DOI: 10.1111/cgf.70044
EUROGRAPHICS 2025 / A. Bousseau and A. Dai
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 2

S-ACORD: Spectral Analysis of COral Reef Deformation

Naama Alon-Borissiouk[1] and Matan Yuval[2] and Tali Treibitz[2] and Mirela Ben-Chen[1]

[1]Technion Israel Institute of Technology, Israel
[2]University of Haifa, Israel

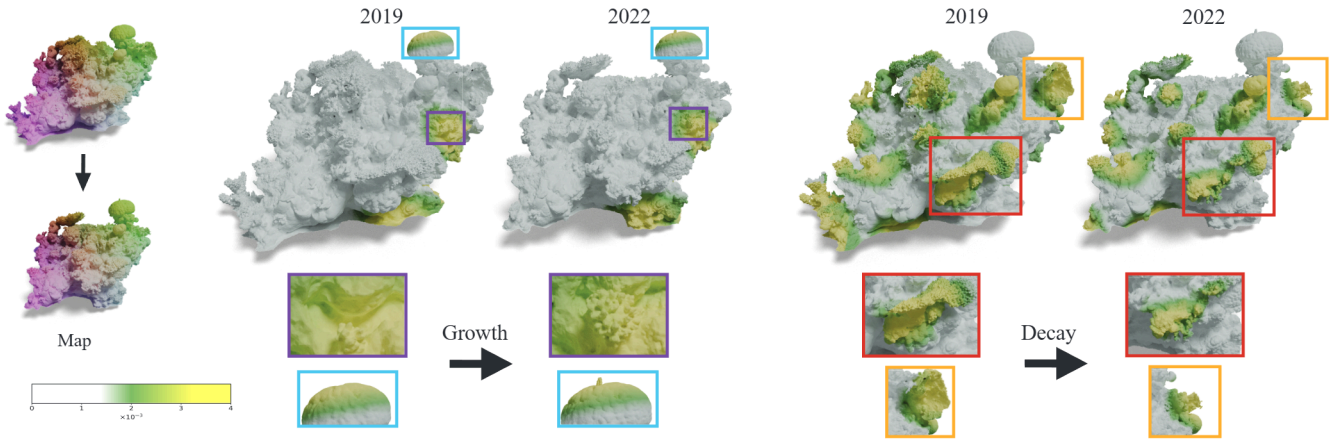

Figure 1: *Analysis and visualization of 3D coral reef models before (2019) and after (2022) a storm. (left) The models are aligned using GO-ICP [YLCJ15], and the map between them is visualized by color-coding: we take a smooth function of the coordinates on the source and map it to the target. Using spectral map visualization [OBCCG13] we identify and show areas on the models of prominent growth (center) and decay (right). For example, we can see the growth of a small extrusion (light blue) and the decay of a colony (red).*

Abstract

We propose an efficient pipeline to register, detect, and analyze changes in 3D models of coral reefs captured over time. Corals have complex structures with intricate geometric features at multiple scales. 3D reconstructions of corals (e.g., using Photogrammetry) are represented by dense triangle meshes with millions of vertices. Hence, identifying correspondences quickly using conventional state-of-the-art algorithms is challenging. To address this gap we employ the Globally Optimal Iterative Closest Point (GO-ICP) algorithm to compute correspondences, and a fast approximation algorithm (FastSpectrum) to extract the eigenvectors of the Laplace-Beltrami operator for creating functional maps. Finally, by visualizing the distortion of these maps we identify changes in the coral reefs over time. Our approach is fully automatic, does not require user specified landmarks or an initial map, and surpasses competing shape correspondence methods on coral reef models. Furthermore, our analysis has detected the changes manually marked by humans, as well as additional changes at a smaller scale that were missed during manual inspection. We have additionally used our system to analyse a coral reef model that was too extensive for manual analysis, and validated that the changes identified by the system were correct.

CCS Concepts

• ***Computing methodologies*** → ***Shape analysis; Mesh geometry models;***

1. Introduction

Coral reefs are an important component in the marine ecosystem, and are an essential habitat for more than 25% of all marine creatures. The effects of global warming on coral reefs has been studied extensively [HBB*03, HGB10, HBB*17]. Death, bleaching, and loss of structural complexity are some of the effects that coral reefs and reef organisms suffer from due to the rising tem-

peratures. Investigating such changes is important for evaluating the condition of marine habitats. Recently, coral reef research has been augmented by underwater photogrammetry, enabling scientists to create 3D models of coral reefs in-situ (e.g., [FBB*16, BD17, FGRL*22, FPW10, MSPS22]). Although photogrammetry is relatively simple, comparing models over time is a challenging problem because there are several sources of noise in 3D reconstruction [LFM*24]. Moreover, reefs are intricate and the marine environment is dynamic, causing objects to move within scenes thus changing their topologies and making the task of matching models over time (i.e., 3D registration) challenging and meticulous.

Structure is one of the most important attributes of reefs because it enables a high level view on the state of the ecosystem. To clarify, reefs are composed of multitudes of organisms and a comparative analysis of the same reef over time can take an object specific/taxonomic approach– i.e., examining the state of each coral separately, or a structural approach i.e., assigning a scalar value to the reef unit through various metrics (e.g., fractal dimension, shelter-space, surface area to volume ratio [YPT*23], where the latter is more holistic, quicker, and cost-effective).

To identify structural changes, scientists first align 3D data-sets of the same coral reef or coral colonies taken at different times in a common frame of reference and then use distance between models [YPT*23, LP20, LMHMM*22, FBB*16]. Our goal is to automate this process, both the *correspondence* (registration) and the *analysis* steps, while addressing the unique challenges that arise in the context of coral reefs.

The foremost challenge is the geometric complexity of the reef models. Due to the fractal nature of coral growth [BR*83, RBSW17, YDRE17], the 3D model contains features at different scales. Hence, our system is required to handle models containing millions of polygons *without* using simplification approaches (that are prone to losing detail). Furthermore, most existing correspondence approaches either require manual annotation of correspondence points, or are based on learning the matching on datasets such as humans, hands, faces etc. These datasets have very different geometric properties than coral reefs. Therefore models trained on these datasets will not generalize to our domain. Finally, the transformation between the reefs is expected to be mostly *rigid*, up to some missing or added components, whereas most contemporary approaches target non-rigid deformations. We show that Globally Optimal ICP [BM92] is an excellent fit for coral reef data: it obtains low correspondence errors when compared with ground truth generated by manually aligned data, *without* requiring manually selected landmarks.

For the analysis part, i.e., change detection of coral reefs over time, we require a *multi-scale* approach, that can identify and visualize changes at different scales. Hence, the Functional Map framework [OBCS*12] with the corresponding visualization [OBCCG13] is an excellent fit for our task. This choice comes with the added challenge of choosing a *functional basis*, which is (1) easily computable on huge datasets and (2) exhibits the multi-scale nature that we require.

Together, our correspondence and analysis tools provide marine biologists with an efficient and accurate system for investigating the deformation of coral reefs and detecting important changes. Figure 1 shows an example of this analysis. We show two models from the dataset, color coded according to the map between them (left). In addition, we show all the regions where our algorithm has indicated growth (center), and decay (right). We zoom in to focus on two areas of interest for each.

