Water Detection with Segmentation Guided Dynamic Texture Recognition

Conference Paper · December 2012

DOI: 10.1109/ROBIO.2012.6491235

3 authors:

Pedro Santana
ISCTE-Instituto Universitário de Lisboa
75 PUBLICATIONS **567** CITATIONS
SEE PROFILE

Ricardo Mendonça
Institute for the Development of New Technologies
31 PUBLICATIONS **196** CITATIONS
SEE PROFILE

J. Barata
Universidade NOVA de Lisboa
293 PUBLICATIONS **3,009** CITATIONS
SEE PROFILE

Water Detection with Segmentation Guided Dynamic Texture Recognition

Pedro Santana, Ricardo Mendonça and José Barata

***Abstract*— This paper proposes a model for water detection in video sequences, which is a key asset of any robot operating in natural environments. By searching the visual input for the waters typically chaotic dynamic texture, the model is able to filter out the static background and even any dynamic object present in the scene. In this work, the waters signature is defined, mostly, in terms of an entropy measure computed from the optical flow obtained across several frames. To foster the classification of motionless regions in the visual input, usually associated to the far field, a segmentation guided label propagation method is used. The model is experimentally validated on 12 diverse videos, acquired from static and moving cameras.**

I. Introduction

While aquatic robots need to stay on water, terrestrial robots should avoid it. This makes water detection a crucial perceptual capability of any outdoor robot. The sheer complexity of this detection problem can be surely reduced if a sensor fusion approach is taken. Interesting sensors include laser scanners, thermal cameras, polarised cameras, and underwater sonars. In this paper, cameras in the visible spectrum are considered. These are small, lightweight, inexpensive, and so valuable for any robotic project.

Typically, vision-based water detection methods rely on still images to perform an instantaneous appearance-based analysis. However, this is barely applicable to rippled water, whose appearance changes over time. In this case, appearance dynamics is caused by the non-linear interactions occurring between turbulence in water's surface and appearance of dissolved sediments, substrate bottom, and reflected structures (e.g., sky and trees). The net effect of these complex interactions is reflected in the chaotic dynamic texture that is often observed in rippled water bodies. Bearing this in mind, this paper proposes the entropy of the optical flow computed across multiple frames as a proper metric for the detection of rippled water.

Smooth water surfaces, i.e., motionless, do not exhibit a detectable dynamic texture. This motionless can be real or apparent. The latter is mostly due to the lack of image resolution to properly compute the optical flow in the far field. To overcome this limitation, the model proposed in this paper includes an appearance-based segmentation guided label propagation stage. In short, this process labels entire image segments as water if these encompass a given percentage of pixels already classified as water by the entropy-based method. This way, a more comprehensive water detection is achieved. Fig. 1 illustrates a typical result obtained with the proposed model.

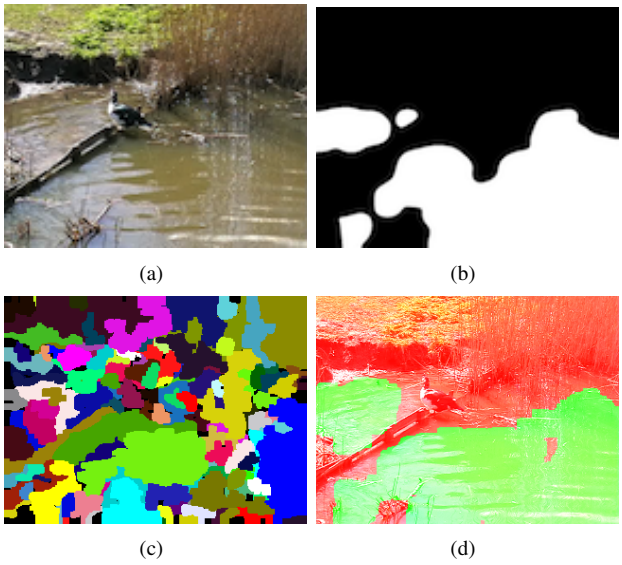


Fig. 1. Key steps of the proposed model in a typical video sequence. (a) First image of the video sequence. (b) The first step of the method refers to the extraction of the visual input's regions that are likely to be water given its dynamic texture signature (white mask). (c) In a subsequent step, an appearance-based segmentation of the video sequence's first image is computed. (d) Then, this segmentation is used to expand the detection results across the image (green overlay). Note that, although the tall grass on the top-left of the video sequence is swaying and the duck is moving, false positives are not generated.

