COVARIANCE BASED FISH TRACKING IN REAL-LIFE UNDERWATER ENVIRONMENT

Concetto Spampinato[1], Simone Palazzo[1], Daniela Giordano [1], Isaak Kavasidis[1] and Fang-Pang Lin[2] and Yun-Te Lin[2]

[1] *Department of Electrical, Electronics and Computer Engineering, University of Catania, Viale Andrea Doria, 6 -95125, Catania, Italy*

[2] *National Center of High Performance Computing, No. 7, R&D 6th Rd., Hsinchu Science Park, Hsinchu City, Taiwan*

{cspampin, palazzosim, dgiordan,isaak.kavasidis}@dieei.unict.it, {fplin, lsi}@nchc.narl.org.tw

Keywords: Object tracking under Extreme Conditions, Covariance Approach, Intelligent Underwater Video Analysis

Abstract: In this paper we present a covariance based tracking algorithm for intelligent video analysis to assist marine biologists in understanding the complex marine ecosystem in the Ken-Ding sub-tropical coral reef in Taiwan by processing underwater real-time videos recorded in open ocean. One of the most important aspects of marine biology research is the investigation of fish trajectories to identify events of interest such as fish preying, mating, schooling, etc. This task, of course, requires a reliable tracking algorithm able to deal with 1) the difficulties of following fish that have multiple degrees of freedom and 2) the possible varying conditions of the underwater environment. To accommodate these needs, we have developed a tracking algorithm that exploits covariance representation to describe the object's appearance and statistical information and also to join different types of features such as location, color intensities, derivatives, etc. The accuracy of the algorithm was evaluated by using hand-labeled ground truth data on 30000 frames belonging to ten different videos, achieving an average performance of about 94%, estimated using multiple ratios that provide indication on how good is a tracking algorithm both globally (e.g. counting objects in a fixed range of time) and locally (e.g. in distinguish occlusions among objects).

1 INTRODUCTION

Typically marine biologists study fish populations in their natural habitat using casting nets in the ocean, human underwater observation and photography (Rouse, 2007), combined net casting and acoustic (sonar) (Brehmera et al., 2006) and human hand-held video filming. However these approaches either are invasive (such as the net casting method) or provide scarce information (such as photography). In order to overcome these limitations, underwater cameras have been widely used in the last years, since they do not influence fish behavior and also provide large amount of video material (with cameras active day and night, the only limitation is the amount of mass memory required to store the videos). On the other hand, it is impractical to manually analyze this huge quantity of video data, both because it requires a lot of time and also because it is error prone – it is unrealistic to assume people can fully investigate all the information in the videos. Therefore, automatic video analysis methods are heavily demanded such as the one devised in the Fish4Knowledge[1] project, which uses live video feeds from ten underwater cameras located in the coral reefs of Taiwan's shores and aims at developing an automatic system for integrated data capturing, video analysis, fish detection and classification, and querying, for the marine biologists to use, in order to study fish populations, behavior and interactions.

The main difficulty in this kind of tasks is the nature of the videos to be processed. Traditionally, such tasks have involved the analysis of videos taken in controlled environments, such as tanks (Morais et al., 2005; Petrell et al., 1997), where for example lighting conditions do not change with time, the background is static to simplify fish detection, the type of fish is known, etc. The lack of these assumptions greatly complicates the task to be accomplished and requires the development of automatic analysis methods which are robust enough to handle all the possible varying conditions of the environment. In this

[1] *http://fish4knowledge.eu*

direction, key roles are played by image preprocessing (e.g. (Cannavò et al., 2006)), object detection, tracking and recognition (e.g. (Spampinato, 2009) and (Spampinato et al., 2010)).

One aspect to deal with when analyzing marine ecosystems is fish tracking, whose importance goes beyond simple population counting. In fact, behavior understanding and fish interactions' analysis, which are interesting perspectives for marine biologists to study, strictly rely on trajectories extracted using tracking approaches. However, tracking presents a few major difficulties, which become greater in underwater environments where objects have multiple degrees of freedom or when the scene conditions cannot be controlled.