We first introduce the dataset that we use [YPT*23], then discuss the design considerations and the setup of our system, and finally demonstrate the effectiveness of our approach by comparing our registration to the ground truth, and comparing our analysis with manual analysis tools used by biologists. To validate our approach we apply our visualization to a very large model that was not analysed previously, and verify with a domain expert that the change areas that we found are indeed relevant.

1.1. Related Work

Shape Correspondence. Shape correspondence is a well-researched topic, and given the vast number of publications in this area a thorough review is beyond our scope. For our goal, we require a method which (1) can handle missing pieces on both models, (2) is fully automatic, (3) can take advantage of the fact that the large bulk of the model does not change, and (4) can handle huge models. There exists recent work on scalable shape correspondence [MCPM25, GRE*23], though we chose to focus on spectral approaches. Many shape correspondence approaches in Computer Graphics are dedicated to non-rigid correspondence [DYDZ22]. While very general, these methods are often more appropriate for articulated characters. Furthermore, these methods frequently rely solely on intrinsic data, potentially missing important cues available for coral reefs. However, methods for rigid correspondence are also prevalent and extensively studied [TCL*12, PCS*15, MAM14]. For rigid correspondence, the gold standard is ICP [BM92] and its myriad of variations, generalizations and improvements. Among these, we chose to use Globally Optimal ICP [YLCJ15], as it fulfills all of our required criteria. We compare to a recent approach for non-rigid registration which is geared towards huge models, and show that GO-ICP indeed performs better when comparing to ground truth obtained by manual registration.

Shape Analysis. Structural complexity deals with the amount of structural features, their sizes, and their spatial arrangement. Complex reefs harbor an array of features of different sizes spread out through them and not clustered in one area. This attribute enables the reef to harbor other marine organisms and provide habitat and niche for them. Therefore, in coral reefs structural complexity correlates with the ability of the ecosystem to provide services. Reefs with high amounts of structural features are able to provide more services, from building habitat and shelter-space for fish and to shoreline protection. Moreover, corals are ecosystem engineers and by growth they build the reef. Therefore, complex reefs often correlate with coral cover and biodiversity [JLS94, GN13, AFDG*09].

Structural complexity is one of the most important features of reefs and the ecological community urgently requires new streamlined methods of studying the changes in reef structure over time. The available methods primarily use model to model distance

[FBB*16, LP20, LMHMM*22] which is time consuming and laborious, especially when trying to view changes colony-by-colony at the reef scale. Our approach considers the *map* between the two 3D models as an object in its own right [OBCCG13, ROA*13], and not only as a means to obtain a *distance*. Thus, we are able to localize the changes the model undergoes, classify them as growth or decay, and sort them by importance, by considering the spectral properties of the map between them.

Scalable Spectral Methods. Our approach is based on converting the pointwise map to a *functional map* [OBCS*12]. The main idea is to consider the map as taking a function on the source model to a function on the target model, instead of mapping a point to a point. This mapping operator is linear (unlike the pointwise map), and can be represented in a reduced basis. The eigenfunctions of the Laplace-Beltrami operator [RWP06, VL08] are often taken as the basis, as they have some optimality properties in a reduced representation [ABK15].

Computing this basis on a mesh with millions of vertices is, however, computationally expensive. Mitigation approaches include spectrum-aware mesh simplification [LLT*20, YS23], intrinsic triangulation [MBRM25, LGC*23], spectral operator coarsening [LJO19, CLJL20], and sampling based approximation of the eigenvectors [NBH18]. Among these, we opted for the latter, as it was closest to working on the original geometry while still being computationally feasible. Recently, a functional map approach which is applicable to large meshes was suggested [MO23]. We demonstrate that computing the pointwise map using GO-ICP leads to smaller ground truth errors than computing a functional map using this approach and then converting it to a pointwise map.

1.2. Contribution

Our main contribution is an end-to-end system for analysing 3D models of coral reefs, including:

- Demonstrating the applicability of GO-ICP [YLCJ15] to automatic coral reef registration and correspondence, including a favorable comparison to a recent functional map method for large meshes [MO23].
- Demonstrating the applicability of FastSpectrum [NBH18] for the spectral representation of maps between large scale models.
- Multi-scale identification of coral reef changes using spectral map visualization [OBCCG13], which (1) correctly reproduces interest points identified by humans, (2) identifies *additional* points of interest missed by humans, and (3) allows us to analyse a full reef that was not previously analyzed.

2. The Coral Reef Data-Set

In March 2020, a severe storm struck and caused significant damage to a coral reef in the Red Sea's bay. To assess the impact of the storm on coral reefs, a unique 3D data-set was collected using underwater photogrammetry, before, directly after, and two years after the storm [YPT*23].

The coral reef models were captured from 2019 to 2022. Data from 2019 were collected before the storm and used as a baseline for comparison, while the 2020 data were obtained three months after the storm to assess immediate impacts. In 2022 models were captured to evaluate the recovery of the reefs. Overall, seven shallow reefs were documented, with three models captured for each reef over the years, resulting in a data-set of 21 textured models called Reefs4D [YPT*23]. In our paper we used six of these models (C1-C5 and Kza5m), yielding a data-set of 18 models.

The 3D models were reconstructed using Structure from Motion (SfM) algorithms in a commercial software: Agisoft Metashape [Agi18], applied to a sequence of 2D images captured with a Nikon DSLR camera. In addition, the models were scaled to uniform dimensions using scale bars with known dimensions, which are embedded in the scene and used for calibration during the reconstruction process. The models from 2019 and 2020, which correspond to the periods before and after the storm, were manually registered and utilized as the ground truth (GT) in our algorithm. The manual registration process was carried out using the CloudCompare software [Clo22]. There, four corresponding points on both models were selected manually, and used as anchors for ICP [BM92] registration. We note that the floor near the corals may change due to the storm and different parts of it may be removed during acquisition and reconstruction. The models have over one million vertices, and models from different years have, in some cases, different resolutions. The Reefs4D dataset is available at https://zenodo.org/records/14616671.

3. Background

3.1. Notation

A triangle mesh is given by $M = (V, \mathcal{F})$, where $V, \mathcal{F}$ are the vertices and faces, respectively, and we+ denote $n = |V|, m = |\mathcal{F}|$. A pointwise map between two triangle meshes is denoted by $T_{21} : M_2 \to M_1$, and maps vertices on M_2 to vertices on M_1. We encode the map as a binary row stochastic matrix $P_{21} \in \{0,1\}^{n_2 \times n_1}$, denoted pointwise correspondence matrix. The matrix element $P_{21}(i,j)$ is 1 if and only if T_{21} maps the i-th vertex of M_2, to the j-th vertex of M_1. $C_{21} \in \mathbb{R}^{k_2 \times k_1}$ denotes the functional map from functions on M_1 to functions on M_2 represented in a reduced basis of sizes k_1, k_2, respectively. $\Phi_i \in \mathbb{R}^{n_i \times k_i}$ is the reduced basis of mesh M_i. We usually denote M_1 as the earlier mesh chronologically, for example, M_1 is a reef mesh from 2019 and M_2 from 2020.

