Change Detection in Marine Observatory Image Streams using Bi-Domain Feature Clustering

Torben Möller*, Ingunn Nilssen†‡, Tim W. Nattkemper*
*Biodata Mining Group, Bielefeld University, Bielefeld 33615, Germany
Email: tmoeller@cebitec.uni-bielefeld.de, tim.nattkemper@uni-bielefeld.de
†Statoil ASA, Research and Technology, Trondheim 7005 Norway
‡Department of Biology, Norwegian University of Science and Technology, Trondheim 7491, Norway
Email: innil@statoil.com

Abstract—**Vision based environmental monitoring using fixed cameras generates large image collections, creating a bottleneck in data analysis. In areas with limited background knowledge of the monitored habitat, this bottleneck can often not be overcome by traditional pattern recognition methods. A new change detection method to identify interesting events such as presence and behavior of different species is proposed. The change detection method uses the new Bi-Domain Feature Clustering (BDFC). BDFC integrates the location of a feature vector in the feature space as well as the location in the image into the clustering. Firstly, BDFC is applied to a time dependent representation of the image stream to identify regions of similar change. Secondly it is applied to a time independent representation to group these changes into categories. These categories can rapidly be assessed by a human observer to bypass the time consuming inspection of the whole data set. To make the posterior browsing of detected changes more efficient, a *relevance* factor computed for each category is proposed.**

The approach is demonstrated with experimental runs, using images from the Lofoten Vesterålen ocean observatory, showing the potential to harvest changes of interest and novelties in large image collections.

I. Introduction

Change detection is a commonly used approach to evaluate images taken from a fixed position, and has already been of great interest in a variety of scientific fields [1] such as video surveillance (e. g. traffic) [2]–[5] and remote sensing [6]–[16] (e. g. deforestation monitoring or monitoring of destruction/construction in urban areas). In the context of underwater imagery applications are still limited [17]–[22].

The two most common change detection approaches are background subtraction (BS) and change vector analysis (CVA). BS aims at building and maintaining a background model of the investigated images and then thresholding the difference between the background model and a *target image*. Lots of efforts have been put into background maintenance as well as in computing a suitable threshold for the pixel differences. CVA methods basically compute the difference image between two images before analyzing the vectors of the difference image (so-called *change vectors*).

Whereas most change detection algorithms compute a *binary* output by assigning each pixel to either *true* or *false*, indicating that changes have occurred or not, some algorithms subdivide the set of changed-pixels into various subclasses.

In [12] and [13] the authors introduce CVA algorithms detecting and categorizing changes based on the length and the direction of each change-vector, respectively. Other change detection algorithms that do not fall within the two categories described abov are for instance based on temporal predictive models [23]–[25], significance tests [26] or slow feature analysis [27]. However, all these methods require a human observer to identify changes of interest in a threshold image or a set of transformed images representing changes.

In recent years, a growing number of fixed long-term underwater observatories (FUO, [28]–[30]) have been deployed to monitor marine habitats. The FUO are in most cases deployed in areas of particular interest, such as areas with high biodiversity. The biological diversity and abundance of species present at these sites represent considerable challenges to a change detection approach:

(1) Prior knowledge about which species to find in the monitored areas and/or how they behave is often limited and can therefore not be used in the design of a change detection method.

(2) As some species occur and/or move much more frequent than other species, change detection in pixel values only is not sufficient to detect relevant changes in the monitored scene.

The contribution of this paper is to introduce a new approach to detect various changes in an area with very limited a-priori knowledge. The approach includes the concept of super-pixel segmentation in feature space and the new Bi-Domain Feature Clustering (BDFC). In contrast to existing methods, this approach finds regions of similar change patterns, groups all detected regions into categories and ranks the categories according to a proposed *relevance* based on frequency and pixel differences.

II. Methods

The aim in the analysis of images from an FUO is the identification of regions of interest as connected components with similar change patterns. As similar changes and patterns are expected at different time-points, the similarity of change features at different time-points must be considered as well. The proposed framework (Fig. 1) integrates these aspects by
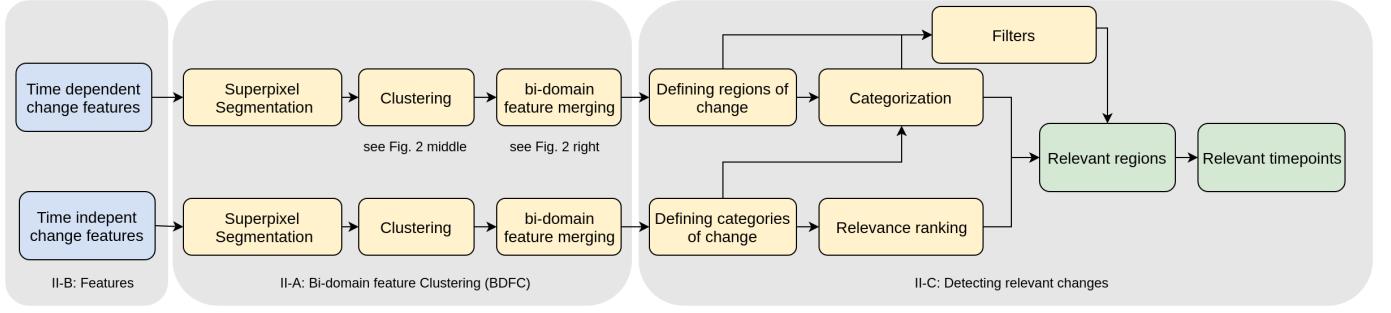
Fig. 1. Time dependent features and time independent features are extracted from an image sequence in Section II-B. Superpixel segmentation and the new Bi-domain Feature Clustering (BDFC) are applied to each of the resulting feature maps in II-A. In Section II-C the change patterns are classified into categories with semantic meaning and *relevant* change is discriminated from *irrelevant* change.