This paper is organised as follows. Section II surveys related work. Section III, presents the optic-flow water detector model, which is followed by the description of the water segmentation process in Section IV. Subsequently, experimental results are presented in Section V. Finally, conclusions and future work directions are provided in Section VI.

II. Related Work

Although laser scanners can be used to detect water bodies by checking the no-return situations [1], [2], [3], the water's appearance is key to cope with the situations in which the beam's incidence angle is not oblique with respect to the water surface [4]. Furthermore, laser scanners are known to produce low quality water/land discrimination in the far field [2], [5]. Hence, an autonomous vehicle equipped with a

P. Santana is with CTS-UNINOVA, New University of Lisbon, Portugal, and also with Instituto Universitário de Lisboa (ISCTE-IUL), Departamento de Ciências e Tecnologias da Informação, Av. das Forças Armadas, 1649-026 Lisboa, Portugal, pfs@uninova.pt

R. Mendonça is with CTS-UNINOVA, New University of Lisbon, Portugal, r.mendonca@campus.fct.unl.pt

J. Barata is with CTS-UNINOVA, New University of Lisbon, Portugal, jab@uninova.pt

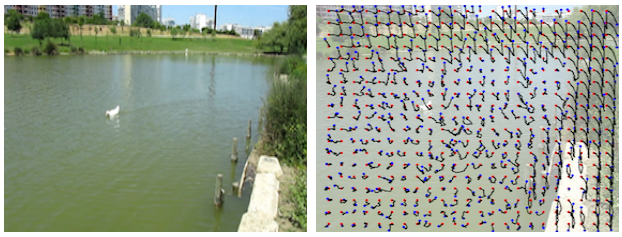
vision-based water detector has higher chances of navigating safely and efficiently. Stereo-vision has been also validated in this domain [6], [7], [5]. The main idea behind the use of stereo is that, reflections on the water are 3-D reconstructed as if they were below the water's surface. However, this approach is limited to processing the near-field, it depends on the presence of reflecting objects as well as on ripples on the water's surface.

In alternative to stereo-based water detection, monocular-based approaches allow, in principle, to segment the water from the land and obstacles across the entire sensor's field of view. The shoreline is an example of high level features that can be extracted from monocular images using symmetry operators [8], [9]. A drawback of using symmetry operators for water detection is that their accuracy depends on the level of reflections displayed on the water's surface.

Appearance-based approaches have also been proposed for the detection of water bodies. For instance, settled water's considerable brightness and low texture can be used as cues supporting its detection [7]. A more complex set of perceptual features can be employed by learning image classifiers from off-line prepared datasets [2], [9], [10]. However, the water's appearance being a function of the water body's floor, reflections imposed by foliage, sky, and other structures, shadows, and ripples in the water's surface, off-line learned classifiers rapidly lose their accuracy. As an open-area water body is approached, the sensed hue gradually changes from the hue of the sky to the hue coming out of the water [11]. Although interesting, the perception of this cue demands for the robot to approach the water body in order to detect it. Sky reflections can also be used directly [12], provided that the water body is settled.

With long range water segmentation in mind, the use of self-supervised learning of water detection classifiers has been recently proposed [13], [14]. The idea is to use geometrical a priori knowledge of the environment's 3-D structure [13] or another sensor, such as an underwater sonar [14], for automatic water/non-water labelling of portions of the visual input and, thus, enabling online generation of training sets of features. Symmetry information can be used to improve the labelling process by reducing the effects of background reflections on the water [15]. Although on-line learning of the water's appearance has a considerable potential, it is always limited by the diversity and quality of the robot's previous experiences in order to properly segment water from land.

From this survey it follows that current detection methods have fundamental limitations in handling rippled water bodies. This is particularly emphasised when the robot needs to face an unlearned environment. To fill this gap, this paper proposes to detect rippled water bodies from their chaotic dynamic texture. In a related approach [16], non-rigid motions detected on the visual input are considered as cues of water presence. However, neither a formal definition or an exhaustive experimental validation is provided. Conversely, the model herein proposed is full formalised, a segmentation guided label propagation included, and extensive validation on a known data set is provided.