3.2. GO-ICP

GO-ICP [YLCJ15] is a globally optimal algorithm for rigid point cloud registration, based on the well-established ICP (Iterative Closest Point) algorithm [BM92]. ICP converges to a local minimum that depends on the initialization. GO-ICP, on the other hand, uses a Branch and Bound (BnB) approach, combined with ICP, to find the global transformation (rotation and translation), that minimizes the objective. BnB segments the problem into manageable sub-problems and evaluates their respective bounds, to identify the optimal solution within the subset. By alternating between ICP and BnB, GO-ICP effectively avoids local minima, and the algorithm converges to a globally optimal solution (if given enough iterations).

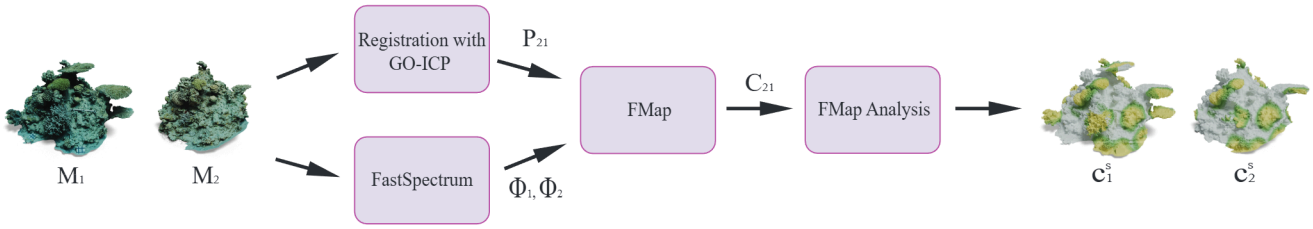

Figure 2: *Framework overview: we first register the two meshes using GO-ICP and obtain a pointwise map. Then we compute an approximation of the Laplace-Beltrami eigenvectors using the FastSpectrum algorithm. The pointwise map together with the eigenvectors is used to compute a functional map, which is then analyzed for visualizing the differences between the models.*

3.3. FastSpectrum

FastSpectrum [NBH18] is a method for approximating the lowest eigenvalues and their corresponding eigenfunctions of the Laplace–Beltrami operator. The approach involves sampling points on the input shape to construct locally supported functions, and then modifying these functions via a partition of unity. This process reduces the search space to a manageable subspace of functions on the mesh. The final step solves a low-dimensional eigenproblem to approximate the eigenfunctions, significantly reducing computational complexity. As proven in the paper, the approximated eigenvectors are orthonormal with respect to the mass matrix of the mesh.

3.4. Spectral Map Visualization

This approach [OBCCG13] takes as input a map T between two shapes, converts it to a functional map C, and then uses the singular vectors and singular values of C (projected back to the input surface), for visualizing the differences between the shapes. The main idea is that the singular values of C imply both the *type* of change, i.e., expansion or shrinkage, and its *importance*.

Notably, functional maps indicate the linear change of basis in the spectral domain $\mathbb{R}^k$ that is required to map between functions on one shape to functions on the second shape. Hence, their SVD decomposition (much like the SVD decomposition of linear transformations in $\mathbb{R}^3$), indicates both the *directions of change* - through the singular vectors, and the *importance* of the change through the singular values.

For example, the functional map corresponding to an isometry, i.e., a pointwise map that preserves geodesic distances, is an orthonromal matrix [ROA*13], namely all its singular values are 1. More generally, functional maps that *contract* the source shape have singular values smaller than 1, with corresponding singular vectors whose support on the shape highlights where the contraction occurs. Similarly, functional maps that *expand* have singular values larger than 1 with similarly corresponding singular vectors. We use these singular values and singular vectors to highlight the changes between coral reefs.

4. Method

4.1. Overview

The overview of our method is summarized in Figure 2. We get as input two models, M_1, M_2 and register them using GO-ICP, obtaining a pointwise correspondence matrix P_{21}. We use FastSpectrum to compute an approximation to the eigenvectors and eigenvalues of the Laplace-Beltrami operator, getting Φ_1, Φ_2. Using the pointwise map and the eigenvectors we compute a functional map C_{21}, which is then analysed using SVD to obtain the main shrinkage and growth distortions. The shrinkage changes are visualized using color coding c_1^S, c_2^S, respectively.

4.2. Pre-Processing

We first pre-process the input meshes (raw data) to be manifold using Meshlab [CCC*08]. This is required, since we compute the Laplace-Beltrami operator, as well as an approximation to its eigenvectors and eigenvalues.

4.3. Registration

Mesh Normalization. The GO-ICP algorithm requires that the input data is normalized in the unit cube. We used two methods for centering and scaling the data:

- **Ground Truth Data (2019 to 2020):** These meshes had been manually registered and were used as ground truth for testing our algorithm. As part of the manual registration, the meshes were centered around the origin using the distance from M_1 (2019) as the reference point. To maintain the relative scale between the meshes, we scaled them using a combined scaling factor that considers both M_1 and M_2 (2020). We calculated and compared their maximum absolute distances from the origin, applying a safety margin (avoiding boundary issues) by scaling down the larger distance. This preserves the manual registration within the unit cube prior to the application of random transformations used to test our algorithm.
- **Other Data (2019 to 2022 and 2020 to 2022):** Each mesh in these datasets was centered individually at the origin. We applied a combined scaling factor to these datasets as well, to preserve the relative proportions of the meshes.

The authors of GO-ICP recommend sub-sampling the target mesh

M_2 for better performance. Given that our 3D models are significantly larger than those used in the original paper, we experimented with several sub-sampling sizes, and concluded that using $1/150$ of the number of vertices of M_1 is the most effective sampling size.

In the GO-ICP process, we opted not to use trimming, although it was recommended, because the algorithm performed satisfactorily without it. Finally, after the registration process, we transformed the registered M_2 back to its original size using the inverse transformation of M_1 to the cube. This step was based on the assumption that M_2 was accurately registered to M_1.

We fine-tuned the remaining GO-ICP parameters to optimize for the performance vs. the accuracy, using a single shape pair. This involved conducting 100 runs of the algorithm on our ground truth data and evaluating the outcomes by calculating the transformation errors relative to the random transformations computed in the preprocessing stage. The errors were quantified as the percentage of π for rotation and as a percentage of the diagonal of M_1's bounding box for translation.

4.4. Spectral Data Computation

The spectral basis used for functional maps varies depending on the application. The most common basis is given by the lowest k eigenvectors of the Laplace-Beltrami operator of the mesh. However, given the large number of vertices, the standard computation process proved to be excessively time-consuming (on the order of hours for computing 300 eigenvectors on a mesh with $700K$ vertices). Instead, we use the FastSpectrum algorithm [NBH18], which is specifically designed to manage dense meshes by sampling them and then extracting *approximate* eigenvectors, where we use 300 sample points. This leads to a reasonable computation time, on the order of a few minutes, for the reef models. The outputs of this stage are the spectral bases $\Phi_i \in \mathbb{R}^{n_i \times k_i}$, for $i \in \{1,2\}$, where n_i is the number of vertices of M_i, and k_i the number of basis vectors that we compute for M_i.