computing two feature representations (Section II-B). The new clustering procedure (Section II-A) is used to cluster the time dependent change features and the time independent change features, respectively. Regions showing similar change patterns are identified using the time dependent change features. The regions are assigned to clusters of the time independent features, which represent categories of change.

The method assumes that all images are in CIELUV [31] color space. Moreover, we assume that the images are taken with the same fixed camera at all time-points implying that all the images have the same height H and the same width W.

The latter assumption allows us to write all images in the processed image sequence as mappings with a common domain $\mathcal{P} = \{(w, h) \mid 1 \leq w \leq W, 1 \leq h \leq H\}$. Consequently, the image taken at time-point t is a mapping $I_t : \mathcal{P} \to \mathbb{R}^3$ assigning to each pixel p a CIELUV color $I_t(p)$. We denote the L, u and v channels of I_t as I_t^1, I_t^2 and I_t^3, respectively.

To retain spatial information during feature extraction so called *feature maps* are used in this paper. A feature map is a mapping $\mathcal{F} : \mathcal{P} \to \mathbb{R}^N$, assigning a feature vector to each pixel and hence is a straightforward generalization of an image. A feature map will always be denoted by $\mathcal{F}_A$ indicating that it is computed by a specific algorithm A ($A \in \{$tmp,int$\}$ Section II-B). Note that in this paper, features are extracted from image sequences not single images.

A. Bi-domain Feature Clustering

Feature maps contain feature vectors and their positions in the image domain, i. e. the pixel position where a feature vector is extracted. Traditional clustering algorithms (vector quantization, hierarchical clustering [32], spectral clustering [33]) do not use the image domain as they are not primary designed for image clustering. To increase the homogeneity in regions of semantic meaning (Fig. 2), we include the image domain into the cluster merging and propose the BDFC algorithm that

(i) performs a traditional clustering of the feature vectors to compute an initial segmentation $\mathcal{I}_A^{\mathrm{b}}$ of the feature map and

(ii) performs a bi-domain merging of clusters that (unlike hierarchical clustering) uses the image domain.

To reduce the influence of noise in the images, we apply superpixel segmentation [34] before applying the Bi-Domain Feature Clustering (BDFC). Superpixel segmentation computes an image representation with a lower level of detail consisting of so called *superpixels* (i. e. connected groups of pixels sharing similar colors) [34]. Thereby, local redundancy is resolved and the complexity of the subsequent BDFC is reduced. However, we apply the approach in the high dimensional feature representation using a straightforward generalization of SLIC [34] to the concept of feature maps. Thus, the pixel set $\mathcal{P}$ is segmented based on the change patterns of the pixels and not based on pixel color.

Let $\mathcal{F}_A$ be a feature map, M the number of superpixels and $\mathcal{M} = \{1, \ldots, M\}$ the set of superpixel-indices. A superpixel segmentation of $\mathcal{F}_A$ is a tuple $(\mathcal{I}_A^{\mathrm{s}}, \mathbf{s}_A)$, where

- the index map $\mathcal{I}_A^{\mathrm{s}} : \mathcal{P} \to \mathcal{M}$ assigns to each pixel p the index of the superpixel containing p in the superpixel map of feature representation A
- $\mathbf{s}_A : \mathcal{M} \to \mathbb{R}^N$ assigns to each superpixel-index m the centroid of the feature vectors $\mathcal{F}_A(p)$ of the pixels p with $\mathcal{I}_A^{\mathrm{s}}(p) = m$.

The superpixel feature map, defined by $\mathcal{F}_A^{\mathrm{s}} = \mathbf{s}_A \circ \mathcal{I}_A^{\mathrm{s}}$, represents $\mathcal{F}_A$ on a lower level of detail. The set of all feature vectors in this map is given by $\mathcal{F}_A^{\mathrm{s}}(\mathcal{P})$ and its cardinality is apparently limited to M.

