Visual Diver Detection using Multi-Descriptor Nearest-Class-Mean Random Forests in the Context of Underwater Human Robot Interaction (HRI)

Arturo Gomez Chavez, Max Pfingsthorn, Andreas Birk
Jacobs University Bremen
Robotics, Computer Science and Electrical Engineering
Campus Ring 1, 28759 Bremen, Germany
a.birk@jacobs-university.de

Ivor Rendulić and Nikola Mišković
University of Zagreb
LABUST - Lab. for Underwater Systems and Technologies
Unska 3, 10000 Zagreb, Croatia
nikola.miskovic@fer.hr

***Abstract*—This paper introduces a visual method for diver detection in the context of Human Robot Interaction (HRI). The detection is treated as a classification problem, where a discriminative model is trained by computing image features of the target (diver) and underwater scenery. This type of scenery poses great challenges due to its high variability, as it often presents high illumination changes, scarce features and image distortions. For this reason, it is desirable to represent this type of images with multiple type of complementary features. The system scalability is, however, lowered as the number of features types increase as the amount of data to represent queries and indexes also increases.**

To remedy this, we modified the *Nearest Class Mean Forests (NCMF)* method, a variant of *Random Forests*, to integrate as many features types as desired without concerning about scalability and performance decay. The system outperforms the common generative tracking methods which fail to encompass different type of distortions into one model and ignore background information. And in contrast to tracking methods using acoustic sensors which output a single value (distance to the diver), our approach outputs a region encompassing the diver's body; information that can be further exploited to enhance underwater HRI. Not to mention that camera setups offer higher flexibility in size and energy consumption constraints than acoustic sensors. All of the system's aforementioned capabilities are tested with real-life data obtained from field experiments.

I. Introduction

Diver detection has a rich history using acoustic sensors [1], [2], [3], [4], with several commercial models available for civil and military applications. However, the requirements for diver detection in the field of underwater human robot interaction (HRI) between a diver and an autonomous underwater vehicle (AUV) are significantly different.

First, size limitations of an AUV engaging in human robot interaction underwater also limit the size of the sensing setup to be used for diver detection and tracking. Additionally, energy limitations of the AUV batteries as well as safety limits for acoustic energy have significant impact on a feasible acoustic setup. On the other hand, large range is not required for underwater HRI, as the AUV would not interact with a diver far away, rather the diver or divers in its immediate surrounding. Similarly, a guaranteed coverage is not required, since the diver could actively seek the field of view of the AUV.

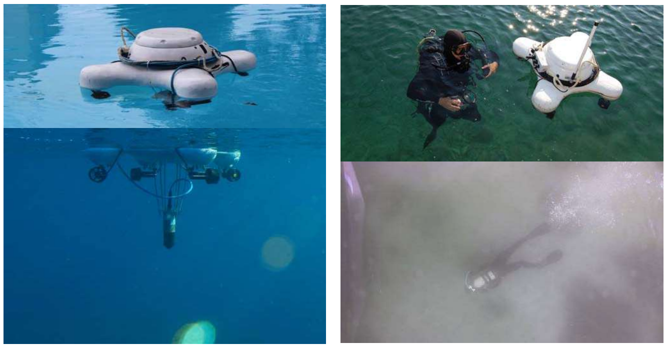

Fig. 1. Left: The Pladypos surface platform with mounted camera and USBL sensor. Right: Deployment of the system for diver tracking and image captured by the mounted camera. Note the low contrasts and reflections.

With this perspective, visual methods are preferable to acoustic ones. Previous research in the field of visual detection of humans, animals and other objects in water has involved public pool monitoring [5], measuring animal populations [6], salient animal motion detection to ease expert evaluation [7], AUV navigation [8], as well as threat detection using active cameras [9].

Only some work has addressed tracking a diver in relation to an autonomous vehicle for the purpose of performing HRI tasks. Sattar and Dudek [10] present a system to track stereotypical diver swimming motion, which has obvious drawbacks when the diver is holding position to start communication with the AUV. Wang *et al.* [11] describe a method for sign language recognition, assuming that the diver has been successfully localized. Our own previous work explored the ideas of motion segmentation as well as using these motions for communication between the diver and the AUV [12], [13], again relying on a steady video feed.

In this paper, a method based on randomized trees is proposed to cope with the problem of diver detection and subsequent tracking. The aim is to develop a robust system that can work with underwater imagery, which due to light backscatter in water, presents non-uniform illumination and image degradation (poor contrast and distortion). A list of these computer vision challenges, both in hardware and software, is presented in [14]. For such conditions, tracking algorithms based on ensemble methods and discriminant classifiers [15] [16] show the best state of the art accuracy and noise-insensitivity. These methods can update online the appearance model of the target object (diver) as image distortions change and they take into consideration background scenery to better locate the object of interest.