For effective map analysis, k_1 should be considerably larger than k_2, as recommended in the map visualization paper [OBCCG13]. Hence, we use k_1 between $230-250$ (according to the number of eigenvectors returned by FastSpectrum, which is always smaller than the sample size), and $k_2 = 30$ for all the experiments.

4.5. Map Computation

The next step is to compute the pointwise map T_{21} from the aligned meshes, the pointwise correspondence matrix P_{21}, and the corresponding functional map matrix C_{21}. We use the K-Nearest Neighbors (KNN) on the aligned meshes for computing the pointwise map $T_{21} : M_2 \to M_1$, by pairing each point of M_2 with its closest point in M_1. The pointwise correspondence matrix $P_{21} \in \mathbb{R}^{n_2 \times n_1}$ is derived in a straight-forward manner from T_{21} as described in Section 3.1. The functional map is given by $C_{21} = \Phi_2^{\dagger} P_{21} \Phi_1 \in \mathbb{R}^{k_2 \times k_1}$.

4.6. Functional Map Analysis and Visualization

In the final stage of our system, we analyze the functional map C_{21} [OBCCG13]. As it is a linear map from $\mathbb{R}^{k_1}$ to $\mathbb{R}^{k_2}$, its singular vectors identify the main directions of change, and the singular values represent the scale of the change. For example, if C_{21} is a rotation matrix (if we take $k_1 = k_2$), then all its singular values are 1, and the corresponding pointwsie map is an isometry – implying that no intrinsic change has occurred. Similarly, the distance between the singular value and 1 indicates the amount of change. See [OBCCG13] Theorem 4.1 for the theoretical underpinning of this approach.

To this end, we compute the singular value decomposition, $USV^T = C_{21}$, where $U \in \mathbb{R}^{k_2 \times k_2}, V \in \mathbb{R}^{k_1 \times k_1}$ are orthogonal matrices of left and right singular vectors, respectively, and $S \in \mathbb{R}^{k_2 \times k_1}$ is a rectangular diagonal matrix of singular values. The singular values on the diagonal of S, denoted by $s_1, s_2, \ldots, s_{k_2}$ indicate the type of change that the map encodes: growth for $s_i > 1$ and shrinkage for $s_i < 1$. The larger $|s_i - 1|$ is, the larger the change is. In Section 5.2 we show the singular value graphs for the coral reef dataset and discuss the implications.

The right singular vector $v_i \in \mathbb{R}^{k_1}$ encodes the area on the surface where the change implied by the corresponding singular value s_i is localized. To visualize it, we color-code the vertices on M_1 using the function $c_1^i = (\Phi_1 v_i)^2 \in \mathbb{R}^{n_1}$, and the vertices on M_2 using $c_2^i = P_{21} c_1^i \in \mathbb{R}^{n_2}$ (the square is taken element-wise). This coloring visualizes the important regions of shrinkage and growth for each singular vector, showing corresponding colors on M_1 and M_2.

To visualize *all* the changes indicating *growth* we take $c_1^G(v) = \max_i \{c_1^i(v) \,|\, s_i > 1\}$, where $v \in V_1$. Similarly, to visualize all the changes that indicate *shrinkage*, we use $c_1^S(v) = \max_i \{c_1^i(v) \,|\, s_i < 1\}$. For the corresponding colors on M_2 we use $c_2^G = P_{21} c_1^G$ and $c_2^S = P_{21} c_1^S$. See for example Figure 1.

We note that the choice of k_2 determines the *maximal* number of changes that we detect, as it determines the dimensions of C_{21}, and thus the number of singular vectors. This is a limitation, as we may miss changes if k_2 is too small as we can see in Section 5.5. In our experiments $k_2 = 30$ was usually sufficient.

Table 1: *The number of vertices of the models in our dataset.*

Model	2019	2020	2022
C1	874K	630K	1.8M
C2	3.2M	326K	1.1M
C3	2.4M	468K	1.9M
C4	1.7M	506K	1.6M
C5	1.5M	286K	1.1M
Kza5m	3.2M	3.5M	3M

5. Experimental Results

The statistics of the dataset, specifically the number of vertices for all the models, is given in Table 1. We did not perform any processing on the meshes, except for making them manifold and, in the registration phase, normalizing them to fit within a unit cube. We implemented the registration and analysis code in Python. We used a machine with an Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz and an NVIDIA GeForce GTX 1080 GPU, where the GPU was only utilized for FastSpectrum. We did not perform any additional optimization of the code.

Table 2: *Statistics of GO-ICP on GT data*

Model	Translation Err [M_1 bbx diagonal]	Rotation Err [%π]	GO-ICP Time [sec]	FastSpectrum Time [sec]	Fmap Time [sec]	Analysis Time [sec]	Total Time [sec]
C1	0.3	0.3	67.8	55.5	19.5	9.6	152.5
C2	0.7	1.6	172.8	135.6	43.3	19.7	371.5
C5	0.1	0.5	107.8	70.5	18.0	11.2	207.5
Kza5m	0.4	0.2	192.6	314.2	86.7	47.8	641.2

Table 3: *Timing of ScalableFM on GT data*

Model	Process Mesh Time [sec]	Approx Spectrum Time [sec]	ZoomOut [sec]	ScalableFM Total Time [sec]	FMAP to Pointwise Map [sec]
C1	70.7	4.8	10.6	86.0	1644.7
C2	137.7	12.5	11.5	161.7	1441.7
C5	85.8	5.8	12.3	189.7	748.6
Kza5m	248.1	4.3	7.5	260.0	6392.1

5.1. Registration

To validate the efficacy of GO-ICP on our dataset we use the manually labeled ground truth, which is available for the models C1, C2, C5 and Kza5m, for the map between 2019 and 2020. We normalize and center the models as described in Section 4.1, and then randomize a rotation and a translation, which is applied to M_2. The transformed M_2 and M_1 are provided as input to GO-ICP. To evaluate the error, we compute the difference between the result and the input rotation as a percentage of π. Rotations are represented using the angle-axis representation, and the rotation error is given by angular distance (see GO-ICP [YLCJ15], Eq.(6)). The difference between the translations is given as a percentage of the diagonal of the bounding box of M_1. We repeat this experiment 100 times for each pair, and average the results.

Table 2 shows the results. We note that all the models except for C2 achieved 0.5% or less error, for both rotation and translation. From the C1-C5 models, C2 has the highest resolution in 2019, and the second lowest resolution in 2020. This has led to somewhat larger average errors of 0.7% and 1.6% for translation and rotation errors, respectively. Increasing the number of samples improved the results, however we opted to use the same parameters for all models as the obtained results on C2 were still satisfactory. Figure 13 has additional data on the distribution of the errors across the experiments, and the correlation between the time spent in GO-ICP and the resulting error.

Table 2 additionally shows the time spent in each part of our algorithm. We note that, as expected, GO-ICP and FastSpectrum (total time on both models) are the heaviest components, accounting for around 80% of the time. In total, even on the largest model (Kza5m), where both 2019 and 2020 have more than 3M vertices, the whole computation took around 10 minutes.

5.1.1. Comparison to ScalableFM

Scalable Functional Maps (FM) [MO23] is a correspondence method for dense non-rigid models, which assumes that an initial map is available for a subset of the vertices. The algorithm uses FastSpectrum with a modification in the sampling technique to compute a sampling of the vertices. Then, instead of taking the full reconstructed eigenvectors (as we do), it restricts computation to the reduced space. Finally, it computes the full functional map using ZoomOut [MRR*19].

