COVARIANCE BASED MODELING OF UNDERWATER SCENES FOR FISH DETECTION

S. Palazzo, I. Kavasidis, C. Spampinato

Department of Electrical, Electronics and Computer Engineering
University of Catania – Italy
{simone.palazzo, kavasidis, cspampin}@dieei.unict.it

ABSTRACT

In this paper we present an algorithm for visual object detection in a underwater real-life context which explicitly models both the background and the foreground for each frame – thus helping to avoid foreground absorption into similar background –, and integrates both colour and texture features (which have proved effective in overcoming the limitations of colour-only appearance descriptors) into a covariance-based model, which provides an elegant way to merge multiple features together and enforce structural relationships. A joint domain-range model combined to a post-processing approach based on Markov Random Field takes into account the spatial dependency between pixels in the classification process, unlike the classical pixel-oriented modeling techniques. Our results show the effectiveness of this approach in the underwater environment, which presents a lot of variety in scene conditions, objects' motion patterns, shapes and colouring, and background activity.

1. INTRODUCTION

The recent technological progress in digital video analysis, information storage capabilities and high-speed computer networks, along with the decrease of the costs of camera devices, have led to a wide diffusion of video-surveillance [1, 2]. Lately, the use of cameras to record video clips of wildlife environments is gaining more and more attention: in fact, this is a non-obtrusive way of studying animal populations in their natural habitats, for reasons which may include endangered species monitoring and hunt/fishing control (e.g. the AQUACAM project[1]), community landscape management and sustainable management of natural resources (e.g. the ECDD Comoro Islands project[2]), ecosystem health evaluation, and the study of animal individual and population behaviour (the latter two being part of the purposes of the Fish4Knowledge project[3], e.g. see [3]). This approach is expected to produce a huge amount of video data available to biologists and researchers, which however may turn out to be a disadvantage when it comes to analyzing it. In order to tackle this problem, many solutions for automatic video processing have been proposed by the scientific community, aiming at minimizing human supervision in the video analysis task, thus reducing drastically the amount of time required to extract information from the available material.

All of these approaches necessarily rely on *object detection* modules, which produce evidence on the presence or absence of potential targets in a video sequence, typically by identifying moving elements in the scene. The classical tendency in the development of a motion detection algorithm consists in applying *background modeling* techniques to build an estimated image of the scene without objects of interest, and comparing each new video frame to this model to identify *foreground* areas. However, backgrounds rarely are perfectly stationary, because of noise, camera motion, environmental conditions, and other factors. Moreover, modeling only the background implies that if an object – or part thereof – has similar colors as the background region on which it lies, it will unavoidably be detected as part of it. These considerations lead to two conclusions: first of all, it is necessary to explicitly model the foreground as well; in this way, for example, a dark red object moving from a blue background to a light red one will be still detected due to the closer resemblance to the foreground model than the background one. Furthermore, this example proves how color similarities may not be sufficient to discriminate between background and foreground pixels accurately; other works, such as [4], showed the importance of introducing texture information in the models, to further accentuate the differences between regions belonging to distinct objects. This requirement is even more important in real-life unconstrained contexts, as highlighted in Porikli's characterization of object detection algorithms under extreme conditions [5].

In this work, we propose an approach for modeling both the foreground and the background and for integrating texture features into the models by using covariance matrices, due to their intrinsic capability to describe both spatial and statistical information of an image region. At each new frame, a Kernel Density Estimation (KDE) exploits the current models to build two probabilistic maps which are transformed into a binary motion map by means of a MAP-MRF approach.

[1] http://c-fish.org/what-we-do/aquacam-research-programme/

[2] http://www.ecddcomoros.org/

[3] http://www.fish4knowledge.eu

The proposed algorithm was evaluated on real-life underwater videos in uncontrolled and unconstrained conditions, which is a much harsher application context than most human/urban-centered environments. In fact, underwater videos are typically characterized by the presence of moving background (e.g. plants), fast lighting changes (due to the gleaming of the sun on the water surface and the sea bed), low contrast images (murky water, algae on the camera lens, storms and typhoons, etc) and in general have a relative low quality in terms of image resolution and video frame rate, due to bandwidth limitations between the cameras and the storage servers.