BDFC computes a segmentation of the feature map $\mathcal{F}_A^{\mathrm{s}}$. The first step in (i) is training a hierarchically growing hyperbolic self-organizing map (H²SOM [35]) with samples from the set of feature vectors $\mathcal{F}_A^{\mathrm{s}}(\mathcal{P})$ (Parameters are addressed in the Result section). A H²SOM is used in this study due to the advantage of simultaneous clustering and dimensionality reduction as well as its good performance even for high dimensional data. However, other clustering algorithms for vector quantization can in principle be applied as well. Let $U_A = \{u_A^1, \ldots, u_A^L\}$ be the set of L prototypes computed by the H²SOM and for $1 \leq \ell \leq L$ let $\mathcal{N}_A(\ell)$ be the set of all neighbors of cluster ℓ in the H²SOM-topology. To compute the initial segmentation to gain the index map $\mathcal{I}_A^{\mathrm{b}}$, each feature vector in $\mathcal{F}_A^{\mathrm{s}}(\mathcal{P})$ is mapped to the index of its best matching
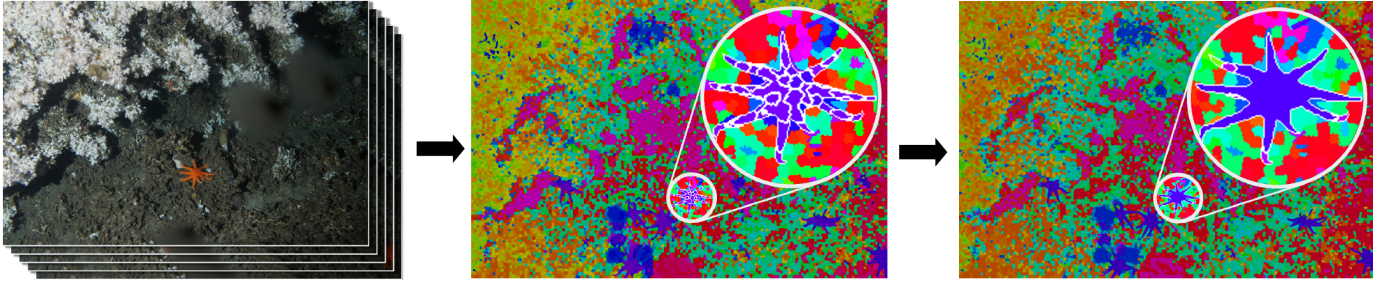
Fig. 2. The BDFC is illustrated for the time independent feature representation. Features are extracted from an image sequence and clustered by a H^{2}SOM [35]. Pixels belonging to one object (see the magnified *starfish*) are initially assigned to different clusters, as indicated by nine shades of purple/blue in the visualization ([36], [37], middle). White contours have been drawn between pixels of the starfish that belong to different clusters, to bring out the fragmentation of the starfish. After cluster merging (right) based on proximities in the image and feature domain, most pixels of the *starfish* belong to one cluster.

prototype in the H^{2}SOM clustering result:

$$\mathcal{I}_A^{\text{b}}(p) = \arg\min_{1 \le \ell \le L} \left\{ \left| \mathcal{F}_A^{\text{s}}(p) - u_A^\ell \right| \right\} \tag{1}$$

A visualization of the H^{2}SOM result can be found in Figure 2.

To take the proximity of cluster indices in $\mathcal{I}_A^{\text{b}}$ into account during the cluster merging procedure in step (ii), let

$$\Omega_A^\ell = \left\{ p \in \mathcal{P} \,\middle|\, \mathcal{I}_A^{\text{b}}(p) = \ell \right\} \tag{2}$$

for every cluster index ℓ. Let $r \in \mathbb{R}$ with $r \ge 1$ ($r = 10$ in this study). We call two pixels p, q *close*, if and only if $d_\infty(p, q) \le r$. With

$$\Omega_A^{\ell,k} = \left\{ (p,q) \in \mathcal{P}^2 \,\middle|\, p \in \Omega_A^\ell, q \in \Omega_A^k, d_\infty(p,q) \le r \right\}$$

we define the *image proximity* of two clusters ℓ and k by

$$\phi_A(\ell, k) = \frac{\sum_{(p,q) \in \Omega_A^{\ell,k}} 1 - \frac{d_\infty(p,q)}{r}}{\left| \Omega_A^\ell \right| + \left| \Omega_A^k \right|}. \tag{3}$$

Note that the choice of the parameter r is not crucial for the cluster merging, but the restriction of the sum in the denominator of $\phi_A(\ell, k)$ increases the computation speed of $\phi_A(\ell, k)$. We use the parameter $n \in \mathbb{N}$ to control the number of clusters merged in the merging process. In this paper, we set the parameter to $n = 5$. We define three conditions to be fulfilled so two clusters with indices ℓ and k are merged:

(A) ℓ is neighbor of k in the H^{2}SOM grid-topology, i. e. $\ell \in \mathcal{N}_A(k)$

(B) the Euclidean distance $d_2(u_A^\ell, u_A^k)$ is one of the lowest n distances $d_2(u_A^\ell, \cdot)$.