To further address these challenges, the main contribution in this paper is a variant of Nearest-Class-Mean Forests (NCMF) [17]. NCMF methods allow dynamic learning as the number of classes increases, without the need to retrain all models from scratch; and at each node, it partitions the sample space by comparing distance between class means instead of comparing values at each feature dimension as in traditional Random Forests [18]. As it is later explain in section II-D, by taking advantage of these properties we can treat each feature-object pair as a new class e.g. SURF-diver, or SIFT-background (non-diver), and examine which centroids best partition the sample set. Hence, this variant of NCMF eliminates the need of encoding the found local descriptors into a higher dimensional codebook [19] [20] [21].

Our main application lies in the EU project *Cognitive Autonomous Diving Buddy (CADDY)* with the objective to allow a better human-robot interaction for the execution of underwater tasks such as the exploration of archaeological sites and terrain mapping. The presented method is tested for single snapshots using a down-looking camera mounted on the *PlaDyPos* surface vehicle (Fig. 1).

II. Multi-Descriptor Nearest Class Mean Forests

To introduce the concept of Nearest Class Mean Forests (NCMFs) for multiple descriptor aggregation, we first explain Nearest Class Mean (NCM) classifier, Random Forests (RFs) and NCMFs. Afterwards, we describe how NCMFs are changed in order to include multiple feature types for each class.

A. Nearest Class Mean classifier

To make use of this classifier, we first compute the mean or class centroid c_k for each class $k \in K$ as follows:

$$c_k = \frac{1}{|I_k|N_k}\sum_{j\in I_k}\sum_{\vec{x}\in X_j}\vec{x} \quad \text{where} \quad N_k = \sum_{j\in I_k}|X_j| \tag{1}$$

In the previous equation, each image I in the dataset is represented by a collection of feature vectors X, where each of these vectors $\vec{x}$ has d dimensionality, $\vec{x} \in \mathbb{R}^d$. Then I_k is the set of images that belong to the class k.

In order to perform Nearest Class Mean (NCM) classification as in [22], we represent a query image Q by a d-dimensional feature vector $\vec{x_Q}$ and search for the closest centroid in the feature space by making $|K|$ comparisons in $\mathbb{R}^d$:

$$k^*(Q) = \arg\min_{k\in K}||\vec{x_Q} - c_k||^2 \tag{2}$$

Equation 2 uses the Euclidean distance to find the nearest class, but any distance definition e.g. Mahalanobis, can replace it.

B. Random Forest

A random forest consists of a collection of T decision trees, each independently trained, in order to reduce the variance of the overall model. To classify a new object from an input vector, the vector is passed down to each of the trees on the forest. Each tree gives a classification (majority voting) or a class probability according to the statistics saved at the leaves l; we say that the tree "votes" for this class. Then the forest chooses the classification having the more votes, or it makes an average of the class probabilities. Based on this, we assume a feature vector of an image $\vec{x}$ was passed down a tree until it arrived at a leaf $l(\vec{x})$, then the final classification is defined as:

$$k^*(\vec{x}) = \arg\max_k \frac{1}{T}\sum_t P^t_{l(\vec{x})}(k), \tag{3}$$

where $P^t_l(k)$ is the distribution over the classes k in a leaf l of a tree t. These distributions are obtained during the training phase of the random forest.

To train each tree in the forest, we start using the complete training set S and, as we go down the tree, each subset of the training data arriving at node n, S^n, is partitioned by a splitting function θ^n. As a result we have two subsets S^n_{left} and S^n_{right}. Commonly, a random set of splitting functions is generated and the best one is selected according to the information gain G, as the following equations describe:

$$G(\theta) = H(S^n) - \sum_{i\in\{left,right\}}\frac{|S^n_i|}{|S^n|}H(S^n_i) \tag{4}$$

$$H(S^n) = -\sum_{k\in K}P(k|S^n)\log_2 P(k|S^n) \tag{5}$$

$$\theta^n = \arg\max_\theta G(\theta) \tag{6}$$

where H denotes the entropy of the class distribution at node n, and $P(k|S^n)$ the fraction of training data belonging to the class k. The left and right nodes are trained with S^n_i accordingly, and the algorithm continues recursively until a stopping criteria is met, commonly a maximum tree depth or a minimum number of samples per leaf is fixed.

C. Nearest Class Mean Forests

NCM forests is a variation of Random Forests, where the splitting functions θ are NCM classifiers presented in section II-A. One difference with NCM classifiers is that in any given node only a fraction of the classes k are used in order to lower the number of comparisons needed. Although this procedure may seem to underfit the data, ensemble methods compensate for this by generating a collection of weak classifiers (decision trees) and then combining their output [18]. Another difference is that Equation 2 has a multiclass output, but in NCMFs Equation 2 will assign each data sample to the left or right child (binary output). Thus, NCMFs implicitly encode a hierarchical structure of the studied classes.