(a) (b)

Fig. 2. Trackers on a typical video sequence. (a) Frame on which trackers were deployed. (b) Trajectories performed by the trackers across n frames. Red dots represent trackers' initial positions, whereas the blue dots represent trackers' final positions. Off water non-linear trajectories are a result of severe camera rotation. Nevertheless, they are not as complex as the trajectories of the trackers deployed on water nor as unpredictable. See for instance that the trajectories of neighbour trackers on water are much more different from each other than the trajectories of neighbour trackers off water.

Dynamic textures are attracting interest from the computer vision community [17]. Auto-regressive linear models [18], [19] are amongst the most studied approaches to the problem. Although demonstrated successful in many dynamic texture sequences, they fail when the time-invariance assumption does not hold, as it is the case when cameras are moving. A more robust alternative is to use of optical flow as motion descriptor [20], [21], [22]. In general, these approaches aim at producing a segmentation of the dynamic textures present in the input video, whatever their nature. For classification purposes, learning is often required. Rather than providing a general purpose segmentation and classification method for dynamic textures, this paper provides an explicit formalism for the rippled water case, based on an entropy measure. This way the method avoids the cost of learning and provides a basis for further formal water characterisation developments.

III. Water Detection

This section proposes a method for detecting chaotic local motion as a cue for the presence of rippled water. The rationale is that the presence of ripples change the light incidence angle and, as a result, the water's appearance in an unpredictable way. A visual tracker deployed on such a surface will exhibit a chaotic trajectory across frames. Conversely, even with a moving camera, visual trackers deployed on land, solid moving objects, and swaying grass will not exhibit a chaotic motion. This assumption holds for most natural environments, in which most of the off-water motion is due to the presence of wind. Hence, the method needs to discriminate between complex and simple trackers' motion to properly detect water. Fig. 2 shows typical trajectories exhibited by trackers deployed on image regions with and without water.

The following sub-sections present three optical flow cues for water detection, as well as a procedure for their integration into a final utility map.

A. Cue 1: Optical Flow's Entropy

One tracker is deployed at each vertex of a virtual $m_w \times m_h$ regular lattice covering the image plane. In our experiments,

$(m_w \times m_h) = (23 \times 18)$ provided a good speed-accuracy trade-off. The outer vertexes are not considered in this process, i.e., $|\Theta| = (m_w - 2) \cdot (m_h - 2)$, where Θ is the set of deployed trackers. At each frame, a small black cross with white border is overlaid on the input image at the tracker's initial position. The goal is to create a fake corner that hinders trackers of drifting away when deployed on an homogeneous surface (e.g., the white hull of a boat). Conversely, the presence of a ripple will be stronger than the fake corner and, thus, compel the tracker to follow it.

A pyramidal implementation of the Lucas-Kanade feature tracker [23] is used to displace the trackers across the subsequent n frames. In our experiments, $n = 40$ was sufficient to allow the extraction of sufficient information from the optical flow in order to discriminate between water a non-water elements. We have tested the proposed model with different feature tracker's search windows, namely, (5×5), (7×7), and (11×11) pixels. The (11×11) option was the one producing the best results in our experiments.

To account for the whole executed trajectory, a given tracker $\mathbf{p}$ is defined as a vector of n positions relative to the trackers initial position in the image plane:

$$\mathbf{p} \equiv (\mathbf{x_0}, \ldots, \mathbf{x_n}), \tag{1}$$

where $\mathbf{x_i}$ is a 2-D vector specifying a position in tracker's initial position frame of reference. Note that, depending on the dynamics of the camera and water, the tracker may produce a linear translation of several pixels between subsequent frames, i.e., typically $||\mathbf{x_i} - \mathbf{x_{i-1}}|| > 1$. Therefore, the vector $\mathbf{p}$ specifies a trajectory as a piecewise linear curve.

The next step in the processing pipeline is to determine the complexity of the tracker's trajectory, i.e., the likelihood that it has been modulated by a chaotic process. For this purpose, the entropy of the piecewise linear curve described by the tracker is computed in the following way:

$$H(\mathbf{p}) = \frac{\log\left(\frac{L(\mathbf{p})}{d(\mathbf{p})}\right)}{\log(n-1)} \cdot d_\Theta(\mathbf{p}), \tag{2}$$

where: $d(\mathbf{p})$ is the diameter of the minimum circle encompassing the trajectory; $d_\Theta(\mathbf{p})$ is a scaled version of $d(\mathbf{p})$ so that the range of $d_\Theta(\cdot)$ values across the set of trackers, $\{d_\Theta(\mathbf{x}), \forall \mathbf{x} \in \Theta\}$, is [0,1][1]; and $L(\mathbf{p})$ is defined as the length of the tracker's trajectory:

$$L(\mathbf{p}) = \sum_{i=0}^{n-1} ||\mathbf{x_i} - \mathbf{x_{i+1}}||. \tag{3}$$

This entropy measure describes the regularity of a line. That is, if all points are aligned sequentially along a straight line, then the segment has entropy zero. Conversely, more tortuous lines have higher entropies. Differently from its original formulation [24], which does not account for the $d_\Theta(\cdot)$ function, this entropy measure penalises small trajectories, which are unlikely to be present in rippled water.