Many different approaches have been studied in literature on how to solve the visual tracking problem such as Kalman filter-based tracking (Doucet et al., 2001), particle filter tracking (Gordon et al., 1979), point feature tracking, mean-shift tracking (Comaniciu and Meer, 2002). However, to the best of our knowledge, only a variation of mean-shift, the CAMSHIFT (Bradski, 1998), has been applied to underwater environments (Spampinato et al., 2008) achieving an average tracking performance (estimated as correct counting rate) of about 85%. However, the CAMSHIFT shows a main drawback when dealing with fish-fish and fish-background occlusions mainly due to the fact that it exploits only color information. In this paper we propose a tracking algorithm where fish are modeled as covariance matrices (Tuzel et al., 2006) of feature built out of each pixel belonging to the fish's region. This representation allows to embody both spatial and statistical properties of non-rigid objects, unlike histogram representations (which disregard the structural arrangement of pixels) and appearance models (which ignore statistical properties). As shown in the experimental results section, the performance of the proposed approach is very encouraging and better that the ones achieved with CAMSHIFT, thus also indicating how our covariance based approach performs very well under extreme conditions.

The remainder of the paper is: Section 2 describes the details of the proposed covariance based fish tracking algorithm; Section 3, instead, shows the achieved tracking results with hand-labeled ground truth data. Finally, Section 4 points out the concluding remarks.

2 COVARIANCE BASED TRACKING ALGORITHM

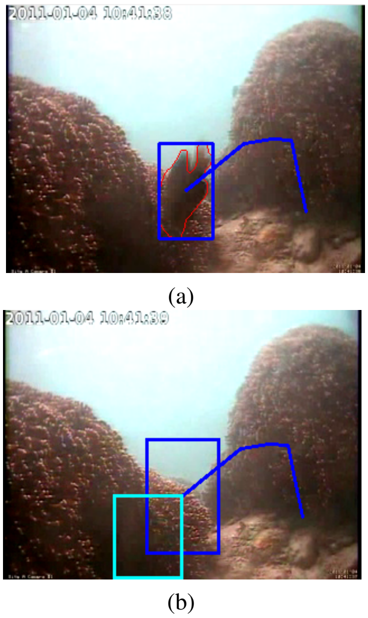
In the following description, we use "tracked object" to indicate an entity that represents a unique fish and contains information about the fish appearance history and its current covariance model; and "detected object" to indicate a moving object, which has not been associated to any tracked object yet. For each detected object, the corresponding covariance matrix is computed by building a feature vector for each pixel, made up of the pixel coordinates, the RGB and hue values and the mean and standard deviation of the histogram of a 5×5 window with the target pixel as centre. The covariance matrix, which models the object, is then computed from this feature vector and associated to the detected object. Afterwards, this matrix is used to compare the object with the currently tracked objects, in order to decide which one it resembles the most. The main issue in comparing covariance matrices lies in the fact that they do not lie on the Euclidean space—for example, the covariance space is not closed under multiplication with negative scales. For this reason, as suggested in (Porikli et al., 2005), we used Förstner's distance (Forstner and Moonen, 1999), which is based on generalized eigenvalues, to compute the similarity between two covariance matrices:

$$\rho\left(C_i, C_j\right) = \sqrt{\sum_{k=1}^{d} ln^2 \lambda_k \left(C_i, C_j\right)} \qquad (1)$$

where d is the order of the matrices and $\left\{\lambda_k\left(C_i, C_j\right)\right\}$ are the generalized eigenvalues of covariance matrices C_i and C_j, computed from

$$\lambda_k C_i x_k - C_j x_k = 0 \quad k = 1 \cdots d \qquad (2)$$

The model of each tracked object is then computed as a mean (based on Lie algebra (Porikli et al., 2005)) of the covariance matrices corresponding to the most recent detections of that object. In order to deal with occlusions, the algorithm handles the temporary loss of tracked objects, by keeping for each of them a counter (TTL) of how many frames it has been missing; when this counter reaches a user-defined value (for 5-fps videos, the best value, obtained empirically, was 6), the object is considered lost and discarded. In order to decide whether a detected object is a feasible candidate as the new appearance of a current tracked object, we check if the detected object's region overlaps, at least partially, with the tracked object's search area, which by default is equal to the bounding box of that object's latest appearance. To manage the temporary loss of an object, and the fact that while the object has not been detected it might have moved away from its previous location, we modify the search area, expanding it proportionally to the number of frames where the object has been missing. In this case, the search area is made



(a)

(b)

Figure 1: Search area expansion: when an object (1(a)) is lost, the corresponding search area (1(b)) is composed of a “central search area” (in dark blue) and a “directional search area” (in lighter blue), oriented towards the estimated direction of the fish.