To train NCMFs, first we denote by K^n to the subset of classes observed in the training data S^n, and by S^n_k the subset of S^n belonging to the class k. Then, for each $k \in K^n$ we compute the centroids c_k^n as in Equation 1. Each of the classes $k \in K^n$ and their respective centroids are assigned randomly to the left or right child of the node n. Hence, our splitting function should perform the following mapping

$$\theta^n : k \longmapsto \{left, right\} \;\text{ where }\; k^*(\vec{x}) = \arg\min_{k \in K^n} \|\vec{x} - c_k^n\|^2 \quad (7)$$

In order to select the best splitting function θ^n, we used Equations 4 to 6. As proved in [17], NCMFs offer state-of-the-art accuracy, perform non-linear classification at node level, have no scalability issues as the number of classes increase and can perform incremental learning. The concept of NCMF for image classification is illustrated in Fig. 2.

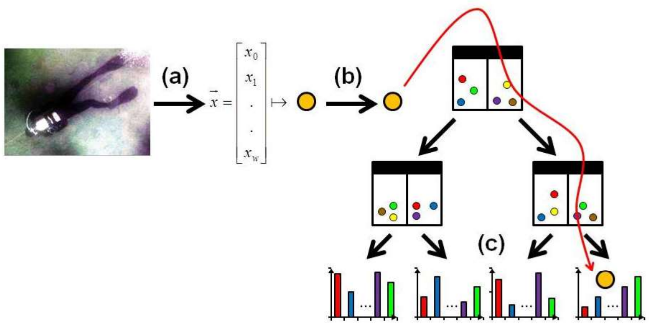
(a)
$\vec{x} = \begin{bmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ x_w \end{bmatrix} \mapsto$
(b)
(c)

Fig. 2. Classification of an image by a NCM tree. (a) The image is encoded into a feature vector $\vec{x}$. (b) The feature vector passes down the tree, following the closest centroid at each level (the followed path is shown with the red arrow and each centroid is represented by a colored dot). (c) When it reaches a leaf, the image is assigned a class distribution.

D. Multi-descriptor aggregation in NCMF

In this section, we propose to do a variation on the standard NCMF methodology in order to aggregate multiple features. Initially we have $|K|$ classes, each one associated with one centroid c_k which is equal to the mean of the feature vectors belonging to k. These feature vectors only represent one type of descriptor, we cannot simply average different type of descriptors because their dimensionality and value ranges are different; not to mention that they represent different types of information.

Now, we have a set of feature types $F = \{f_1, f_2, ..., f_z\}$ and a set of classes $K = \{k_1, k_2, ..., k_w\}$. In this case, to train a node n with incoming data S^n; we picked a random subset of class-descriptor pairs of the form (f_i, k_j) e.g. SURF-diver or SIFT-background, and compute their centroids using Equation 1. Then, at node n we will have a collection of centroids c_p^n where $p \in P^n$, and P^n is the set of all possible class-feature pairs that can be constructed at node n, $P^n = \{(f, k) | \forall f \in F, \forall k \in K^n\}$.

Likewise, each input image is now represented by z different feature vectors, each corresponding to a type of descriptor. Having this in mind, we make the following change to Equation 7 (splitting function):

$$\theta^n : p \longmapsto \{left, right\} \;\text{ where }\; p^*(\vec{x}) = \arg\min_{p \in P^n} \|\vec{x} - c_p^n\|^2 \quad (8)$$

And as before, we optimize θ^n using the equations in section II-B. It is important to notice that when we apply Equation 8, the feature vector $\vec{x}$ has to be of the same type as indicated by $p = (f_i, k_j)$. In this work the distance metric used is the Bhattacharyya distance [23] instead of the Euclidean since it considers the variance in each vector dimension. Algorithm 1 offers an overview of how to grow a tree within this modified version of NCMF.