We use the ground truth (GT) models as the input initial map to ScalableFM. Note that ScalableFM normalizes the input models to have unit surface area, which incidentally also localizes them in the unit cube. Following the guidelines in the paper, we sample the meshes to 3000 vertices and extract approximately 250 eigenvectors, as we do in our algorithm. The rest of the parameters are the same as provided in the demo code. For the initial map, we create a functional map of size 100×100 based on K-nearest neighbors

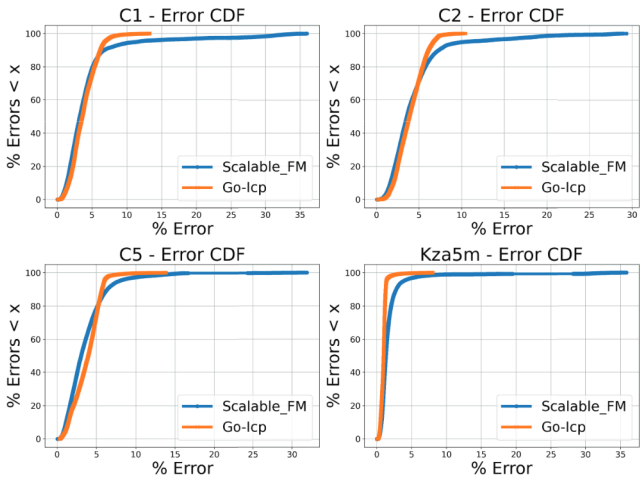

Figure 3: *Ground truth mapping error of ScalableFM in comparison to our approach. We calculate the distances of vertices in M_1 (2019) to their corresponding vertices in M_2, expressed as a percentage of the diagonal length of M_2's bounding box (x axis). The y axis shows the cumulative distribution of the error.*

(KNN) of the samples, as implemented in the ScalableFM demo. The final output from the ZoomOut algorithm is a 200×200 functional map. We use the ScalableFM demo to extract the full pointwise map from the functional map for comparison with our results.

First, we compare ScalableFM and our optimal results, using Go-ICP for registration of the GT models (2019 to 2020) subject to random transformations, as described in Section 5.1. To measure the accuracy of the mappings, we use the distances of vertices in M_1 (2019) to their corresponding vertices in M_2, expressed as a percentage of the diagonal length of M_2's bounding box. Explicitly, we compute $\text{err} = \frac{\|X_1[p2p,:]-X_2\| \times 100}{\text{diagonal\_len}(M_2)}$, where $X_i \in \mathbb{R}^{n_i \times 3}$ represents the 3D coordinates of the vertices of M_i. Figure 3 displays the cumulative distribution plots of this error metric for each of the four coral meshes analyzed, for ScalableFM and our GO-ICP results.

Figure 4 shows qualitatively the mapping result of ScalableFM (b) compared to our mapping result (c). The source 2019 models are shown in (a), and the maps are color-coded using a smooth function of the x, y, z coordinates of the source. Notably, the ScalableFM mappings exhibit significant distortions, despite the use of GT models as input.

In this phase, we measured the time efficiency of the ScalableFM algorithm in a single run, detailed in Table 3, and found that the results were faster than those from our existing pipeline (see Table 2). However, the computation of the pointwise correspondence matrix from the functional map requires significantly more time, taking approximately 25 minutes or longer. This extended duration is critical as we need to compute P_{21} for subsequent comparison with our visualization pipeline. The timing for ScalableFM is categorized in Table 3 into "Process Mesh Time" for all normalization and sampling processes, "Approx Spectrum Time" for the modified FastSpectrum, and "ZoomOut" for the KNN, initial mapping, and ZoomOut refinement.

Next, we test our visualization pipeline using the outputs from ScalableFM to compare it to our visualization results. We use the outputs of ScalableFM, which include the P_{21} mapping and the approximated full eigenvectors to compute a functional map of size $k_2 \times k_1$, as required by our approach (note that the functional map that ScalableFM outputs is square by default).

For our algorithm, we need P_{12} to map the singular vectors from M_2 to M_1. Therefore, we run ScalableFM again with the inputs swapped to obtain P_{12} as output. Figure 5 illustrates that the ScalableFM map is not accurate enough to be applicable in our visualization pipeline. We observe areas marked as changed in M_2; however, due to mapping errors, an unrelated area is highlighted as corresponding in M_1. For example, we can see in (a) an incorrect correspondence in mesh C1 for singular value $s > 1$. The comparison with the ground truth clearly indicates that our method outperforms ScalableFM in terms of mapping accuracy. We can see in this experiment that if the registration is not good in a way that introduces large area distortions, the visualization will highlight that. For an additional example see Fig. 2(e) in [OBCCG13], where this method was used for finding problematic regions in the map. However, if we trust the registration method (as we do in our method, since we validated v.s. the ground truth), then this approach will correctly visualize the changes that the surface has undergone.

5.2. Singular values

The distribution of the singular values of the functional map encodes information about the trends exhibited by the map. In Figure 6 we show the singular value graphs of all the maps we computed (for all 6 models, 3 maps for each model).

Note for example that the 2019-2020 maps exhibit mostly singular values smaller than 1 (e.g., C3 has *no* singular values above 1). This indicates that most of the changes were losses or decays. This is expected, as this map encodes the change that the storm caused, which was detrimental to the reef. On the other hand, the map of C2 from 2020-2022 exhibits considerably more moderate changes, with most of the singular values in the neighborhood of 1, and many singular values above 1. Indeed, the previous analysis has shown that this period included recovery and growth [YPT*23].

Thus, the singular values graph provides important insight in one view regarding the change that the reef has undergone in the period that has passed between the capture of the first and second model. A deeper understanding of the *location* of the changes is obtained by considering the singular vectors, as we show next.

5.3. Visualization with Singular Vectors

As described in Section 4.6, the singular values are ranked from 1 to k_2, where $s_i > 1$ indicates growth changes from the most to the least significant, and $s_i < 1$ signifies shrinkage from minor to major decay. We examined $k_2 = 30$, covering 30 singular values representing changes. Figure 7 (top row) illustrates the changes in the C2 model from 2019 (top row) to 2022 (bottom row). (d), (e), and (f) depict decay observed in larger values of i with $s_i < 1$, showing visually structural changes where large parts of the branching corals have collapsed. For the value of $i = 2$ and $s_i > 1$ (b) we see boundary difference. Figure 7 (bottom row) further illustrates these dynamics in the Kza5m model, spanning 2019 (top) to 2022 (bottom).

5.4. Visualization: Comparison with GT

In our main experiment, we compare the reef changes that were detected by our algorithm with those identified by biologists. The ground truth (GT) includes the (x, y, z) coordinates of locations marking decay of the coral reef between 2019 and 2020. We use the GT of five models (C1-C5). We exclude Kza5m, which contains over 700 corals and is challenging to track manually, and therefore did not have ground truth annotations.