[1]With this normalisation procedure, we avoid the specification of free parameters. The underlying assumption is that the visual input covers both water and non-water regions.

B. Cue 2: Inter-Trackers Dissimilarity

So far, we have seen how water's dynamic texture can be detected by means of analysing the entropy of the trajectories executed by a set of trackers deployed on the video sequence. However, trackers' trajectories can be influenced in complex ways by camera attitude changes, thus requiring an additional mechanism for camera rotation invariance. This challenge can be considerably mitigated by making the assumption that the environment is piecewise planar and, thus, that rotation-induced influences are felt likewise in neighbour trackers. Hence, rotation-induced complex trajectories, which could be mistakenly associated to the presence of water, can be filtered out by checking the dissimilarity among the trajectories exhibited by trackers and the ones of their neighbours. The chaotic nature of the water's dynamic texture ensures considerable differences among neighbour trackers, even in the presence of complex camera attitude changes (see Fig. 2).

Formally, dissimilarity between a given tracker $\mathbf{p}$ and a given tracker $\mathbf{q}$ is defined by their Euclidean metric:

$$d(\mathbf{p}, \mathbf{q}) = \sum_{i=0}^{n} ||\mathbf{x_i} - \mathbf{y_i}||, \tag{4}$$

where $\mathbf{y_i}$ represents a 2-D position in the trajectory executed by $\mathbf{q}$:

$$\mathbf{q} \equiv (\mathbf{y_0}, \ldots, \mathbf{y_n}). \tag{5}$$

Finally, Eq. 4 can be used to compute the dissimilarity between the trajectory of a given tracker $\mathbf{p}$ and the trajectories of its neighbour trackers:

$$D^\star(\mathbf{p}) = \frac{1}{|\Phi(\mathbf{p})|} \sum_{\forall \mathbf{q} \in \Phi(\mathbf{p})} d(\mathbf{p}, \mathbf{q}), \tag{6}$$

where $\Phi(\mathbf{p})$ represents the set of trackers composing the neighbourhood of $\mathbf{p}$. Here we take this neighbourhood as the four immediate neighbours. To remove outliers, i.e., isolated trackers exhibiting a complex trajectory, a minimum-dissimilarity post-processing operation is run:

$$D(\mathbf{p}) = \min_{\forall \mathbf{q} \in \{\Phi(\mathbf{p}), \mathbf{p}\}} \{D^\star(\mathbf{q})\}. \tag{7}$$

C. Cue 3: Trackers' Initial Conditions Sensitivity

To further reduce the chances of tracking a dynamic object rather than a chaotic dynamic texture, the tracker's sensitivity to small variations in the initial conditions is verified. Such a sensitivity is a fingerprint of chaotic processes. Here, this sensitivity is tested by deploying a second set of trackers a frame later than the frame in which the original trackers were deployed. The trajectories of both sets are then compared. The one-frame difference enforces small differences to the initial conditions that should be observed as large discrepancies in the resulting trajectories:

$$T^\star(\mathbf{p}) = d(\mathbf{p}, \mathbf{p^{+1}}), \tag{8}$$

where $\mathbf{p}^{+1}$ refers to a tracker deployed one frame later than the frame in which $\mathbf{p}$ was deployed. As before, a minimum-dissimilarity post-processing operation is carried out:

$$T(\mathbf{p}) = \min_{\forall \mathbf{q} \in \{\Phi(\mathbf{p}), \mathbf{p}\}} \{T^{\star}(\mathbf{q})\}. \quad (9)$$

D. Fusing Water Detection Cues

To finally determine whether $\mathbf{p}$ is tracking a water body, the three cues $H(\mathbf{p})$, $T(\mathbf{p})$, and $D(\mathbf{p})$ are first normalised to avoid any bias and, then, are fused as follows:

$$F(\mathbf{p}) \quad = \quad H_{\Theta}(\mathbf{p}) \cdot \min\big(D_{\Theta}(\mathbf{p}), T_{\Theta}(\mathbf{p})\big), \quad (10)$$

where $H_{\Theta}(\mathbf{p})$, $D_{\Theta}(\mathbf{p})$, and $T_{\Theta}(\mathbf{p})$ are the normalised versions of $H(\mathbf{p})$, $D(\mathbf{p})$, and $T(\mathbf{p})$, respectively. The normalisation procedure follows the logic used to normalise d_{Θ} (see above).