Algorithm 1 MD-NCMF algorithm *MD-NCMF-Tree(S^n)*

```
input: $S^n$, sample data
persistent: $\mathcal{F}$, set of feature descriptors
            $p_{limit}$, max number of class-descriptor pairs
            $\theta_{limit}$, max number of splitting functions
if stopsplitting($S^n$)=true then
    return createLeafNode($S^n$)
else
    $K^n \leftarrow$ pick classes present in $S^n$
    $P^n \leftarrow \emptyset$
    for i=1 to $p_{limit}$ do
        $p \leftarrow w(K^n, \mathcal{F})$, pick randomly a class feature pair
        if $p \notin P^n$ then
            $P^n \leftarrow P^n \cup p$
        end if
    end for
    $C^n \leftarrow$ computeCentroids($P^n$)
    for j=1 to $\theta_{limit}$ do
        $\theta^n \leftarrow$ createSplitFcn($P^n, C^n$)
        split $S^n$ according to $\theta^n$ (Equation 8)
        for all $\vec{x} \in S^n$ do
            $S^n_{left} \leftarrow \theta^n(k^*(\vec{x}))$ =left
            $S^n_{right} \leftarrow \theta^n(k^*(\vec{x}))$ =right
        end for
        $G \leftarrow$ informationGain($S^n_{left}, S^n_{right}$)
    end for
    select $\theta^n$ that achieve the highest $G$
    return createDecisionNode($\theta^n$,MD-NCMF-Tree($S^n_{left}$),
                    MD-NCMF-Tree($S^n_{right}$))
end if
```

The inputs the user has to define are the type of features F used, the maximum number of class-descriptor pairs p_{limit} to generate and the maximum number of splitting functions θ_{limit} to tests at each node n. Algorithm 1 is only an outline, cases like when the number of class-descriptor pairs p is less that p_{limit} or when these pairs cannot generate θ_{limit} splitting functions have to be considered. The stopping criteria is the same one used for Random Forests, when the number of samples $|S^n|$ at node n is less than a threshold μ, the class distribution is stored at that leaf-node.

III. Object detection by Random Sampling

In section II we introduced a method based on NCMFs to classify input images; but for Human Robot Interaction (HRI) systems, image areas indicating informative features about a person are necessary to further process them for gesture or body language recognition, for example. Thus, a saliency map showing the image region where the diver is located is the expected input for such systems; where high saliency regions are likely to contain parts of the diver's body, while lower saliency regions are associated with background.

To build such saliency map, we follow a similar outline as in [24] with random sampling.

1. Extract a random subwindow w from the query image I, which is represented as a 3D point containing position x, y and size s according to some probability density function *P(M)*.
2. Classify the subwindow using MD-NCMF as in section II.
3. Based on the previous classification, propagate the information about the subwindow to the neighboring areas and compute the new saliency map.
4. Update *P(M)* for all 3D points $X = (x, y, s)$ in order to start the iteration again. From this probability density function PDF, another PDF $\hat{P}(O)$ showing the target's location can be obtained.

In order to guide the sampling such that enough subwindows on the object (diver) are sampled, two PDFs are used. The first one indicates the probability $\hat{P}(O|X)$ of an object being present at the given patch and the second one the probability of already have explored that area $P(E|X)$; and in contrast with $\hat{P}(O|X)$ is not an estimation. These two probability maps are then combined by multiplication and normalization into a single PDF $P(M)$ used to generate the random patches. Patches with high probability of being an object and being in an unexplored region are favored.

To begin with, the whole image needs to be explored and we do not have any knowledge about whether there is an object in the image or not. For this reason, the probability maps are first initialized uniformly such that $\hat{P}_0(O|X) = 0.1$ and $P(E|X) = 1$. Afterwards, as patches are sampled from the image, the next update functions are used:

$$P_t(E|X) = max(0.1, P_{t-1}(E|X) - 0.05 \cdot N(w_t)) \qquad (9)$$

$$\hat{P}_t(O|X) = min(1, max(0.1, \hat{P}_{t-1}(O|X) + 0.05 \cdot Z_t \cdot N(w_t))) \qquad (10)$$

Z_t is the classification given by the MD-NCMF when the patch w_t is the query. $N(w_t)$ is a function that computes a neighboring region of w_t. If the current point X is within that region, the probability maps will be updated for that value of X. This neighboring function can be user-defined to include cubic, spherical or some other type of regions in the neighborhood of X. In the present work cubic regions were chosen. Iterating through these equations sufficient amount of times produces smooth results as shown in the next sections.

IV. Experiments

A. Comparing MD-NCM Forests with ERC-Forests

We used the GRAZ-02 test set [25], available at http://www.emt.tugraz.at/~pinz/data, as a benchmark for our framework before testing it in the underwater scenario. This dataset encompasses three object categories: persons, bicycles and cars; plus a set of negative images (not containing the first three classes). It is considered challenging due to the high intra-class variations, significant amount of background clutter, illumination changes and partial occlusions of the objects as shown in Figure 3. Also the background is highly variable in each of the classes, which makes it difficult to recognize objects based on context.