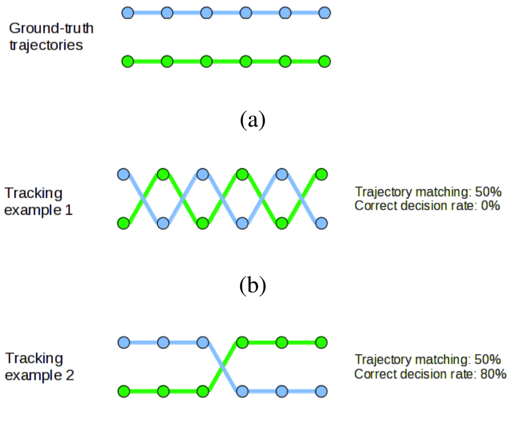
up of two rectangles: a “central search area”, which is a rectangle centered at the object’s previous location, initialized to the object’s previous bounding box and expanded evenly in all directions; and a “directional search area” which takes into consideration the recent motion trend of the object to give a simple estimate of its direction, and which consists of a rectangle with a vertex at the object’s previous location and the correspondent opposite vertex located accordingly with the estimated object’s direction. An example of the search area expansion mechanism is shown in Figure 1. The steps performed by the proposed tracking algorithm are shown in Algorithm 1.

Algorithm 1 Pseudo-code implementation of the tracking algorithm

```
detected_objects ← runDetectionAlgorithm()
tracked_objects ← getCurrentlyTrackedObjects()
feasible_associations ← ∅
Compute covariance matrix for each detected object
for $D_j$ in detected_objects do
    $D_j$.computeCovarianceMatrix()
end for
Compute set of possible associations between tracked objects and detected objects
for $T_i$ in tracked_objects do
    for $D_j$ in detected_objects do
        if $T_i$.getCurrentPosition() overlaps $D_j$.getBlob() then
            $d_{ij}$ ← computeCovarianceDistance($T_i$.currentModel(),$D_j$.covarianceMatrix())
            feasible_associations.add($(T_i, D_j, d_{ij})$)
        end if
    end for
end for
sortByDistance(feasible_associations)
Assign each detected object to the covariance-closer tracked object
for $(T_i, D_j, d_{ij})$ in feasible_associations do
    if not $D_j$.isAssigned() then
        $D_j$.assignTo($T_i$)
        $T_i$.updateModel()
    end if
end for
If a tracked object has been missing for too many frames, remove it
for $T_i$ in tracked_objects do
    if $T_i$.foundInCurrentFrame() then
        $T_i$.resetTTL()
    else
        $T_i$.decreaseTTL()
        if $T_i$.getTTL() = 0 then
            tracked_objects.remove($T_i$)
        end if
    end if
end for
Add new tracked objects
for $D_j$ in detected_objects do
    tracked_objects.createNew($D_j$)
end for
```

3 RESULTS

To test the proposed algorithm we used 10 sample underwater videos. Each videoclip was 10 minutes long, sampled at 320×240 with a 24-bit color depth, at a frame rate of 5 fps. In total we had 30000 frames and 1262 unique fish. The recorded scenes were featured by: 1) sudden and gradual light changes together with the periodical gleaming typical of underwater scenes, 2) bad weather conditions (e.g. cloudiness, storms and typhoons) that affect image contrast, 3) murky water: the clarity of the water changed during the day due to the drift and the presence of plankton and 4) multiple fish occlusions: due to the absence of the third dimension (we process 2D images) a lot of occlusions among fish were present in the analyzed scenes. For each video, ground-truth data (against which we compared the output of the algorithm) was hand-labeled. For each frame, the set of significant (i.e. large enough to be clearly identified as a fish) fish was selected, and for each of such fish its position and contour was hand-drawn. Tracking identification numbers (IDs) were used to label detections in different frames as belonging to the same unique fish, in such a way that fish which underwent temporary occlusions would then be re-assigned to the same tracking ID. We directly fed our tracking algorithm with the detection data (i.e. the fish) provided by the ground truth, so that the tracking results would not be influenced by detection errors. The performance of our algorithm was also compared with the one achieved by the CAMSHIFT since it is the only approach tested on underwater videoclips in (Spampinato et al., 2008). To assess the *ground-truth-vs-algorithm* comparison we adopted the following metrics, which are based on the ones existing in the literature (Bashir and Porikli, 2006), but that describe the performance of a tracking algorithm both globally, at the trajectory level (e.g. the correct counting rate and