The fusion function is then used to create a $m_w \times m_h$ utility map. This map is then smoothed with a 11×11 Gaussian filter and resized to match the video's resolution.

We have compared fusion function $F(\mathbf{p})$ against two other more obvious ones. In the first compared function, $F_2(\mathbf{p}) = \big(H_{\Theta}(\mathbf{p}) + D_{\Theta}(\mathbf{p}) + T_{\Theta}(\mathbf{p})\big)/3$, one of the cues is sufficient to raise the confidence on the presence of water, whereas in the second compared function, $F_3(\mathbf{p}) = H_{\Theta}(\mathbf{p}) \cdot D_{\Theta}(\mathbf{p}) \cdot T_{\Theta}(\mathbf{p})$, a single cue is able to lower the confidence to zero. Conversely, $F(\mathbf{p})$ exploits the entropy cue as primary detection source. In our experiments, $F(\mathbf{p})$ reported systematically better results than $F_2(\mathbf{p})$ and $F_3(\mathbf{p})$.

Finally, to determine which pixels are labelled as water, the utility map is thresholded.

IV. Segmentation Guided Label Propagation

This section describes the image segmentation procedure used to expand the water detection results to motionless, often apparent, regions of the input image. Apparent motionless water regions are typically found in the far field, mostly due to limited camera resolution.

A. Image Segmentation

In this paper we follow a clustering-based segmentation method, based on the work of Blas et al. [25]. This method basically stands on a two-staged unsupervised learning process. In the first stage, the method clusters appearance-based descriptors of all pixels present in the input image. The clustering process is implemented with the K-means algorithm. In our experiments, $k = 32$ provided the best results. These clusters, also called basis vectors, embody the most representative textures present in the input image. Then, every pixel in the input image is assigned to the closest basis vector in Euclidian terms (see Fig 3(b)).

Motivated by the good results obtained by Achar et al. [13] on modelling the appearance of water, the appearance descriptor vector herein considered encompasses texture information encoded in 5×5 Laws' masks [26]. In opposition to the work of Achar et al, in which the Lab colour space was used to complete the appearance descriptor vector, here the $c_1c_2c_3$ colour space was selected. This option follows

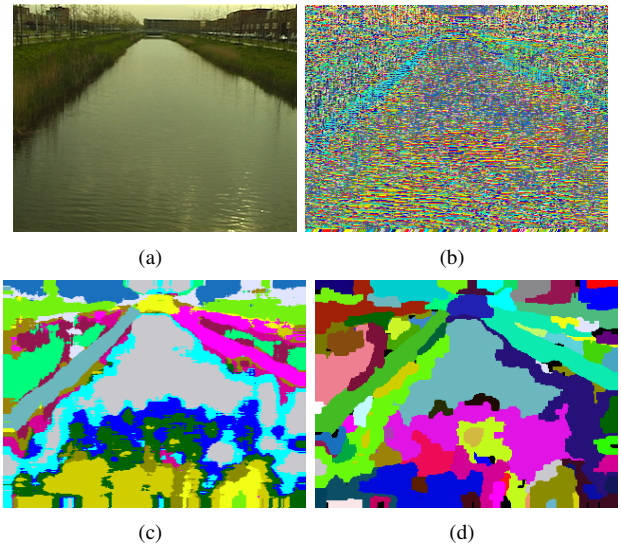


Fig. 3. Stages of the segmentation process over a typical input image. (a) Input image. (b). Stage 1 results. (c) Stage 2 results. (d) Segments aggregation.

the well-known shadow and illumination invariance of the $c_1c_2c_3$ colour space in outdoor environments [27], [28].

In the second stage, statistics of larger areas are learned. Concretely, the histograms of basis vectors over a window of 15×15 pixels are computed. Then, these histograms are clustered together with K-means. In our experiments, $k = 16$ provided the best results. This way, local contextual information is used to bring together similar regions. As for the first, the second stage finishes by re-classifying the input image using the computed set of clusters (see Fig 3(c)).