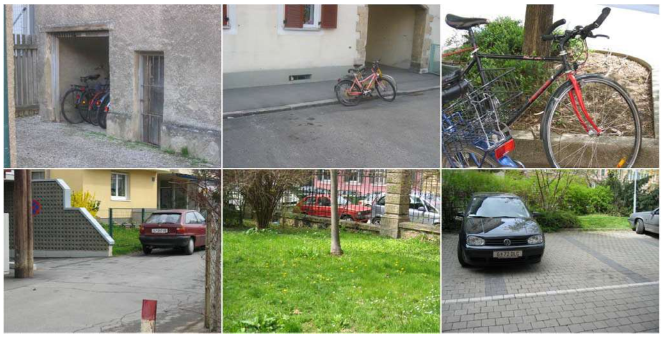

Fig. 3. Samples from the GRAZ-02 database showing the high variation between images of the same class due to clutter, occlusions and different perspectives; bikes are shown in the top row and cars in the bottom row.

We followed the experimental setup as defined by Opelt and Pinz [25]; each of the object classes where tested against the negative category, 150 images of each class were used for training and 150 images for testing. Training was done using whole images, no masks. For each image we applied the following keypoint detectors: Maximally Stable Extremal Region (MSER) [26] and Harris Affine Regions (HAR) [27], which are represented by SIFT [28] and DAISY [29] descriptors respectively. Also a 768-D feature vectors were computed from raw HSL color pixels transformed by a Haar wavelet [30] in 16×16 subwindows as in [24].

The performance is measured with classification rates at equal error rate (EER), Table I shows the mean values over 10 learning runs since the method is randomized. We use five trees in the MD-NCM Forest, at every node we choose $P^n = 5$ out of the $P = 12$ class-descriptor pairs available to generate the splitting functions, and the tree branches stop growing when the number of samples at node n is less or equal than $\mu = 10$.

TABLE I
Classification Rate at EER in GRAZ-02 dataset

	Bike vs Neg	Cars vs Neg	Persons vs Neg
Opelt *et al.* [25]	76.5%	70.7%	81.0%
ERC-Forests [24]	84.4%	79.9%	-
MD-NCM Forests	89.6%	84.2%	92.3%

The features used in this test where chosen because as explained in section IV-B2 they have proven to perform better when used together; this is due to the fact that they offer complementary information about the query image. It is important to mention that in this paper we mostly focused on the use of MD-NCMFs on the diver-localization scenario; but further benchmarking against state of the art techniques and inclusion of higher number of features in our model has to be done to fully determine its accuracy and scalability for a wide range of applications.

B. MD-NCM Forests for underwater diver localization

1) Data gathering: The experiments were conducted in July 2014 in Split, Croatia. A diver conducted several passes on a straight-line transect at average depth of 7 meters and was followed and filmed by an autonomous surface vehicle (ASV).

The ASV used was PlaDyPos (shown in Fig. 1), a vehicle developed and built by University of Zagreb. It is 0.35m high, 0.707m wide and long and weights approximately 25kg. The ”X” configuration of 4 thrusters allow omni-directional motion, i.e. motion in the horizontal plane under any orientation, with velocity up to 0.7m/s.

The systems available on PlaDyPos, apart from compass, batteries and CPUs, include a u-blox Neo-6P GPS module, a Bullet M2 wireless modem for communication with ground station, a Ultra-Short Baseline (USBL) used for determining diver's relative position to the vehicle and a Bosch Flexidome IP Starlight 7000VR mounted in cylindrical waterproof casing. The Bosch camera is shown in Fig. 4. Due to lack of space on the platform, the camera was positioned very close to the water surface and was not even fully submerged at all time. This led to artifacts in the form of visible glow on camera casing, which can also be appreciated in the diver's picture in Fig. 1.

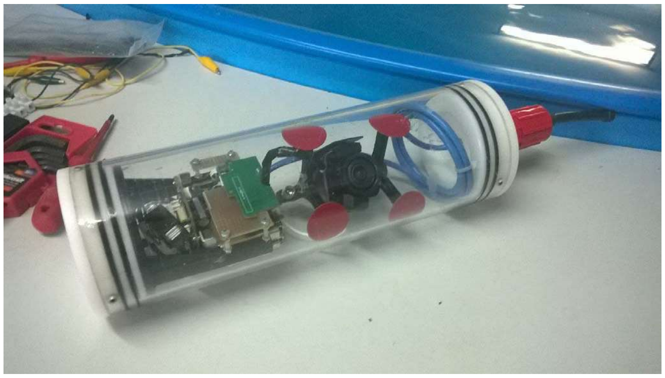

Fig. 4. Bosch Flexidome IP Starlight 7000VR camera mounted in waterproof casing.