(C) the image proximity $\phi_A(\ell, k)$ is one of the highest n image proximities $\phi_A(\ell, \cdot)$

The clusters are merged iteratively, as described in algorithm 1, until no clusters satisfy (A), (B) and (C). A visualization of the final result of the BDFC-algorithm is shown in Figure 2.

B. Feature extraction

Let $(I_1, \ldots, I_T)$ be an image sequence and let the mean image $\overline{I} = 1/T \cdot (I_1 + \cdots + I_T)$ model the background of the scene. To compute change features, we compute a new representation J_t from each image I_t, encoding the

while *found_clusters_to_merge == true* **do**
    found_clusters_to_merge = false
    for $1 \le k < l \le L$ **do**
        if *(A)* **and** *(B)* **and** *(C)* **then**
            found_clusters_to_merge = true
            $u_A^k = \frac{|\Omega_k| \cdot u_A^k + |\Omega_l| \cdot u_A^\ell}{|\Omega_k| + |\Omega_l|}$
            $\mathcal{N}_A(k) = \mathcal{N}_A(k) \cup \mathcal{N}_A(\ell)$
            $\mathcal{I}_A^{\text{b}}(p) = k \;\forall p \in \Omega_l$
        end
    end
end

Algorithm 1: Clusters are merged until no pair of clusters exists that satisfies the conditions (A), (B) and (C). When cluster l is merged to cluster k, u_A^k is set to the weighted mean of the cluster centers. Moreover cluster k absorbs all neighbors of l in the H^{2}SOM-topology as well as all pixels former associated to l.

pixel differences to the mean image: $J_t(p) = \left| \overline{I}(p) - I_t(p) \right|$ ($\forall p \in \mathcal{P}$, $1 \le t \le T$). To account for external effects having impact on the whole image (e. g. change in lighting) we standardize each channel J_t^c ($c \in \{1, 2, 3\}$) of an image J_t, i. e. we define $\tilde{J}_t^c(p)$ to be the standard score of $J_t^c(p)$.

To compute the change features representing the temporal development of change at a given position, we consider the temporal sequence at position p in channel c, defined by $\tilde{\mathbf{J}}^c(p) = \left(\tilde{J}_1^c(p), \ldots, \tilde{J}_T^c(p) \right)$.

To separate trend (i. e. long term signal change) from fluctuation (i. e. short term signal change) we apply wavelet transformation [38] using a Daubechies filter [38]. With W_T and W_F the trend and the fluctuation computed by the wavelet transformation and $\frown$ the concatenation of vectors (i. e. $(x_1, x_2) \frown (y_1, y_2) = (x_1, x_2, y_1, y_2)$), we define the feature map $\mathcal{F}_{\text{tmp}}$ by

$$
\begin{aligned}
\mathcal{F}_{\text{tmp}}(p) \;=\; & W_T\!\left(\tilde{\mathbf{J}}^1(p) \right) \frown W_F\!\left(\tilde{\mathbf{J}}^1(p) \right) \\
\frown \;& W_T\!\left(\tilde{\mathbf{J}}^2(p) \right) \frown W_F\!\left(\tilde{\mathbf{J}}^2(p) \right) \\
\frown \;& W_T\!\left(\tilde{\mathbf{J}}^3(p) \right) \frown W_F\!\left(\tilde{\mathbf{J}}^3(p) \right)
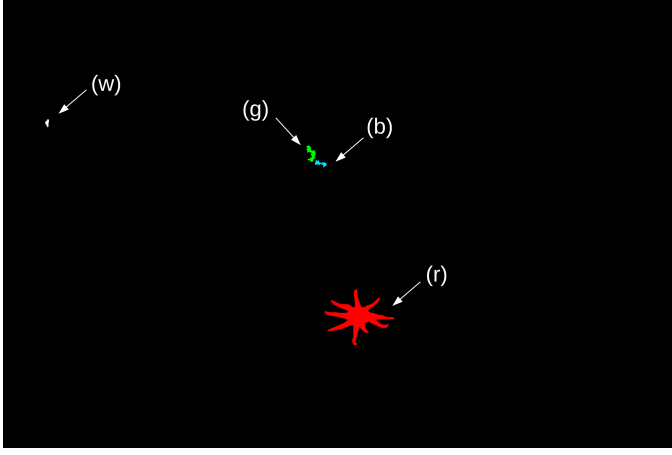\end{aligned} \tag{4}
$$

Fig. 3. Connected components of a cluster using the example of a specific cluster ℓ: Non-black color indicates pixels assigned to cluster ℓ. Cluster ℓ consists of 4 connected components as indicated by the colors white (w), green (g), blue (b) and red (r).