Figure 8 shows c_1^S on all 2019 models, namely the regions on M_1 where $s_i < 1$, indicating shrinkage. GT points are overlaid as spheres: red for true positives (changes identified by both our algorithm and manual inspection) and black for false negatives (changes noted manually but missed by our algorithm). Our algorithm accurately detected all marked changes in two models (C2 and C5), but missed 5 out of 36 in C1, 10 out of 59 in C3 and 1 out of 6 in C4.

Figure 9 shows some examples of these true positives and false negatives, corresponding to Fig. 8. We show for each example: our coloring based on the singular vectors (top two figures), the image texture (bottom two figures), where the 2019 models are the left two figures, and 2020 models are on the right. Figures (b), (d), (e),

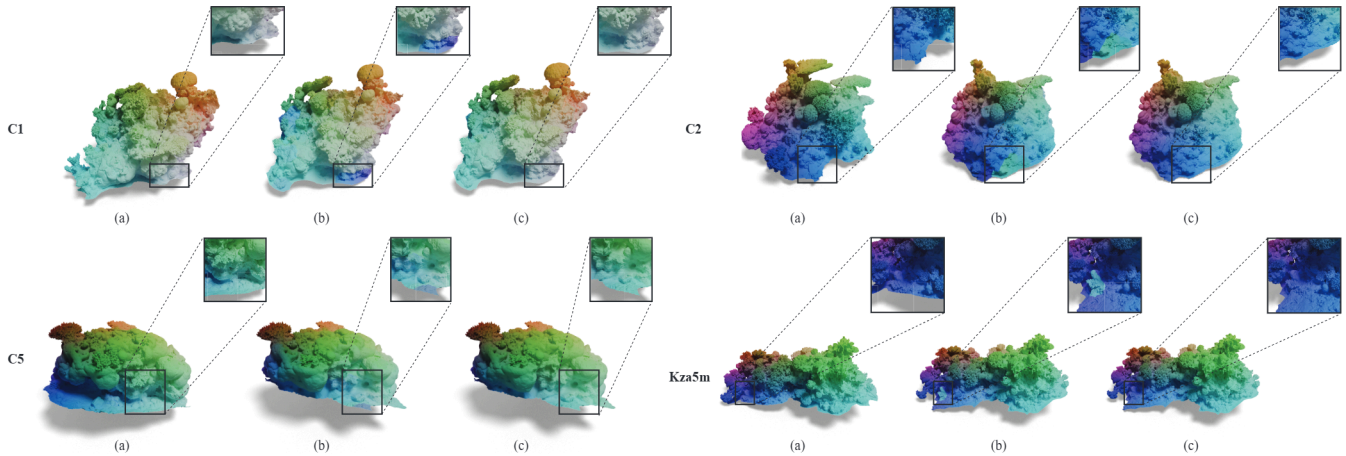

Figure 4: *Visual comparison of correspondences with ScalableFM algorithm on the models C1 (top-left), C2 (top-right), C5 (bottom-left) and Kza5m (bottom-right) between 2019-2020. (a) 2019, (b) 2020 ScalableFM (c) 2020 Ours. Note the discontinuities evident in the maps in (b). The maps are color-coded using a smooth function of the x, y, z coordinates of the source. See results section for more details.*

and (f) illustrate successfully detected removals, highlighting our algorithm's successes. On the other hand, In (a) and (c), we observe missed changes: (a) shows an undetected broken part in C1, and in (c) although three colonies were removed from C3, our algorithm detected only two, mistakenly identifying the three similar changes as a single event. Note, though, that the missing encrusting coral (that looks like "suction cups") is visible almost exclusively in the texture image, and not as a geometric deformation, since its polyps are too small compared to the resolution of the reconstruction. Therefore, this change is missed by our algorithm. It is also possible for false negatives to arise if there are more notable changes than the top $k=30$ changes that we identify. Hence k must be set by the user for the specific size and use case of the map, i.e., for maps with many expected changes choose $k>30$.

Finally, our algorithm detects true changes that were missed during manual inspection. These include delicate changes that can be overlooked due to the methodologies currently used by biologists or simply human error. See e.g., Fig. 10, C2 model; the change is notable, suggesting that its omission was likely an oversight.

5.4.1. Outliers

We identified several changes that biologists did not mark in their ground truth (false positives), including differences in coral boundaries (variations in the boundaries obtained during the 3D reconstruction), and non reef elements such as scale bars and color charts used for photogrammetry reconstruction. Examples of such boundary changes, significant enough for our algorithm to identify, are shown in Figure 11 (a,b) and Figure 7 (c). Examples of non-reef elements appear in Figure 11 (c,d).

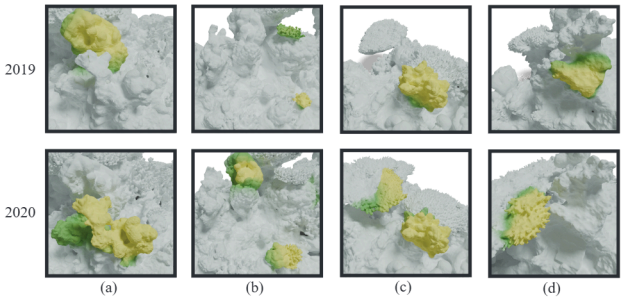

Figure 5: *Visualization pipeline using the ScalableFM mapping results. Top (resp. bottom) row shows the 2019 (resp. 2020) meshes. (a) C1 mesh with singular value $s>1$, (b) C1 mesh with $s<1$, (c) C2 mesh with $s>1$, (d) C2 mesh with $s<1$. Note that since the mapping is incorrect the highlighted regions in the 2019 meshes do not correspond to those of the 2020 meshes.*

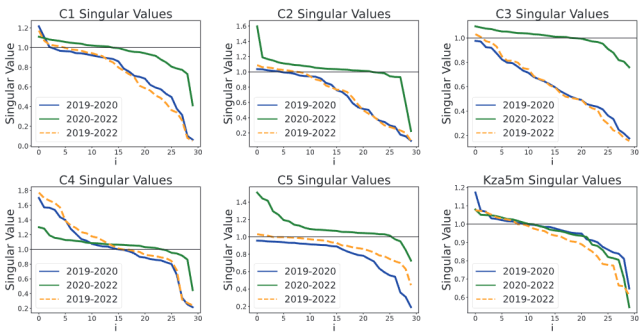

Figure 6: *Singular values of the functional maps between early and later years (2019-2020,2019-2022,2020-2022) for all the models. Note that most of the 2019-2020 graphs show singular values below 1, indicating shrinkage and loss, as expected following the storm.*