In the remainder of this paper, Section 2 describes in detail the modeling and detection processes; Section 3 shows the performance of the algorithms, computed on a set of manually labeled underwater videos; finally, Section 4 presents, respectively, conclusive considerations on the proposed work and some ideas concerning its current limitations and the future developments we plan to experiment.

2. METHOD

The joint domain-range model described herein is inspired by the work presented in [6]. In the original work, the background and foreground models are defined in a 5-dimensional pixel feature space, namely the (x, y) coordinates and the (R, G, B) colour channels. In order to describe our variant of the model, it is necessary to explain how we represent each pixel's information first.

Given the pixel p at location (x, y), a squared neighbourhood of size $w \times w$ is extracted, and for each pixel belonging to this subset a feature vector is computed, which contains: the (x, y) coordinates, the (R, G, B) colour channels; the H channel in the HSV colour space; the first four statistical moments of the Local Binary Patterns (LBP) [7] histogram over the pixel's neighbourhood.

Given this set of feature vectors, the corresponding covariance matrix $C_{x,y}$ is computed. In order to convert this structure to a scalar value (which will be used as main appearance feature in the joint domain-range model), we give an overall estimate of the pixel's neighbourhood variance as:

$$v_{x,y} = \sqrt{\sum_{i=0}^{n} \lambda_{x,y,i}^2} \tag{1}$$

where n is the order of $C_{x,y}$ (i.e. the length of the feature vectors used to compute it) and $\{\lambda_{x,y,i}\}_{i=1\dots n}$ is the set of $C_{x,y}$'s eigenvalues.

Each pixel $\boldsymbol{p}$ is then represented as the (x, y, v) vector, and the joint domain-range model consists in the corresponding 3-dimensional space, on which the *pdf*s of the background and foreground models are built. This is performed by means of Kernel Density Estimation [8]: given the sets $\psi_b = \{\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_n}\}$ and $\psi_f = \{\boldsymbol{f_1}, \boldsymbol{f_2}, \dots, \boldsymbol{f_m}\}$, respectively containing all background and foreground samples, the corresponding *pdf*s can be approximated as:

$$P\left(\boldsymbol{p}|\psi_b\right) = \frac{1}{n} \sum_{i=1}^{n} \varphi\left(\boldsymbol{p} - \boldsymbol{b_i}\right) \tag{2}$$

$$P\left(\boldsymbol{p}|\psi_f\right) = \frac{1}{m} \sum_{i=1}^{m} \varphi\left(\boldsymbol{p} - \boldsymbol{f_i}\right) \tag{3}$$

where $\varphi\left(\boldsymbol{x}\right)$ is a KDE kernel function with the usual properties of unitary integral, symmetry, zero-mean and with identity covariance, such as a multivariate Gaussian.

In order to reduce the dimensionality of the model matrices and the frame processing time, the Binned KDE [9] is used, i.e. the image space ((x, y) coordinates) and the appearance space (v coordinate) are quantized into a relatively small $X \times Y \times V$ space. Of course, this also requires that the KDE kernel be discretized; in this work, a vector kernel along the v dimension is used. Therefore, the model structures we use are two P_b and P_f matrices, having size $X \times Y \times V$, representing at all times the current values of $P\left(\boldsymbol{p}|\psi_b\right) = P_b(x, y, v)$ and $P\left(\boldsymbol{p}|\psi_f\right) = P_f(x, y, v)$, respectively.

This representation allows to achieve three objectives: first of all, a spatial dependency relationship between pixels is introduced, due to KDE; secondly, the foreground model is managed separately from the background's, according to the principles mentioned in Section 1; finally, texture features are seamlessly integrated to colour information into a unique description.

2.1. Model Creation

When the algorithm is started, the first N frames are used to initialize the background model. For each pixel in each frame, a (x, y, v) vector is computed – appropriately quantized for the Binned KDE – and the discrete KDE kernel is applied at its location, thus increasing the $\{P_b\left(x, y, v \pm \Delta v\right)\}_{\Delta v=0\dots n}$ model cells, where $2n + 1$ is the length of the kernel vector, by the appropriate quantity (e.g. the maximum value in the center, at $\Delta v = 0$, and progressively decreasing values as Δv increases). After this procedure has been completed for all pixels, the P_b matrix is normalized by the total number of pixels used for the initialization.