Note that $\mathcal{F}_{\text{tmp}}$ is based of time dependent feature vectors. To compute change features that describe time invariant change patterns, let σ be the permutation arranging the sequence $\tilde{\mathbf{J}}^c(p)$ in descending order to $\hat{\mathbf{J}}^c_{\cdot}$:

$$\hat{\mathbf{J}}^c_{\cdot} = \left(\tilde{J}^c_{\sigma(1)}(p), \ldots, \tilde{J}^c_{\sigma(T)}(p) \right) \qquad (5)$$

Again we use wavelet transformation and define the feature map $\mathcal{F}_{\text{int}}$ by

$$
\begin{aligned}
\mathcal{F}_{\text{int}}(p) &= W_T\left(\hat{\mathbf{J}}^1_{\cdot}(p)\right)^\frown W_F\left(\hat{\mathbf{J}}^1_{\cdot}(p)\right) \\
&\quad ^\frown W_T\left(\hat{\mathbf{J}}^2_{\cdot}(p)\right)^\frown W_F\left(\hat{\mathbf{J}}^2_{\cdot}(p)\right) \qquad (6) \\
&\quad ^\frown W_T\left(\hat{\mathbf{J}}^3_{\cdot}(p)\right)^\frown W_F\left(\hat{\mathbf{J}}^3_{\cdot}(p)\right)
\end{aligned}
$$

C. Detecting relevant changes

To find *relevant* changes in the image sequence, regions that share a particular change pattern are computed. These regions are classified into categories and since a *relevance index* is assigned to each of the categories they can be sorted, permitting a rapid manual posterior browsing. Finally, the time-points where changes occur are determined for every region.

To find regions that show similar change features, we apply BDFC to $\mathcal{F}_{\text{tmp}}$ (Figure 1) to compute merged clusters with centers $U_{\text{tmp}} = \left\{ u^1_{\text{tmp}}, \ldots, u^L_{\text{tmp}} \right\}$ and an index map $\mathcal{I}^{\text{b}}_{\text{tmp}}$. For a cluster ℓ ($1 \leq \ell \leq L$) the set $\Omega^\ell_{\text{tmp}} = \left\{ p \in \mathcal{P} \,\middle|\, \mathcal{I}^{\text{b}}_{\text{tmp}}(p) = \ell \right\}$ is composed of pixels showing similar change patterns, i. e. belonging to the same merged cluster. As shown for one example cluster in Figure 3, a set Ω^ℓ_{tmp} consists of a number n_ℓ of connected components, referred to as $\mathcal{R}^\ell = \left\{ R^\ell_1, \ldots, R^\ell_{n_\ell} \right\}$. Note that $\mathcal{R} = \bigcup_{1 \leq \ell \leq L} \mathcal{R}^\ell$ is a segmentation of $\mathcal{I}^{\text{b}}_{\text{tmp}}$ into regions.

Next, the regions computed from the time dependent feature representation are assigned to groups of changes at different time points, supposed to be caused by the same kind of event (like passing of one specific species). The groups are computed from the time independent feature representation and referred to as *change categories*. We apply BDFC to $\mathcal{F}_{\text{int}}$ (Figure 1) to compute merged clusters with centers $U_{\text{int}} = \left\{ u^1_{\text{int}}, \ldots, u^K_{\text{int}} \right\}$ and an index map $\mathcal{I}^{\text{b}}_{\text{int}}$. Each cluster of the (time independent) cluster indices k ($1 \leq k \leq K$) defines one change category denoted by C_k and each region $R \in \mathcal{R}$ is assigned to a category $\kappa(R)$ by

$$\kappa(R) = \arg\max_{1 \leq k \leq K} \left| \left\{ p \in R \,\middle|\, \mathcal{I}^{\text{b}}_{\text{int}}(p) = k \right\} \right|. \qquad (7)$$

Having regions and categories defined, an index is assigned to every category that estimates the degree of importance for a change. Without knowledge about the experts' preferences, we define the relevance based on frequency and pixel differences. With the *mass center* of the categories defined by

$$m = \frac{1}{|\mathcal{R}|} \sum_{1 \leq k \leq K} \left| \left\{ R \in \mathcal{R} \,\middle|\, \kappa(R) = k \right\} \right| \cdot u^k_{\text{int}} \qquad (8)$$

the relevance $r(C_k)$ of a category C_k is defined by

$$r(C_k) = d_2(m, u^k_{\text{int}}).$$

This allows us to rank the categories C_k according to the computed relevance. To further facilitate the non-automatic posterior browsing, we propose additional filters like the region size $\left| R^\ell_i \right|$ and the length of the prototype u^ℓ_{tmp}. The set of regions satisfying both filters will be denoted by $\tilde{\mathcal{R}} \subseteq \mathcal{R}$.