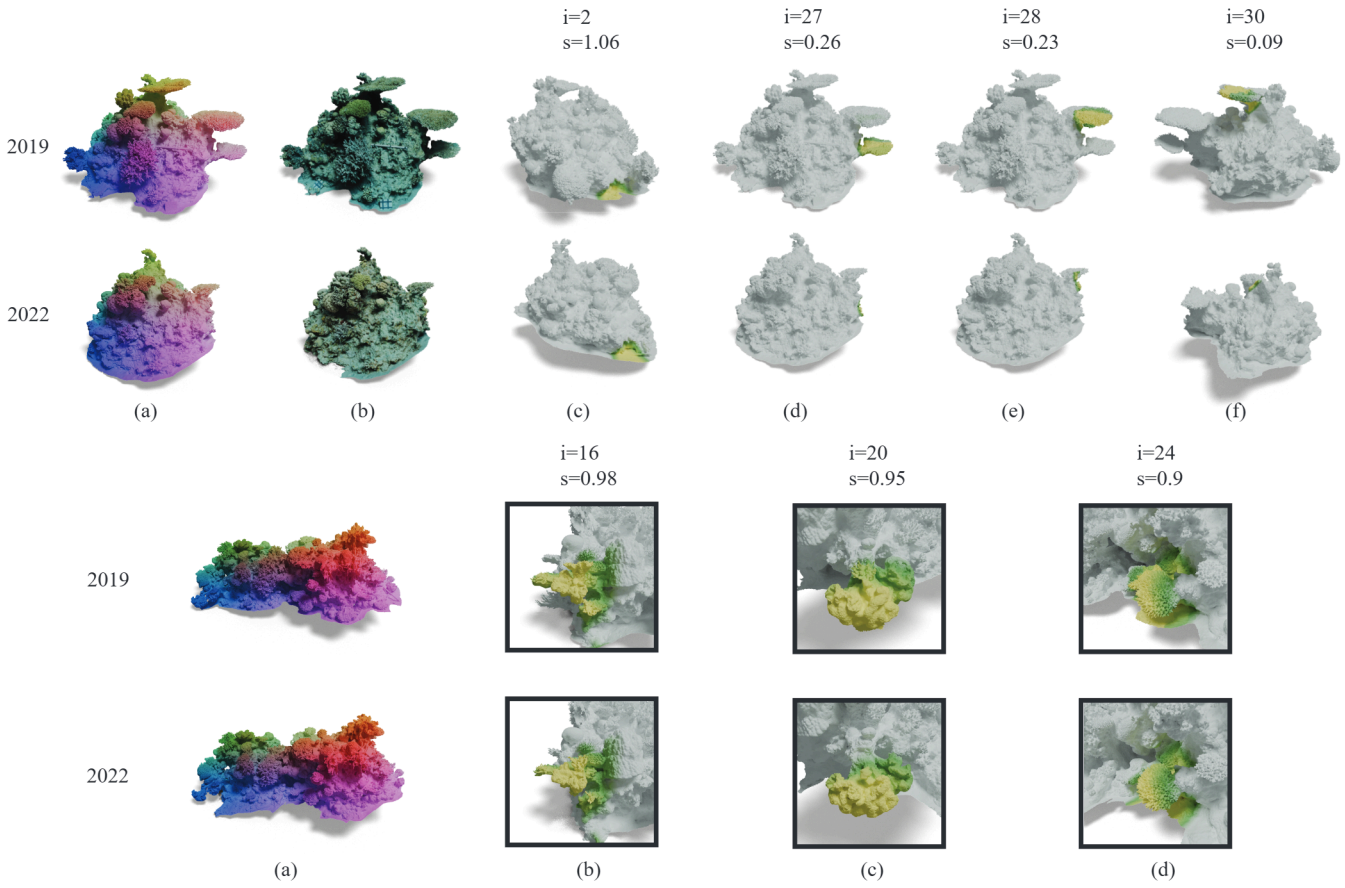

Figure 7: *Examples of visualizations with s values for the model C2 (up) and Kza5m (down) between the years 2019 to 2022. For C2 we show (a) the map (b) the textured models, (c)-(f) s coloring from the SVD on the functional map representing the changes - growth and decay. For Kza5m we show (a) the map, and (b)-(d) zooms of decay changes. See the results section for more details.*

5.5. Analysis of A New Coral Reef Using Our System

We evaluate the performance of our algorithm on the Kza5m coral reef mesh. This model presents significant challenges for manual registration due to its intricate structure, overlapping corals, and high structural complexity. These complexities have deterred expert investigation, and as a result, there is no established ground truth (GT) for this mesh. For testing, we employed three layers in CloudCompare: 2019 and 2020 with texture, and our visualization as a layer indicating coral reef decay. The output of the S-ACORD method includes two data types: 'add' and 'fall.' 'Add' indicates a positive addition, e.g., coral growth, to the mesh in the subsequent year, while 'fall' signifies corals broken off by the storm. We analyzed 20 instances of the 'fall' data ($k = 11$ to $k = 30$, see Figure 14), finding that most of them (19 out of 20) correctly identified actual changes, such as coral colony removal or deformation, thereby facilitating change detection for the user. However, one detection, detection 23 indicated no actual change, which we assume is due to issues with the 3D reconstruction.

To summarize, when testing our method on the Kza5m model, we found that it effectively helps the user to focus on areas of change. This method excels particularly in scenarios involving complete colony removal and also in detecting areas where the mesh is deformed. However, there is a limitation due to the predefined limit on the number of changes ('add' and 'fall') k, set at 30 in our study. For instance, as illustrated in Figure 12, a part of the reef that fell was not marked as a change (false negative) when $k = 30$. It was only when k was increased to 31 that the method identified and marked it as a 'fall' change.

6. Conclusions and Future Work

We presented a system for automatically computing correspondences between 3D models of coral reefs, analysing them and visualizing the main regions of change. Furthermore, the changes are presented in an order of importance, from largest to smallest. We have shown that our registration surpasses a closely related alternative when compared to ground truth generated manually. In addition, we showed that our approach finds the regions of interest that were marked by humans, as well as additional regions of interest of smaller scale that were missed by them.

Further, our algorithm enables the analysis of models that are too

intricate for a human observer to study manually, such as Kza5m. In this model, there are over 700 objects which make it very hard for a biologist to inspect each coral. Our method performed well on this model, and provided a reliable and helpful output regarding changes in that reef site over time, that were manually validated.

Our approach does not consider the texture image at all, although it is an important signal that can help overcome lower resolution captures. Furthermore, corals have various morphologies: branching corals are most easily detected via geometrical analysis while encrusting corals, generally, are better visible in the texture. As a result, combined with the lower resolution capture of the 2020 models, our algorithm missed some of the features in the C3 coral. We believe that adding texture to the algorithm is an interesting avenue for future work. Another promising direction is replacing the spectral basis with one which is more appropriate for the type of maps that we are interested in. Specifically, we identify the growth and decay using the surface area, however these can be perhaps also identified volumetrically. Future work may take into account more extrinsic geometry by working with other functional bases, e.g. those which take the volume into account [WBCPS18] or those that incorporate crease information [HSA*23]. We also plan to explore the applicability of our algorithm to other complex structures, such as detecting changes in cortical surfaces. This would extend the utility of our approach from coral reef analysis to broader applications in medical imaging and biological research.

We believe there is much more that can be done with ecological 3D data in general, and reef analysis in particular. On one hand, real-world biological and ecological data is usually diverse and challenging, leading to new problems in graphics and geometry processing. On the other hand, as underwater data collection means have improved, the large bottleneck now is data analysis. Climate change is causing rapid ecosystem changes while manual data analysis does not scale. Thus detecting these changes while they are happening is difficult, which delays response. We hope that our approach will inspire more cross-disciplinary collaborations, encouraging further work on the important topic of automatic coral reef analysis, using the new emerging data and the wide variety of geometry processing algorithms.