At this stage, the background model P_f is not left empty, although no foreground pixels have been detected yet, but is set to $P_f\left(x, y, v\right) = \gamma$, for each (x, y, v) cell in the model, where γ is a low value (in this work we used 0.1), accounting for the possibility of observing any uniformly distributed pixel value at any locations. The background update procedure, which will take into account the properties of the objects which appear in the following frames, is described in Section 2.3.

2.2. Classification

As new video frames become ready, the current appearance of the observed scene is analyzed to identify areas which present (non-background) motion. In particular, the probabilities that each pixel belongs to either the background or the foreground are computed. Thanks to the discrete KDE representation of the models, such computation is straightforward, since the probability that pixel $\boldsymbol{p} = \{x, y, v\}$ belongs to the background or the foreground models are simply $P_b(x, y, v)$ and $P_f(x, y, v)$, respectively.

Then, a candidate motion binary map $M(x, y)$ is built where each pixel is classified according to the log-likelihood ratio:

$$M(x, y) = \begin{cases} 0 & \text{if} - \ln \frac{P_b(\boldsymbol{p})}{P_f(\boldsymbol{p})} > T \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

where $\boldsymbol{p}$ is the pixel vector at location (x, y); 0's and 1's in the output motion map represent background and foreground pixels in the current video frame. However, at this stage object contours may not not be perfectly identified and noisy spots are present in the motion map; for these reasons, a post-processing step is required. This step aims at verifying that the detected regions satisfy certain empirical requirements of real-world objects, e.g. minimum size and spatial coherency. In our work we use the MAP-MRF approach described in [6] to process the log-likelihood ratio map, and compute a final foreground map by removing mis-detections, filling and smoothing contours, while preserving object boundaries.

2.3. Model Update

After pixel classification has been completed, it is necessary to update the background and foreground models in view of the current image data.

2.3.1. Background update

The background update procedure consists in integrating the current frame's classification results into the KDE estimation, namely the $P(\boldsymbol{p}|\psi_b) = P_b(x, y, v)$ function, with ψ_b representing the current background KDE support points. In order to take into account the possibility that new objects appear in the scene (or that background pixels are misclassified), we update the background model with all pixels in the current image. We call this set $\psi_{b,\text{curr}}$. The $P(\boldsymbol{p}|\psi_{b,\text{curr}}) = P_{b,\text{curr}}(x, y, v)$ function is computed from the $\psi_{b,\text{curr}}$ set using KDE estimation, the same way as the background initial model was computed in Section 2.1. A weighted mean between the current background model $P_b(x, y, v)$ and the $P_{b,\text{curr}}(x, y, v)$ distribution computed from the current frame is applied to compute the new background model:

$$P_{b,\text{new}}(x, y, v) = \alpha P_{b,\text{curr}}(x, y, v) + (1 - \alpha) P_b(x, y, v) \quad (5)$$

2.3.2. Foreground update

The foreground model is recomputed every time from the ψ_f set of pixels detected as foreground in the current frame. As for the background, KDE is applied to estimate the $P_f(\boldsymbol{p}|\psi_f) = P_f(x, y, v)$ *pdf*. Similarly to what was done at the model initialization step (Section 2.1), a small γ constant value is added to $P_f(x, y, v)$ (after which normalization is performed) to account for the appearance of new objects in the frame.

3. EXPERIMENTAL RESULTS

The proposed approach was tested using the I2R dataset [10] which comprises 9 videos, recorded using a static camera, depicting a variety of scenes featured by camera motion, dynamic textures, and cyclic motion. We then compared our approach with recent state-of-the-art approaches, in particular with MoG, the bayesian model proposed by Sheikh *et al.* [6], the complex foreground model [10] , SILTP [4] and the model described by Narayana *et al.* in [11]. The ground truth on the standard datasets was available with the datasets themselves and consisted of 20 frames manually labeled for each video sequence. The results were measured in the terms of *F-Measure* defined as:

$$F = \frac{2 \times recall \times precision}{recall + precision} = \frac{2 \times TP}{2TP + FN + FP} \quad (6)$$