Finally, all time-points where changes occur are computed for every region $R \in \tilde{\mathcal{R}}$. To do so, we define a measure encoding how well a border of a region matches a contour in the image I_t taken at time-point t: Let B be the set of pixels defining the border of R. For $p \in B$, we denote with i_p, o_p the (well-defined) pixels in the 8-connected neighborhood from p that lie on the line through p perpendicular to B. Note that one of the pixels i_p and o_p is inside R and one is outside R. We define the *coherence* of B to I_t by

$$d^{B,t} = \frac{1}{|B|} \sum_{p \in B} d_1(I_t(i_p), I_t(o_p)) \qquad (9)$$

In order to find all time-points where changes occur in the region R, we compute a threshold τ from the sequence $d^{B,1}, \ldots, d^{B,T}$. Let σ_R be the permutation that brings the sequence $d^{B,1}, \ldots, d^{B,T}$ in ascending order, i. e.

$$d^{B,\sigma_R(t)} \leq d^{B,\sigma_R(t+1)} \qquad \forall 1 \leq t \leq T - 1. \qquad (10)$$

With

$$m = \arg\max_{1 \leq t \leq T-1} d^{B,\sigma_R(t+1)} - d^{B,\sigma_R(t)}, \text{ and} \qquad (11)$$

$$\tau = \frac{d^{B,\sigma_R(m)} + d^{B,\sigma_R(m+1)}}{2} \qquad (12)$$

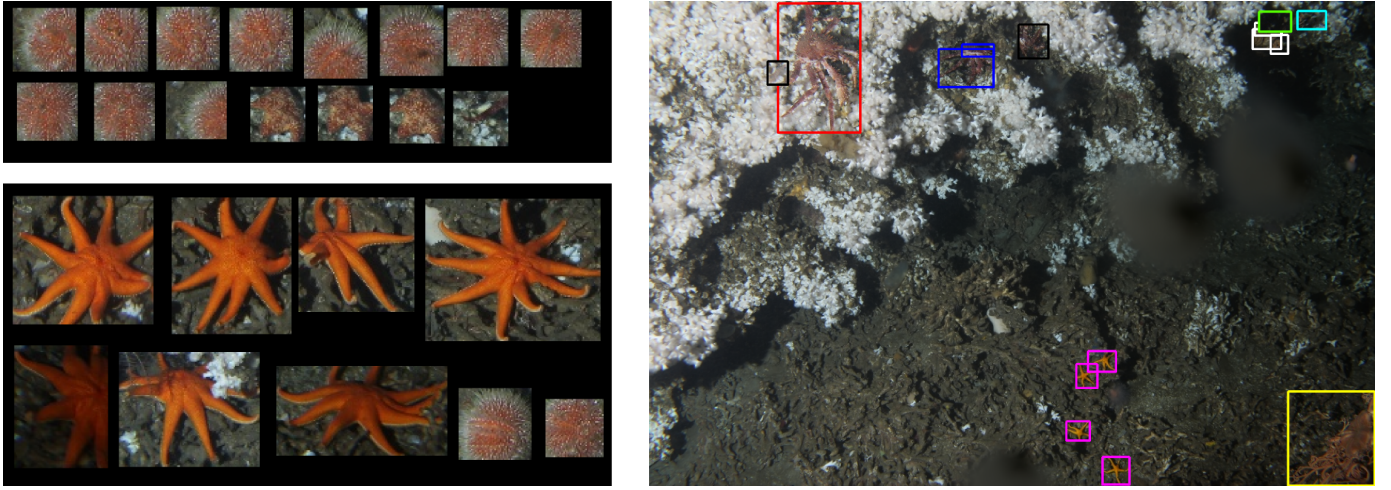we say change occurs at time-point T, if and only if $d^{B,t} > \tau$.

Fig. 4. Changes identified by the algorithm. Left: Grouped changes identified by the algorithm data set 1. The most relevant category and the second most relevant category are shown on top and bottom, respectively. Right: Changes identified by the algorithm for data set 2. All changes within the eight most relevant categories are copied into the background image. Bounding boxes are drawn in different colors indicating different categories, (yellow = medusahead, red = big crab, magenta = starfish, etc.).

III. RESULTS

We evaluate our method on images of the Lofoten-Vesterålen (LoVe) ocean observatory [30]. LoVe is a fixed marine observatory monitoring a coral reef in the Norwegian Sea (N 68° 54.474′, E 15° 23.145′). The observatory that was deployed in October 2013 takes one image every 60 minutes. All images are available online at http://love.statoil.com/. In the present paper, we evaluate the results of the algorithm applied to 6 data sets. Each data set is a sequence of 24 images (one day). We use the same parameter settings for each data set: The superpixel segmentation was performed setting the weight factor (Section II-A) to 8 and the number of superpixels to $M = 12800$ corresponding to a superpixel-size of 80 pixels in average. For the H^{2}SOM clustering, the H^{2}SOM with 3 rings and spread factor 8 was trained with a random set of $0.25 \cdot M$ samples. The *tolerance* and the *radius* used during cluster merging (Section II-A) were set to 5 and 10, respectively. For the wavelet transformation in the feature extraction (Section II-B), we use Daubechies Filter of length 6. See Section IV for more information on the wavelet transformation. As proposed in Section II-C, we apply additional filters to the categorized regions: A region R_i^ℓ of pixels mapped to the cluster center u_{tmp}^ℓ is considered in the evaluation if and only if (i) the size of the region is at least 400 pixels and (ii) $\left| u_{\text{tmp}}^\ell \right| \geq 0.2 \cdot \max_{1 \leq l \leq L} \left| u_{\text{tmp}}^l \right|$.
As an example Figure 4 shows changes identified by the algorithm in 2 of the six data sets.