(a)

(b)

(c)

Figure 2: Difference between the trajectory matching score and the correct decision rate. Fig. 2(a) shows two ground truth trajectories of two fish, whereas the other two images represent two examples of tracking output. In Fig. 2(b), although the tracker fails at each tracking decision the trajectory matching score is 50%, whereas the correct decision rate is 0. Differently, in Fig. 2(c) the tracker fails only in one step and the trajectory matching score is 50% (as the previous case) whereas the correct decision rate is 80% (4 correct associations out of 5).

the average trajectory matching), and locally, at the single tracking decision level (e.g. the Correct decision rate):

- *Correct counting rate (CCR)*: percentage of correctly identified fish out of the total number of ground-truth fish.
- *Average trajectory matching (ATM)*: average percentage of common points between each ground-truth trajectory and its best-matching tracker-computed trajectory.
- *Correct decision rate (CDR)*: let a "tracking decision" be an association between a fish at frame t_1 and a fish at frame t_2, where $t_1 < t_2$; such tracking decision is correct if it corresponds to the actual association, as provided by the ground truth. The correct decision rate is the percentage of correct tracking decisions, and gives an indication on how well the algorithm performs in following an object, which is not necessarily implied by the average trajectory matching (see Figure 2).

Table 1 shows the results obtained by the covariance tracking algorithm compared to the ones achieved by the CAMSHIFT algorithm, in terms of the above-described indicators. It is clear how our approach performs better than CAMSHIFT and also has a very good absolute accuracy, being able to correctly identify more than 90% of unique objects with a very high degree of correspondence to the ground-truth trajectories.

Table 1: Comparison between the results obtained by the proposed algorithm and CAMSHIFT on the ground-truth data.

	Covariance tracker	CAMSHIFT
CCR	91.3%	83.0%
ATM	95.0%	88.2%
CDR	96.7%	91.7%

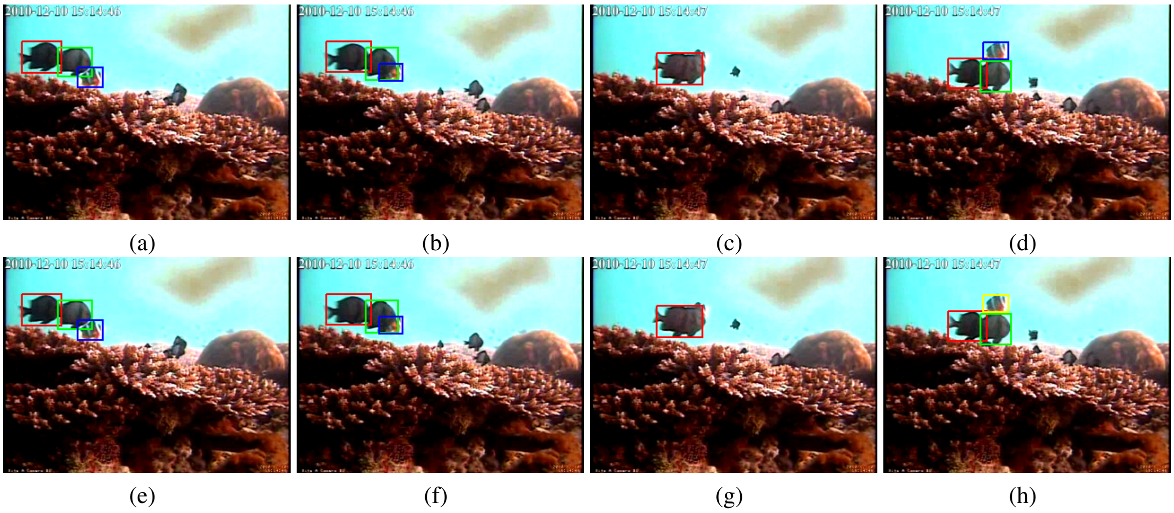
Figure 3 shows an example of how the proposed algorithm and CAMSHIFT handle fish-fish occlusions. It is possible to notice that the covariance tracker is able to correctly follow the clownfish (in the blue box) after it is hidden behind the other fish, whereas in the same frame sequence CAMSHIFT is not able to identify the clownfish's appearances as belonging to the same unique fish, and after the occlusion it starts tracking it as a new object.