At first, the purpose of the conducted experiment was to test the accuracy of tracking algorithms using only acoustic data. Diver was equipped with an acoustic modem and PlaDyPos was tracking him by calculating the relative position with USBL. Video data was used to provide real-time feedback on the diver's location and later serve as a reference for validation of acoustic tracking precision. Since the video data was transmitted to the surface using a limited-bandwidth wireless network, JPEG images with size 768 × 432 pixels were time-stamped and recorded at a rate of approximately 10 Hz. This exact same video feedback was used to built the training and testing dataset for our version of random forests. The training data set consists of 240 images of each category (diver and background) and the testing dataset of 50 each.

2) Feature Selection: For a given image, we can compute different local keypoint or region detectors to detect multiple features; and for each one of them we can apply different type of descriptors. The main objective is to represent the image with multiple types of complementary features, since different images can exhibit different kinds of low-level characteristics according to the view-perspective, occlusions and possible distortions.

In order to see how MD-NCM Forests integrate different type of information about a query (complementary features), we combine several keypoint detectors and descriptors to train our model for the diver dataset collected. Table II shows classification rate results of these combinations, where the combination MSER-SIFT and HAR-DAISY is the best. These two type of features include two different detectors and two different descriptors. Maximally Stable Extremal Regions (MSER) often detect blobs of high contrast and Harris Affine Regions tend to be centered in corner-like features as shown in Figure 5. This indicates that MD-NCMF is effectively exploiting and integrating the complementary characteristics among different features.

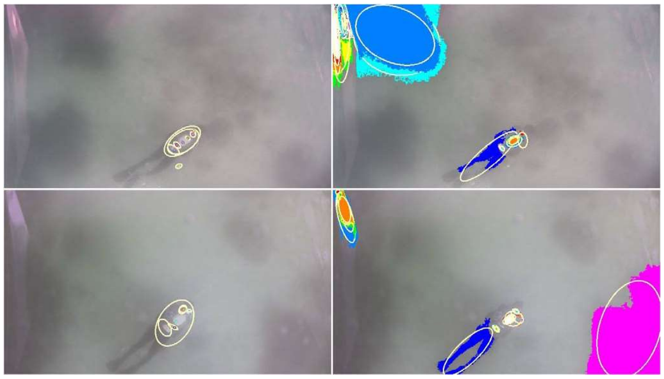

Fig. 5. Left: Most relevant corner-like features detected by Harris-Affine Regions (HAR). Right: High-contrast blobs detected by Maximally Stable Extremal Regions (MSER). The yellow ellipses indicate the transformation invariant region selected by the detector.

TABLE II
CLASSIFICATION RATE AT EER WITH DIFFERENT FEATURE COMBINATIONS IN DIVER DATASET

	LOG-SIFT	LOG-DAISY	MSER-SIFT	MSER-DAISY	HAR-SIFT
LOG-DAISY	76.4%				
MSER-SIFT	80.2%	82.5%			
MSER-DAISY	81.3%	83.1%	79.2%		
HAR-SIFT	81.1%	82.9%	83.4%	83.6%	
HAR-DAISY	82.3%	82.7%	**86.4%**	84.8%	80.5%

C. Diver Localization

In this part of the experiments, we tested how MD-NCMF integrated with patch random-sampling (explained in section III) performs in the diver localization task. The algorithm is stopped after $n = 2000$ samplings and was used with three different models; the first two only use one type of

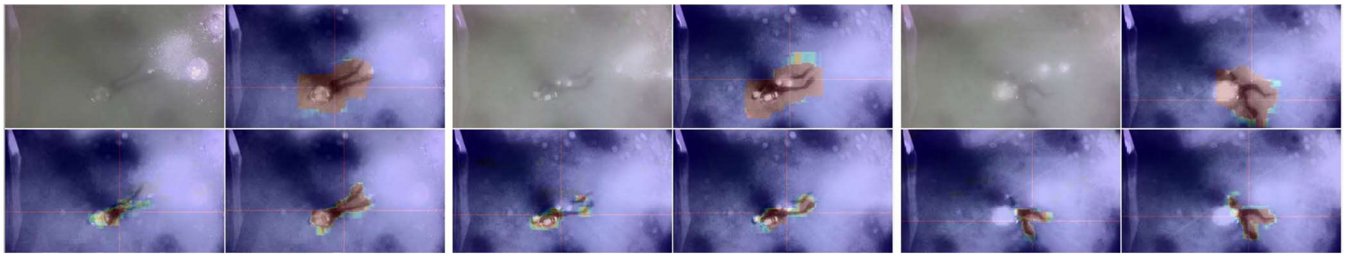

Fig. 6. Sequence of images showing the saliency maps generated with the models from section IV-C. Each set of images is formed by the original diver image (top left), and the saliency map from the MSER-SIFT model (top right), the HAR-DAISY model(bottom left) and the Multiple Feature model (bottom right). Probabilities $\hat{P}(O)$ of a pixel being part of the diver are depicted in color scale, where red means a probability close to 1.0 and dark blue close to 0.

feature and the third uses the combination MSER-SIFT with HAR-DAISY which worked best in our previous experiment. Table III proves again that using more than one type of feature gives better classification rates; however, the ultimate objective is to observe the difference in using multiple features when creating the diver's saliency map as it is the one used for localization. Figure 6 shows some of the diver's images and their respective saliency map.