In a final step, a segmentation is computed by determining the connected components of the previously re-classified image. The resulting segmentation is cleaned from spurious segments using an area-based filter. Then, adjacent segments of similar appearance are aggregated (see Fig 3(d)). By applying this aggregation rule in a sequential way across segments, i.e., from top to down and left to right, a single pass is usually sufficient to attain convergence. Similarity is empirically defined by a Bhattacharyya distance between the histograms representing the segments smaller than 0.1. In our experiments, this threshold enabled sufficient aggregation while respecting the segmentation imposed by the water boundaries.

B. Label Propagation

Similar in spirit with the idea proposed by Ghosh and Mulligan for ground vehicles [29], the pixels labelled as water are used as seeds for a label propagation process. However, in opposition to ours, both positive and negative detection results were used by Ghosh and Mulligan.

Our method proceeds by re-labelling as water all pixels encompassed by segments that have, at least, half of their pixels already labelled as water. In our experiments, this ratio provided good results in terms of cancelling out the effects of optical flow false positives, which would otherwise

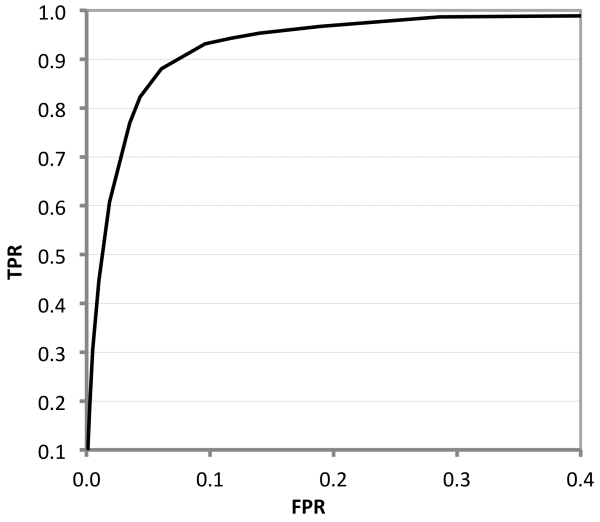


Fig. 4. Proposed model's ROC curve. The plot is the average of the ROC curves built over all video sequences obtained from the data set. ROC curves are obtained by varying the threshold applied to the utility map. The resulting binary (water/non-water) map feeds the label propagation process, which is then used to compute both TPR and FPR.

be propagated throughout the segments. As can be depicted in Fig 1(d), this simple label propagation procedure helps reducing the number of false negatives and, as a result, improve the water segmentation process.

V. Experimental Results

To validate the proposed model, a set of 12 videos from the DynTex data set [30] was prepared. The selection criteria were: (1) both water and land surfaces are simultaneously present throughout the video sequences; (2) the environments are diverse; and (3) videos acquired with both static and moving cameras are included. Videos were acquired at 25 frames per second, encompassing a total of 7757 frames. For validation purposes, ground-truth was generated for every 25 frames. That is, at every 25 frames, a new video sequence of $n = 40$ frames is analysed. Optical flow was computed over a 720×576 image resolution, whereas the label propagation process was run over a down sampled version of the videos, namely 320×240. OpenCV [31] was used for low-level image processing routines.

Fig. 4 plots the Receiver Operating Characteristic (ROC) curve of the proposed method. The curve shows that a very high True Positive Rate (TPR) is enabled without paying a high cost in terms of False Positive Rate (FPR). For instance, 0.95 of TPR is attained with only 0.1 of FPR.

Fig. 5 depicts the model's output in key frames of the tested data set, which encompasses highly diverse environments, lighting conditions, and scales. Despite all this diversity, the model shows to be able to properly segment rippled water bodies from the background. These results confirm that the presence of chaotic optical flow in the robot's visual field is a strong indicator of the presence of a rippled water body. As a result, the proposed model