Figure 5 shows the precision of the algorithm in 6 data sets, each represented by one graph: Each of the N changes identified by the algorithm was evaluated manually either as *relevant* or *irrelevant*. For $1 \leq n \leq N$ we denote by $t(n)$ the number of true positives, i. e. the number of changes within the n most relevant changes that are labeled manually as relevant. The precision is computed as $P(n) = t(n)/n$ and
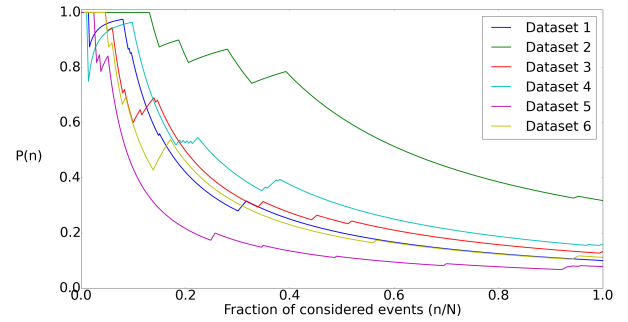

Fig. 5. Each graph shows the evaluation of the methods result when applied to one of the 6 datasets. With N the number of changes found by the method in one dataset and with $n < N$, the precision (i. e. the fraction of true positives in the n most relevant changes) is plotted against n/N.

plotted against n/N. Note that the recall of the method was not computed, as manual detection of all relevant changes is not feasible due to the high abundance of changes in the given image sequences and the absence of a ground truth. However, most regions of the visual field are considered in the graphs and the method assigns a high relevance factor to most true positives. This indicates that most changes of interest belong to the most relevant changes identified by the proposed method.

IV. DISCUSSION AND CONCLUSION

In this paper we have presented a change detection method including the new BDFC algorithm, which shows the potential to be used in various future image processing tasks. The change detection method categorizes and orders various changes present within a fixed image frame by their relevance. Thereby, the method enables the identification of a variety of different changes such as presence and movements of species within images from a fixed position over time. The

feature extraction method has some range of adaptability and provides perspectives for future work. A possible approach is to adapt the change detection to focus on specific species by applying the feature extraction to feature maps instead of images. Another approach is to switch to a multiresolution based wavelet approach in the feature extraction to account for changes that happen slowly overtime (e. g. a change of color of a sessile species). However, this paper has shown that the methods detects movements from unknown species in areas with limited prior knowledge. With this ability, the method has the potential to be used in a variety of image detection scenarios without considerable individual adjustments.

ACKNOWLEDGMENT

REFERENCES

[1] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 294–307, March 2005.

[2] C. Su and A. Amer, "A real-time adaptive thresholding for video change detection," in *Image Processing, 2006 IEEE International Conference on*, Oct 2006, pp. 157–160.

[3] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of gaussians-based background subtraction for bayer-pattern image sequences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 3, pp. 365–370, March 2011.

[4] B. Yin, J. Zhang, and Z. Wang, "Background segmentation of dynamic scenes based on dual model," *Computer Vision, IET*, vol. 8, no. 6, pp. 545–555, 2014.

[5] A. Ferone and L. Maddalena, "Neural background subtraction for pan-tilt-zoom cameras," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, no. 5, pp. 571–579, May 2014.

[6] J. Morisette and S. Khorram, "An introduction to using generalized linear models to enhance satellite-based change detection," in *Geoscience and Remote Sensing, 1997. IGARSS '97. Remote Sensing - A Scientific Vision for Sustainable Development., 1997 IEEE International*, vol. 4, Aug 1997, pp. 1769–1771 vol.4.

[7] M. J. Carlotto, "Detection and analysis of change in remotely sensed imagery with application to wide area surveillance," *Image Processing, IEEE Transactions on*, vol. 6, no. 1, pp. 189–202, Jan 1997.

[8] C. Clifton, "Change detection in overhead imagery using neural networks," *Applied Intelligence*, vol. 18, no. 2, pp. 215–234, 2003. [Online]. Available: http://dx.doi.org/10.1023/A%3A1021942526896

[9] T. Yamamoto, H. Hanaizumi, and S. Chino, "A change detection method for remotely sensed multispectral and multitemporal images using 3-d segmentation," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 39, no. 5, pp. 976–985, May 2001.

[10] F. Bovolo, L. Bruzzone, and M. Marconcini, "A novel approach to unsupervised change detection based on a semisupervised svm and a similarity measure," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 46, no. 7, pp. 2070–2082, July 2008.