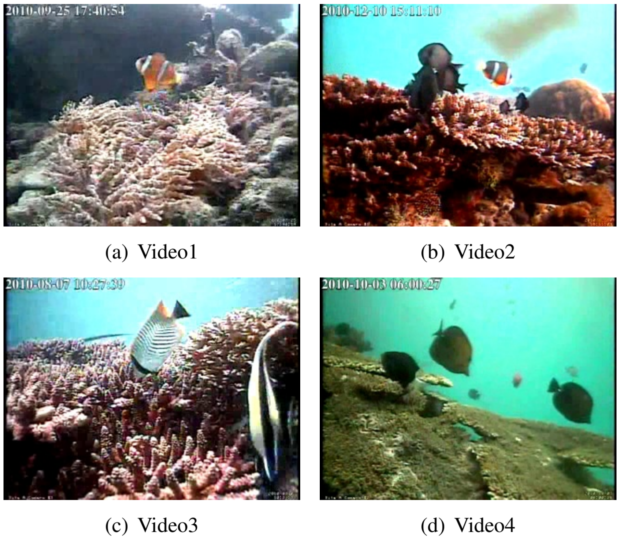
where TP, FP and FN are, respectively, true foreground pixels, false positives and false negatives. The final F score is averaged between all the ground truth frames. We set the algorithms' parameters as follows: $T = 5$, $w = 7$ and $X = 24$, $Y = 36$, $V = 100$. However, an adaptive mechanism has been also adopted to change the parameters according to the likelihood masks and image regions. This was justified by the fact that, whereas, for uniform backgrounds (e.g. the street or the river's water) the number of bins of the model can be kept low , in case of complex and dynamic backgrounds (e.g. the trees or algae) this number must be sensibly higher, in order to increase the cutoff frequency of the model given the high frequencies in these regions. Table 1 shows the achieved F-measures compared with some state of the art methods. Although in some cases our algorithm performs slightly worse than other approaches, on average it outperforms the state of the approaches and its performance are more stable (lower standard deviation) across the different scenes.

To test the effectiveness of the approach in detecting fish in extremely complex and dynamic underwater scenes, we used four videos 320x240 with 5 fps taken from the Fish4Knowledge repository. Fig. 1 depicts the analyzed scenes with *Video1* showing scenes with strong periodic background movements, *Video2* complex background and dynamic textures, *Video3* changes of luminosity and *Video4* low contrasted scenes.

The ground truth was hand-labeled by using the tool proposed in [12]. Benchmark comparisons are provided for

Video	MoG	Complex Foreground [10]	Sheikh's method [6]	SILTP [4]	Narayana's method [11]	Our Method
AirportHall	57.86	50.18	59.21	68.02	71.28	**73.43**
Bootstrap	54.07	60.46	39.23	72.90	76.89	**77.23**
Curtain	50.53	56.08	59.74	92.40	**94.07**	91.18
Escalator	36.64	32.95	44.57	**68.66**	49.43	67.49
Fountain	77.85	56.49	57.31	85.04	85.97	**88.12**
ShoppingMall	66.95	67.84	71.24	79.65	**83.03**	75.67
Lobby	68.42	20.35	47.36	**79.21**	60.82	73.15
Trees	55.37	75.40	62.41	67.83	**87.85**	85.62
WaterSurface	63.52	63.66	84.66	83.15	**92.61**	89.71
Average	59.02	53.71	58.41	77.42	77.99	80.17

Table 1. F-Measure for different methods with the I2R datasets.



(a) Video1

(b) Video2

(c) Video3

(d) Video4

Fig. 1. Shots from the set of four videos used to evaluate the proposed method.

Video	mMoG	Intrinsic Model	ViBe	Our Method
Video 1	75.68	35.60	**85.81**	78.56
Video 2	60.93	54.48	65.33	**71.12**
Video 3	71.98	78.10	78.78	**82.15**
Video 4	81.09	**82.98**	62.68	80.21
Average	72.42	62.79	73.15	78.01

Table 2. F-Measure for different methods with the underwater video sequences.

methods previously applied successfully to the same underwater domain [13]: a modified version of MoG (mMoG), Intrinsic Model, ViBe and Sheik's method (for which the code was available online). The results are reported in Table 2. Also on underwater video sequences, our approach outperforms the existing approaches.