4 CONCLUDING REMARKS

In this work we tackled the problem of fish tracking, which shows several difficulties due to the unconstrained environment, the uncontrolled scene conditions and the nature of the targets to be tracked, i.e. fish, whose motion tends to be erratic, with sudden direction and speed variations, and whose appearance can undergo quick changes. In order to deal with these problems, the approach we adopted is based on a covariance-based modeling of objects, which has proved to be suitable for tracking non-rigid objects in a noisy environment, by representing an object's spatial and statistical information in a unique compact structure, the covariance matrix. The performance evaluation showed that the proposed tracker outperforms CAMSHIFT (previously applied to the same scenes) and is able to correctly detect more than 90% of the objects, with a correct decision rate higher than 96%. Since this approach has proved to be very effective on real-life underwater environments, further developments on this work will investigate the use and adaption of this algorithm in different contexts, e.g. pedestrian or vehicle tracking in urban environments (Faro et al., 2008) and (Faro et al., 2011).

5 ACKNOWLEDGMENTS

This research was funded by European Commission FP7 grant 257024, in the Fish4Knowledge project (*www.fish4knowledge.eu*).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3: Tracking results on three occluding fish with the proposed approach (top row) and CAMSHIFT (bottom row). We can see that the CAMSHIFT tracker fails to recognize that the fish in the yellow box is the same as the one in the blue box.

REFERENCES

Bashir, F. and Porikli, F. (2006). Performance Evaluation of Object Detection and Tracking Systems. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS2006)*.

Bradski, G. R. (1998). Computer Vision Face Tracking For Use in a Perceptual User Interface. *Intel Technology Journal*, (Q2).

Brehmera, P., Chib, T. D., and Mouillotb, D. (2006). Amphidromous fish school migration revealed by combining fixed sonar monitoring (horizontal beaming) with fishing data. *Journal of Experimental Marine Biology and Ecology*, 334(1):139–150.

Cannavò, F., Giordano, D., Nunnari, G., and Spampinato, C. (2006). Variational method for image denoising by distributed genetic algorithms on grid environment. In *Proc. of WETICE-2006*, pages 227 – 232.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo methods in practice*. Springer Verlag.

Faro, A., Giordano, D., and Spampinato, C. (2008). Evaluation of the trac parameters in a metropolitan area by fusing visual perceptions and cnn processing of webcam images. *IEEE Transactions on Neural Networks*, 19(6):1108–1129.

Faro, A., Giordano, D., and Spampinato, C. (2011). Integrating location tracking, traffic monitoring and semantics in a layered its architecture. *IET Intelligent Transport Systems*, 5(3):197–206.

Forstner, W. and Moonen, B. (1999). A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University.

Gordon, N., Doucet, A., and Freitas, N. (1979). An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control*, 24(6):843–854.

Morais, E. F., Campos, M. F. M., Padua, F. L. C., and Carceroni, R. L. (2005). Particle filter-based predictive tracking for robust fish counting. *Computer Graphics and Image Processing, Brazilian Symposium on*, 0:367–374.

Petrell, R., X.Shi, Ward, R., Naiberg, A., and Savage, C. (1997). Determining fish size and swimming speed in cages and tanks using simple video techniques. *Aquacultural Engineering*, 16(63-84).

Porikli, F., Tuzel, O., and Meer, P. (2005). Covariance tracking using model update based on lie algebra. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*.

Rouse, W. (2007). Marine biology research experiment: Population dynamics of barnacles in the intertidal zone.

Spampinato, C. (2009). Adaptive objects tracking by using statistical features shape modeling and histogram analysis. In *Seventh International Conference on Advances in Pattern Recognition*, pages 270–273.

Spampinato, C., Chen-Burger, Y.-H., Nadarajan, G., and Fisher, R. B. (2008). Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *VISAPP (2)*, pages 514–519.

Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H. J., Fisher, R. B., and Nadarajan, G. (2010). Automatic fish classification for underwater species behavior understanding. In *Proceedings of the ACM ARTEMIS 2010*, pages 45–50, New York, NY, USA. ACM.

Tuzel, O., Porikli, F., and Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. *Proc. 9th European Conf. on Computer Vision*.