[11] T. Celik and K.-K. Ma, "Unsupervised change detection for satellite images using dual-tree complex wavelet transform," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 3, pp. 1199–1210, March 2010.

[12] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 1, pp. 218–236, Jan 2007.

[13] F. Bovolo, S. Marchesi, and L. Bruzzone, "A framework for automatic and unsupervised detection of multiple changes in multitemporal images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 50, no. 6, pp. 2196–2212, June 2012.

[14] L. Bruzzone and D. Prieto, "Automatic analysis of the difference image for unsupervised change detection," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 38, no. 3, pp. 1171–1182, May 2000.

[15] ——, "An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images," *Image Processing, IEEE Transactions on*, vol. 11, no. 4, pp. 452–466, Apr 2002.

[16] L. Bruzzone and F. Bovolo, "A novel circular approach to change detection in pair of images extracted from image time series," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, July 2014, pp. 1140–1143.

[17] R. D. Macleod and R. G. Congalton, "A quantitative comparison of change-detection algorithms for monitoring eelgrass from remotely sensed data," *Photogrammetric Engineering and Remote Sensing*, vol. 64, no. 3, pp. 207–216, 1998.

[18] O. Delaunoy, N. Gracias, and R. Garcia, "Towards detecting changes in underwater image sequences," in *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, April 2008, pp. 1–8.

[19] K. Lebart, C. Smith, E. Trucco, and D. Lane, "Automatic indexing of underwater survey video: algorithm and benchmarking method," *Oceanic Engineering, IEEE Journal of*, vol. 28, no. 4, pp. 673–686, Oct 2003.

[20] K. Lebart, E. Trucco, and D. Lane, "Real-time automatic sea-floor change detection from video," in *OCEANS 2000 MTS/IEEE Conference and Exhibition*, vol. 2, 2000, pp. 1337–1343 vol.2.

[21] M. Johnson-Roberson, O. Pizarro, and S. Williams, "Saliency ranking for benthic survey using underwater images," in *Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on*, Dec 2010, pp. 459–466.

[22] E. Vahtme, T. Kutser, J. Kotta, and M. Prnoja, "Detecting patterns and changes in a complex benthic environment of the baltic sea," *Journal of Applied Remote Sensing*, vol. 5, no. 1, pp. 053 559–053 559–18, 2011. [Online]. Available: http://dx.doi.org/10.1117/1.3653271

[23] S. Kesler and A. Elfishawy, "Adaptive change detection in image sequence," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*, Apr 1990, pp. 2189–2192 vol.4.

[24] Z.-S. Jain and Y. Chau, "Optimum multisensor data fusion for image change detection," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 9, pp. 1340–1347, Sep 1995.

[25] Toyama *et al.*, "Wallflower: principles and practice of background maintenance," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 255–261 vol.1.

[26] M. J. Black, D. J. Fleet, and Y. Yacoob, "Robustly estimating changes in image appearance," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 8–31, 2000.

[27] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 5, pp. 2858–2874, May 2014.

[28] Vardaro *et al.*, "A Southeast Atlantic deep-ocean observatory: first experiences and results," *Limnology and Oceanography: Methods*, vol. 11, pp. 304–315, 2013.

[29] O. N. Canada. (2014) NEPTUNE in the NE Pacific. [Online]. Available: http://oceannetworks.ca/installations/observatories/neptune-ne-pacific/

[30] O. R. Godø, S. Johnson, and T. Torkelsen, "The love ocean observatory is in operation," *Marine Technology Society Journal*, vol. 48(2), 2014.

[31] J. Schanda, *Colorimetry: understanding the CIE system*. John Wiley & Sons, 2007.

[32] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *The Computer Journal*, vol. 26, no. 4, pp. 354–359, 1983.

[33] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1083–1094, May 2015.

[34] Achanta *et al.*, "Slic superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.

[35] J. Ontrup and H. Ritter, "A hierarchically growing hyperbolic self-organizing map for rapid structuring of large data sets," in *Proceedings of the 5th Workshop on Self-Organizing Maps, Paris (France)*, 2005.

[36] J. Kölling, D. Langenkämper, A. Sylvie, K. Michale, and T. W. Nattkemper, "WHIDE - A web tool for visual data mining colocation patterns in multivariate bioimages," *Bioinformatics*, vol. 28, no. 8, 2012.

[37] K. Gorzolka, J. Kölling, T. W. Nattkemper, and K. Niehaus, "Spatio-Temporal Metabolite Profiling of the Barley Germination Process by MALDI MS Imaging," *PLOS ONE*, vol. 11, no. 3, 2016.

[38] I. Daubechies, *Ten lectures on wavelets*, ser. CBMS NSF regional conference series in applied mathematics ; 61. Philadelphia, Pa.: Society for Industrial and Applied Mathematics, 1992, vol. 61.