TABLE III
Classification Rate at EER for diver localization

MSER-SIFT	HAR-DAISY	Combined
78.6%	80.7%	86.4%

The model trained with MSER-SIFT features usually recognizes large blobs as the diver and, when he is occluded, other background elements are included in the detection or the detection fails. The ensemble method based on HAR-DAISY is better at selecting only patches that correspond to the diver, but these are small or scattered through the diver's body. We can appreciate that the model trained with both features offers a balance between the previous models, the detected areas are usually connected and adjust better to the diver's body. Our saliency map also offers confidence areas, as each subregion is related to the probability $\hat{P}(O)$ explained in section III.

In order to quantify our results, in the testing dataset we manually labelled the image's area where the diver is located. We compute the ratio of pixels that were detected by the saliency map to the ones in the manually selected area (true positive rate or TPR). A pixel is counted as being part of the diver if its associated probability is equal or higher than 0.9. In the same manner, we obtain the ratio of pixels detected as being of the diver to those outside the ground-truth areas (false positive rate FPR). All of the previous to know how well the saliency map adjusts to the diver's limbs and body and how accurate it is. Table IV shows these results and proves that the combined model works best.

The experiments were done with a 4th generation 2.6 GHz Intel core i7-4710HQ processor, and the average processing time using n = 2000 samplings for the combined MD-NCMF model (2 features) is 1.43 seconds.

TABLE IV
Average true positive and false positive rates of each MD-NCMF model in the diver dataset

	TPR	FPR
MSER-SIFT	92.3%	11.1%
HAR-DAISY	64.2%	2.4%
Combined	84.1%	5.6%

D. Limitations

One important parameter in our algorithm is the number of patches to be sampled in order to generate the saliency maps; which is application dependant since different tasks require higher or lower confidence values $\hat{P}(O)$ and cluttered scenery requires finer sampling. As of now, this parameter has to be tuned manually.

Further experimentation needs to be done in order to find out the scalability of our proposed method e.g. how the processing time changes as we increased the number of features. With greater number of features the model becomes more robust to image distortions but perhaps the required processing time will prohibit the implementation of a tracking module. On the other hand, experiments were done only with single snapshots; for tracking it is possible to use the history of the diver's pose in order to narrow down the search area. In our framework, the probabilities of the exploration map $P(E)$ in section III will have to be changed depending on the previous analysed images.

Also it is important to mention that in the diver localization scenario, bubbles from the oxygen tank were the major source of diver's occlusion. For some applications, these oxygen bubbles can be considered as a reference for the diver's position, or their location can be used as a substitute when the diver cannot be seen. In this work, we chose not to consider them as part of the diver because our goal is to use this framework to enhance Human-Robot-Interaction, where precise location of the diver's limbs is needed. For this reason, oxygen bubbles that occluded the diver in the training dataset where manually labelled as part of the background scenery.

V. Conclusions

In this paper, we introduced a variation of the Nearest-Class-Mean Forests in order to aggregate multiple features without concerning about memory compression problems, scalability or performance degradation. The use of different type of features to represent complex environments such as underwater terrain outperforms single-feature approaches, as shown in the diver-localization scenario where high variance illumination, low contrast and occlusions are present. This is due to the fact that each feature copes better with specific type of image distortions, and they complement each other when aggregated into a single model.

In the process of developing the MD-NCMF framework, we showed that it offers state of the art accuracy for object recognition by testing it with the GRAZ-02 dataset. More tests are needed to quantify its scalability and performance when the input data (number of classes and samples) starts incrementing. Nonetheless, for our application of interest, the algorithm successfully recognizes the diver and generates a saliency map of his location (pose) showing confidence values. These saliency maps adjust to the diver's limbs and core more naturally; thus, they facilitate the processing of human-action understanding algorithms necessary in HRI applications e.g. gesture detection and interpretation. Also, we have an indicator of which image-region detectors and descriptors work best with underwater imagery thanks to the real-life data gathered during AUVs field testing. Further work needs to be done in order to offer an insight of how to tune some of the parameters used and to implement a full diver's tracking module.

Acknowledgements

The research leading to the results presented here has received funding from the European Community's Seventh Framework Programme (EU FP7) under grant agreement n. 611373 "Cognitive autonomous diving buddy (CADDY)".

