed method outperforms the three state-of-the-art detectors both in terms of accuracy and running time. The running times include the computation of low-level features from raw pixels, and a scanning window procedure. The relatively low values of recall and precision of the competing approaches suggest that our underwater video dataset is

Detector	Precision (%)	Recall (%)	Running Time
Ours	$\mathbf{(66.2 \pm 5.4)\%}$	$\mathbf{(64.12 \pm 3.9)\%}$	**187.5s**
ISM [16]	47.2%	59.2%	$\approx$ 5 min
HOG [10]	38.5%	51.4%	$\approx$ 4 min
Part-based [12]	44.1%	56.3%	$\approx$ 8 min

Table 1. Average detection results on our underwater video dataset, and the comparison with the three state-of-the-art detectors.

Tracker	Prec.	Accur.	False Neg.	False Pos.	ID Switch
Ours	**79.6%**	**74.2%**	**15.2%**	**2.7%**	**2.8**
[5]	62.0%	62.9%	16.8%	13.3%	4.6
Ours + Detector [12]	73.2%	65.2%	25.2%	5.1%	5.8

Table 2. CLEAR MOT [2] results on our underwater video dataset, and comparison with the state-of-the-art tracker.

very challenging.

Table 5 presents our CLEAR MOT results [2], and the comparison with the state-of-the-art approach to multitarget tracking presented in [5]. The CLEAR MOT metrics include the following values: Precision — ratio of intersection and union of ground-truth and detected bounding boxes around fish; Accuracy — the sum of false negative rate and false positive rate; and the Number of ID switches per ten seconds of video footage. The tracker of [5] and the tracker used in the proposed method both use the same detections obtained via our fish detector. As can be seen, we significantly outperform the tracker of [5]. It is worth noting that we improve our precision rate after tracking. While our fish detector yields precision of 66.2%, many false positives are eliminated in tracking, resulting in precision of 79.6%. We also ran our tracker on fish detections produced by the part-based detector of [12]. These results are presented in the bottom row of Table 5. As can be observed, our tracker works better in conjunction with our detector, as expected, since our detector yields better precision and recall rates.

After estimating the fish tracks, we computed motion histograms associated with each track. The motion histograms were classified using the RF into six classes (Sec. 4). Our classification rate, averaged over 30 test videos is 79.8%. We have also considered learning the six event classes using a linear Support Vector Machine (SVM) classifier from the 30 training videos. To this end, we used the default parameters of the publicly available software tool LibSVM (Weika). The resulting SVM classification rate is only 58.7%. This suggests that a non-linear classifier is more appropriate for our purposes.

6. Conclusion

We have presented an approach to fine-grained video classification aimed at facilitating a specific biological research. The videos show moving fish. The videos are categorized into six very similar classes, depending on the sea depth and time of the day when the videos are captured. As demonstrated by our experimental results, the state-of-the-art is not able to address such a fine-grained video categorization problem. By contrast, our approach successfully and efficiently identifies six distinct motion patterns of fish. Our approach consists of three main steps: fish detection, tracking, and classification of extracted fish motion features. We have presented a new fish detector that outperforms the state-of-the-art shape- and part-based detectors. Our fish tracking has been formulated as finding the maximum weight clique of a graph composed of pairs of fish detections from all consecutive video frames. We have characterized the motions of tracked fish by histograms of fish displacements. These histograms have been classified with the Random Forest for video classification. The presented approach advances the state of the art, and presents a viable solution for automated video analytics in highly specialized studies, like those in marine biology.

Acknowledgement

S. Todorovic acknowledges the support of the National Science Foundation under grant NSF IIS 1018490. The UH team was supported in part by the UH Hugh Roy and Lillie Cranz Cullen Endowment Fund. L. Ciannelli acknowledges support from the Oregon Sea Grant R/ECO-23.

References

[1] M. Amer, S. Todorovic, E. Bilgazyev, S. Shah, I. Kakadiaris, and L. Ciannelli. Underwater video dataset, Jun. 2011.

[2] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the $CLEAR\ MOT$ metrics. *Image Video Processing*, 1:1–10, 2008.

[3] I. M. Bomze, M. Pelillo, and V. Stx. Approximating the maximum weight clique using replicator dynamics. *IEEE Transactions on Neural Networks*, 11(6):1228–1241, 2000.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *Proc. IEEE International Conference on Computer Vision*, Kyoto, Japan, Sept. 27–Oct. 4 2009.

[6] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Proc. IEEE International Conference on Computer Vision*, Barcelona, Spain, Nov. 6–13 2011.

[7] T. Brox, A. Brunh, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. IEEE European Conference in Computer Vision*, Prague, Czech Republic, May, 11–14 2004.

[8] L. Ciannelli, P. Fauchald, K. Chan, V. Agostini, and G. Dingsor. Spatial fisheries ecology: recent progress and future prospects. *Journal of Marine Systems*, 71:1–10, 2007.

[9] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

[10] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, Jun. 20-25 2005.

[11] M. Decker, L. Ciannelli, K. Chan, C. Ladd, H. Liu, and R. Bordeur. Changes in the biomass and distribution of bering sea jellyfish in relation to regime shifts, 2009. Aquatic Sciences Meeting.

[12] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, Jun. 13-18 2010.

[13] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proc.* 10^{th} *European Conference on Computer Vision*, Marseille, France, Oct. 12-18 2008.

[14] C. H. Kuo, C. Huang, and R. Nevitia. Multi-target tracking by on-line learned discriminative appearance models. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, Jun. 13-18 2010.

[15] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.

[16] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal on Computer Vision*, 77(3):259–289, 2008.

[17] B. Leibe, K. Schindler, and L. V. Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proc.* 11^{th} *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, Oct. 14-20 2007.

[18] Y. Li, C. Huang, and R. Nevatia. Learning to associate: hybridboosted multi-target tracker for crowded scene. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, Jun. 20-25 2009.

[19] Y. Li and R. a. Nevatia. Key object driven multi-category object recognition, localization and tracking using spatio-temporal context. In *Proc.* 10^{th} *European Conference on Computer Vision*, Marseille, France, Oct. 12-18 2008.

[20] N. Shafique and M. W. Haering. A rank constrained continuous formulation of multi-frame multi-target tracking problem. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.

[21] D. Sohn, L. Ciannelli, and J. Duffy-Anderson. Comparison of early life history stages of pacific halibut and greenland halibut in the eastern bering sea: abundance, distribution, and drift pathways, 2010. Ocean Science Meeting.

[22] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal on Computer Vision*, 75(2):247–266, 2007.

[23] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, Jun. 20-25 2009.

[24] R. Xu and D. Wunsch. Survey on clustering algorithms. *2005*, 16:645–678, IEEE Transactions on Neural Networks.

[25] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computation Survey*, 38(4):13–21, 2006.

[26] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, Jun. 24-26 2008.