The results showed that modeling background and foreground by covariance matrices of color and texture features outperforms the existing approaches and is able to identify objects in extreme conditions that are very unlike to happen in scenes involving people or other object (for instance the algae movements is not comparable to tree movements given the morphology of the algae and the strong marine currents).

4. CONCLUSIONS

Although being tackled for decades now, the problem of identifying moving objects in a video sequence is still open to debate, especially in non-urban environments, where the scene conditions cannot be controlled, the targets' appearance may be harder to distinguish than humans' and is subject to camouflage, and the background presents strong activity and variability. In this work we propose an object detection algorithm which aims at improving the accuracy of this task in the above-mentioned conditions, with a specific reference to the underwater environment. Our method has been devised based on three ideas: firstly, as assessed in the recent literature, the background and the foreground models have to be explicitly separated; this helps to detect a moving object even when it moves over a background with similar visual appearance as the target, since the foreground model would be able to describe it better than the background one. Secondly, there are cases when colour features alone do not allow to fully discriminate between background and foreground pixels; for this reason, texture features have been integrated. Given the intrinsic differences between these two kinds of features, a covariance-based model has been adopted to merge the two types of information, providing as a collateral effect a representation which incorporates a structural description of a pixel's neighbourhood. Thirdly, in order to enforce further the dependency between pixels in the same region, a joint domain-range model based on Kernel Density Estimation and a post-processing MAP-MRF framework have been used. The results we presented show that the union of the above-mentioned techniques yields excellent results in a harsh and problematic environment as the submarine one.

5. REFERENCES

[1] F Cupillard, A Avanzi, F Bremond, and M Thonnat, "Video understanding for metro surveillance," in *IEEE International Conference on Networking Sensing and Control*. 2004, vol. 1, pp. 186–191, Ieee.

[2] Alberto Faro, Daniela Giordano, and Concetto Spampinato, "Adaptive Background Modeling Integrated With Luminosity Sensors and Occlusion Processing for Reliable Vehicle Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1398–1412, Dec. 2011.

[3] Concetto Spampinato, Simone Palazzo, Bastian Boom, Jacco Ossenbruggen, Isaak Kavasidis, Roberto Di Salvo, Fang-Pang Lin, Daniela Giordano, Lynda Hardman, and Robert Bob Fisher, "Understanding fish behavior during typhoon events in real-life underwater environments," *Multimedia Tools and Applications*, 2012.

[4] Shengcai Liao, Guoying Zhao, Vili Kellokumpu, Matti Pietikainen, and Stan Z Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 0, pp. 1301–1306, 2012.

[5] Fatih Porikli, "Achieving real-time object detection and tracking under extreme conditions," *J. Real-Time Image Processing*, vol. 1, no. 1, pp. 33–40, 2006.

[6] Yaser Sheikh and Mubarak Shah, "Bayesian Object Detection in Dynamic Scenes," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR05*, vol. 1, pp. 74–79, 2005.

[7] T Ojala, M Pietikainen, and D Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," 1994.

[8] M Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[9] Peter Hall and M P Wand, "On the accuracy of binned kernel density estimators," *Journal of Multivariate Analysis*, vol. 56, no. 2, pp. 165–184, 1996.

[10] Liyuan Li, Weimin Huang, Irene Y H Gu, and Qi Tian, "Foreground object detection from videos containing complex background," *Proceedings of the eleventh ACM international conference on Multimedia MULTIMEDIA 03*, vol. 03, pp. 2, 2003.

[11] Manjunath Narayana, Allen Hanson, and Erik Learned-miller, "Background Modeling Using Adaptive Pixelwise Kernel Variances in a Hybrid Feature Space," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2104–2111, 2012.

[12] Isaak Kavasidis, Simone Palazzo, Roberto Di Salvo, Daniela Giordano, and Concetto Spampinato, "A semi-automatic tool for detection and tracking ground truth generation in videos," in *Proceedings of the ACM VIGTA 2012*, New York, NY, USA, 2012, VIGTA '12, pp. 6:1—-6:5, ACM.

[13] Isaak Kavasidis and Simone Palazzo, "Quantitative performance analysis of object detection algorithms on underwater video footage," in *Proceedings of the 1st ACM International Workshop on Multimedia Analysis for Ecological Data, MAED12*, 2012, pp. 57–60.