References

[1] M. Chantler, D. Lane, D. Dai, and N. Williams, "Detection and tracking of returns in sector-scan sonar image sequences," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 143, pp. 157–162(5), June 1996.

[2] R. Stolkin and I. Florescu, "Probability of detection and optimal sensor placement for threshold based detection systems," *Sensors Journal, IEEE*, vol. 9, no. 1, pp. 57–60, 2009.

[3] R. Kessel and R. Hollett, "Underwater intruder detection sonar for harbour protection: State of the art review and implications," in *Proceedings of the IEEE International Conference on Technologies for Homeland Security and Safety (TEHOSS2006), Istanbul, Turkey*, 2006, pp. 207–215.

[4] A. Rodningsby and Y. Bar-Shalom, "Tracking of divers using a probabilistic data association filter with a bubble model," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 3, pp. 1181–1193, 2009.

[5] H.-L. Eng, K.-A. Toh, W.-Y. Yau, and J. Wang, "Dews: A live visual surveillance system for early drowning detection at pool," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 2, pp. 196–210, Feb 2008.

[6] J. Fabic, I. Turla, J. Capacillo, L. David, and P. Naval, "Fish population estimation and species classification from underwater video sequences using blob counting and shape analysis," in *Underwater Technology Symposium (UT), 2013 IEEE International*, March 2013, pp. 1–6.

[7] A. Gebali, A. Albu, and M. Hoeberechts, "Detection of salient events in large datasets of underwater video," in *Oceans, 2012*, Oct 2012, pp. 1–10.

[8] D. Lee, G. Kim, D. Kim, H. Myung, and H.-T. Choi, "Vision-based object detection and tracking for autonomous navigation of underwater robots," *Ocean Engineering*, vol. 48, no. 0, pp. 59 – 68, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0029801812001370

[9] A. Weidemann, G. R. Fournier, L. Forand, and P. Mathieu, "In harbor underwater threat detection/identification using active imaging," in *Proc. SPIE*, vol. 5780, 2005, pp. 59–70. [Online]. Available: http://dx.doi.org/10.1117/12.603601

[10] J. Sattar and G. Dudek, "Underwater human-robot interaction via biological motion identification," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[11] C.-C. Wang, H.-C. Hsu, and M.-S. Zeng, "Hand signal recognition for diver-machine interface," in *OCEANS '02 MTS/IEEE*, vol. 2, 2002, pp. 1231–1236 vol.2.

[12] H. Bülow and A. Birk, "Diver detection by motion-segmentation and shape-analysis from a moving vehicle," in *IEEE Oceans*, 2011.

[13] H. Buelow and A. Birk, "Gesture-recognition as basis for a human robot interface (hri) on a auv," in *OCEANS 2011*, Sept 2011, pp. 1–9.

[14] F. Sun, J. Yu, and D. Xu, "Visual measurement and control for underwater robots: A survey," in *Control and Decision Conference (CCDC), 2013 25th Chinese*, May 2013, pp. 333–338.

[15] G. Xingfang, M. Yaobin, and K. Jianshou, "Ensemble tracking based on randomized trees," in *Control Conference (CCC), 2012 31st Chinese*, July 2012, pp. 3818–3823.

[16] S. Avidan, "Ensemble tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 261–271, Feb 2007.

[17] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool, "Incremental Learning of NCM Forests for Large-Scale Image Classification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, Jun. 2014, pp. 3654–3661.

[18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 4, pp. 591–606, April 2009.

[20] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3304–3311.

[21] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 3384–3391.

[22] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 11, pp. 2624–2637, Nov 2013.

[23] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, 2000, pp. 142–149 vol.2.

[24] F. Moosmann, E. Nowak, and F. Jurie, "Randomized Clustering Forests for Image Classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 9, pp. 1632–1646, Sep. 2008.

[25] A. Opelt and A. Pinz, "A.: Object localization with boosting and weak supervision for generic object recognition," in *SCIA 2005. LNCS*. Springer, 2005, pp. 862–871.

[26] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761 – 767, 2004, british Machine Vision Computing 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885604000435

[27] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds. Springer Berlin Heidelberg, 2002, vol. 2350, pp. 128–142.

[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of International Conference on Computer Vision*, 1999, pp. 1150–1157.

[29] E. Tola, V.Lepetit, and P. Fua, "A Fast Local Descriptor for Dense Matching," in *Proceedings of Computer Vision and Pattern Recognition*, Alaska, USA, 2008.

[30] E. Stollnitz, T. Derose, and D. Salesin, "Wavelets for computer graphics: a primer.1," *Computer Graphics and Applications, IEEE*, vol. 15, no. 3, pp. 76–84, May 1995.