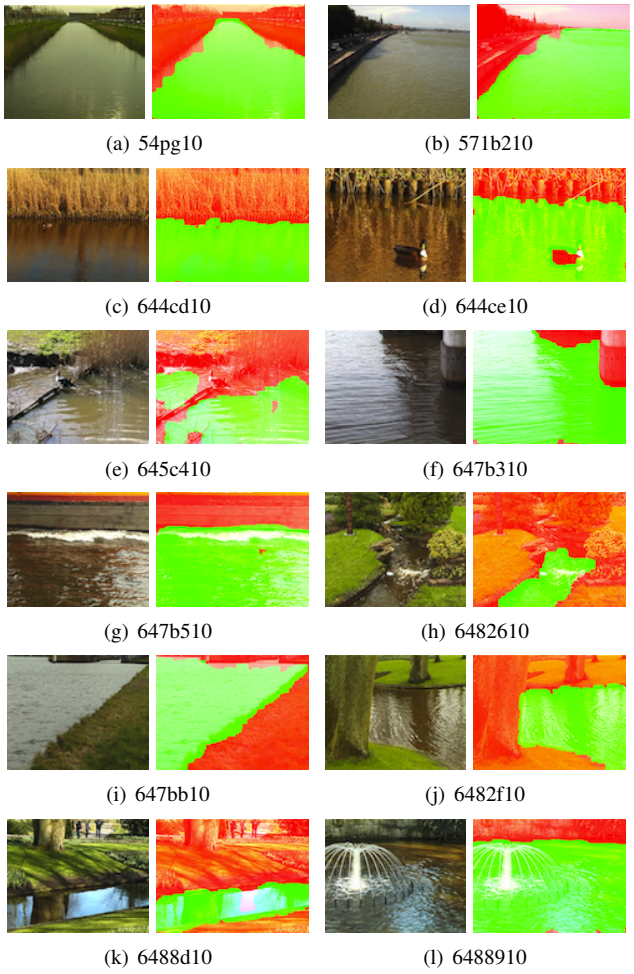


Fig. 5. Proposed model's output (green overlay) on key frames of each of the tested videos. Captions refer to the DynTex's identifier of each video.

overcomes key limitations found in previous work, which is mostly limited to cope with settled water bodies or heavily depends on expensive learning procedures (see Section II).

It is worth noting that the presence of camera motion does not hinder the model's from producing a correct output (see Fig. 5(i)-(l)). This is an important result as it is not always possible to properly stabilise the video feed on a mobile robot. Being able to cope with moving objects is also key. This is demonstrated by the system's ability to filter out: a boat crossing at high speed the camera's field of view from the right to the left (Fig. 5(g)); a moving duck (Fig. 5(c)-(e)); and swaying tall grass (Fig. 5(e)).

The model's accuracy owes to the synergistic operation occurring between the water labelling process based on dynamic texture recognition and the segmentation-based label propagation process. The latter allows the system to propagate labels to the far field, where the camera's limited image resolution disables a proper dynamic texture analysis. Furthermore, by filtering out segments with a low number of water labelled pixels, the rate of false positives is diminished.

VI. Conclusions

A model for water detection based on its dynamic texture was proposed. The goal is to enable safe robot navigation in natural environments. The method relies on the chaotic nature of water's dynamic texture to exploit a measure of entropy over the trajectories obtained from optical flow trackers over several frames. To further improve the results, in particular when the water is locally insufficiently rippled, a label propagation method was proposed. Overall, the method allows detecting water regions despite their instantaneous appearance, which can be a rather unpredictable cue. As future work, we expect to exploit probabilistic models in alternative to the deterministic function used to blend the several water presence cues. In the same line, for improved accuracy and robustness, we also expect to include a spatio-temporal smoothing (tracking) mechanism by means of probabilistic graphical models. Finally, we also envision the use of trackers stabilisation techniques, aided by inertial systems [32], to reduce the optical flow induced by camera motion.

Acknowledgments

This work was developed within the scope of the RIVERWATCH experiment from the European FP7 Large Scale Integrating Project ECHORD and it was also partially supported by CTS multi-annual funding, through the PIDDAC Program funds.

References

[1] T. Hong, S. Legowik, M. Nashman, N. I. of Standards, and T. (US), "Obstacle detection and mapping system," US Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, Tech. Rep., 1998.

[2] L. Matthies, P. Belluta, and M. McHenry, "Detecting water hazards for autonomous off-road navigation," in *SPIE proceedings series*. Society of Photo-Optical Instrumentation Engineers, 2003, pp. 231–242.

[3] L. Elkins, D. Sellers, and W. Monach, "The autonomous maritime navigation (amn) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles," *Journal of Field Robotics*, vol. 27, no. 6, pp. 790–818, 2010.

[4] T. H. Hong, C. Rasmussen, T. Chang, and M. Shneier, "Fusing ladar and color image information for mobile robot feature detection and tracking," in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, Marina Del Ray, CA, March 2002, pp. 1–6.

[5] T. Huntsberger, H. Aghazarian, A. Howard, and D. Trotz, "Stereo vision–based navigation for autonomous surface vessels," *Journal of Field Robotics*, vol. 28, no. 1, pp. 3–18, 2011.

[6] J. Larson, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," DTIC Document, Tech. Rep., 2007.

[7] T. Yao, Z. Xiang, J. Liu, and D. Xu, "Multi-feature fusion based outdoor water hazards detection," in *Proc. of the International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2007, pp. 652–656.

[8] A. Subramanian, X. Gong, J. Riggins, D. Stilwell, and C. Wyatt, "Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications," in *Proc. of the IEEE OCEANS Conference*. IEEE, 2006, pp. 1–6.

[9] X. Gong, A. Subramanian, and C. Wyatt, "A two-stage algorithm for shoreline detection," in *Proc. of the IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2007, pp. 40–46.

[10] A. Xu and G. Dudek, "A vision-based boundary following framework for aerial vehicles," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 81–86.

[11] A. Rankin and L. Matthies, "Daytime water detection based on color variation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 215–221.

[12] A. Rankin, L. Matthies, and P. Bellutta, "Daytime water detection based on sky reflections," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 5329–5336.

[13] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh, "Self-supervised segmentation of river scenes," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 6227–6232.

[14] H. Heidarsson and G. Sukhatme, "Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 3160–3165.

[15] A. Chambers, S. Achar, S. Nuske, J. Rehder, B. Kitt, L. Chamberlain, J. Haines, S. Scherer, and S. Singh, "Perception for a river mapping robot," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 227–234.

[16] F. Snyder, D. Morris, P. Haley, R. Collins, and A. Okerholm, "Autonomous river navigation," *Proceedings of SPIE, Mobile Robots XVII*, pp. 221–232, 2004.

[17] D. Chetverikov and R. Péteri, "A brief survey of dynamic texture description and recognition," *Proc. of the Intl. Conf. on Computer Recognition Systems*, pp. 17–26, 2005.

[18] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2001, pp. II–58.

[19] A. Chan and N. Vasconcelos, "Layered dynamic textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1862–1879, 2009.

[20] R. Vidal and A. Ravichandran, "Optical flow estimation and segmentation of multiple moving dynamic textures," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 516–521.

[21] S. Fazekas, T. Amiaz, D. Chetverikov, and N. Kiryati, "Dynamic texture detection based on motion analysis," *International journal of computer vision*, vol. 82, no. 1, pp. 48–63, 2009.

[22] V. Traver, M. Mirmehdi, X. Xie, and R. Montoliu, "Fast dynamic texture detection," *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 680–693, 2010.

[23] J. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker description of the algorithm," *Intel Corporation, Microprocessor Research Labs, OpenCV Documents*, 1999.

[24] A. Balestrino, A. Caiti, E. Crisostomi, and G. Grioli, "A generalised entropy of curves: an approach to the analysis of dynamical systems," in *47th IEEE Conference on Decision and Control (CDC)*. IEEE, 2008, pp. 1157–1162.

[25] M. Blas, M. Agrawal, K. Konolige, and A. Sundaresan, "Fast color/texture segmentation for outdoor robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, Piscataway, 2008, pp. 4078–4085.

[26] K. Laws, "Rapid texture identification," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 238, 1980, pp. 376–380.

[27] T. Gevers and A. Smeulders, "Color-based object recognition," *Pattern Recognition*, no. 32, pp. 453–464, 1999.

[28] D. Song, H. Lee, J. Yi, and A. Levandowski, "Vision-based motion planning for an autonomous motorcycle on ill-structured roads," *Autonomous Robots*, vol. 23, no. 3, pp. 197–212, 2007.

[29] S. Ghosh and J. Mulligan, "A segmentation guided label propagation scheme for autonomous navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 895 –902.

[30] R. Péteri, S. Fazekas, and M. Huiskes, "Dyntex: A comprehensive database of dynamic textures," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1627–1632, 2010.

[31] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., Sebastopol, CA, 2008.

[32] M. Hwangbo, J. Kim, and T. Kanade, "Inertial-aided klt feature tracking for a moving camera," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2009, pp. 1909–1916.