Acknowledgments. Matan Yuval was supported by the Data Science Research Center at the University of Haifa, the Murray Foundation for student research, and the Rohr Family Foundation. Naama Alon-Borissiouk and Mirela Ben Chen acknowledge the support of the Israel Science Foundation (grant No. 1073/21). We thank Ahmad Nasikun for helpful discussions. We also thank Hsueh-Ti Derek Liu for his Blender Toolbox, used for the visualizations throughout the paper.

References

[ABK15] AFLALO Y., BREZIS H., KIMMEL R.: On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences 8*, 2 (2015), 1141–1160. 3

[AFDG*09] ALVAREZ-FILIP L., DULVY N. K., GILL J. A., CÔTÉ I. M., WATKINSON A. R.: Flattening of caribbean coral reefs: region-wide declines in architectural complexity. *Proceedings of the Royal Society B: Biological Sciences 276*, 1669 (2009), 3019–3025. 2

[Agi18] AGISOFT L.: Agisoft photoscan professional (version 1.2. 6)(software). *Professional Edition* (2018). 3

[BD17] BURNS J., DELPARTE D.: Comparison of commercial structure-from-motion photogrammety software used for underwater three-dimensional modeling of coral reef environments. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42*, 2/W3 (2017), 127–131. 2

[BM92] BESL P. J., MCKAY N. D.: Method for registration of 3-d

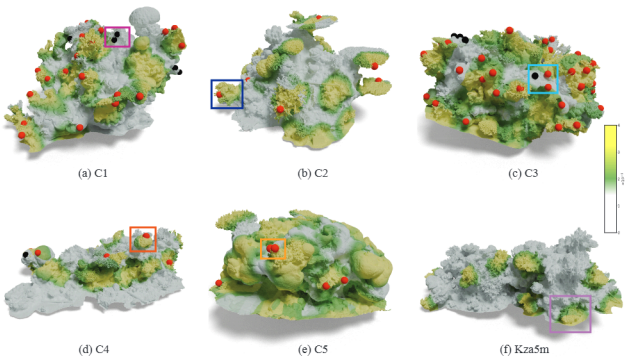

Figure 8: *Comparison of our results with GT human analysis. We show the 2019 Models colored using our algorithm, showing decay ($s_i < 1$) with respect to the corresponding 2020 models. The colored areas indicate regions where the support of the singular vector (i.e., non-zero regions) is localized. The spheres mark the GT points identifying changes that were detected by humans. Model (f) does not have a GT due to its size and complexity. Red spheres mark true positive and black mark false negative. Colored boxes focus on examples that are zoomed in Figure 9.*

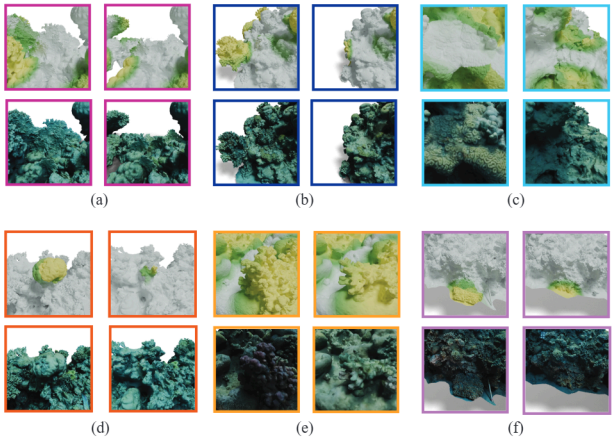

Figure 9: *Zoom on the changes from Figure 8, including true positives (b,d,e,f) and false negatives (a,c). Each group of 4 images shows a coral from 2019 (top left), and its decay in 2020 (top right). We also show the textured models in 2019 (bottom left) and 2020 (bottom right). Each example corresponds to a different model in Figure 8 with matching color and letter. For example, zoom (e) corresponds to the marked orange box in Model C5.*

shapes. In *Sensor fusion IV: control paradigms and data structures* (1992), vol. 1611, Spie, pp. 586–606. 2, 3

[BR*83] BRADBURY R. R., REICHELT R. R., ET AL.: Fractal dimension of a coral reef at ecological scales. *Marine Ecology Progress Series-pages: 10: 169-171* (1983). 2

[CCC*08] CIGNONI P., CALLIERI M., CORSINI M., DELLEPIANE M., GANOVELLI F., RANZUGLIA G.: MeshLab: an Open-Source Mesh Processing Tool, 2008. doi:10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136. 4

[CLJL20] CHEN H., LIU H.-T. D., JACOBSON A., LEVIN D. I. W.: Chordal decomposition for spectral coarsening. *ACM Trans. Graph. 39*, 6 (Nov. 2020). URL: https://doi.org/10.1145/3414685.3417789, doi:10.1145/3414685.3417789. 3

[Clo22] CLOUDCOMPARE G.: Cloudcompare, 2022. URL: http://www.cloudcompare.org/. 3

[DYDZ22] DENG B., YAO Y., DYKE R. M., ZHANG J.: A survey of non-rigid 3d registration. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 559–589. 2

[FBB*16] FERRARI R., BRYSON M., BRIDGE T., HUSTACHE J., WILLIAMS S. B., BYRNE M., FIGUEIRA W.: Quantifying the response of structural complexity and community composition to environmental change in marine communities. *Global change biology 22*, 5 (2016), 1965–1975. 2, 3

[FGRL*22] FERRARI R., GONZALEZ-RIVERO M., LEON J. X., BURNS J. H., FIGUEIRA W. F., SANDIN S. A., DAVIES A. J.: Advances in 3d habitat mapping of marine ecosystem ecology and conservation, 2022. 2

[FPW10] FRIEDMAN A., PIZARRO O., WILLIAMS S. B.: Rugosity, slope and aspect from bathymetric stereo image reconstructions. In *OCEANS'10 IEEE SYDNEY* (2010), IEEE, pp. 1–9. 2

[GN13] GRAHAM N. A., NASH K. L.: The importance of structural complexity in coral reef ecosystems. *Coral reefs 32*, 2 (2013), 315–326. 2

[GRE*23] GAO M., ROETZER P., EISENBERGER M., LÄHNER Z., MOELLER M., CREMERS D., BERNARD F.: Sigma: Scale-invariant global sparse shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 645–654. 2

[HBB*03] HUGHES T. P., BAIRD A. H., BELLWOOD D. R., CARD M., CONNOLLY S. R., FOLKE C., GROSBERG R., HOEGH-GULDBERG O., JACKSON J. B., KLEYPAS J., ET AL.: Climate change, human impacts, and the resilience of coral reefs. *science 301*, 5635 (2003), 929–933. 1

[HBB*17] HUGHES T. P., BARNES M. L., BELLWOOD D. R., CINNER J. E., CUMMING G. S., JACKSON J. B., KLEYPAS J., VAN DE LEEMPUT I. A., LOUGH J. M., MORRISON T. H., ET AL.: Coral reefs in the anthropocene. *Nature 546*, 7656 (2017), 82–90. 1

[HGB10] HOEGH-GULDBERG O., BRUNO J. F.: The impact of climate change on the world's marine ecosystems. *Science 328*, 5985 (2010), 1523–1528. 1

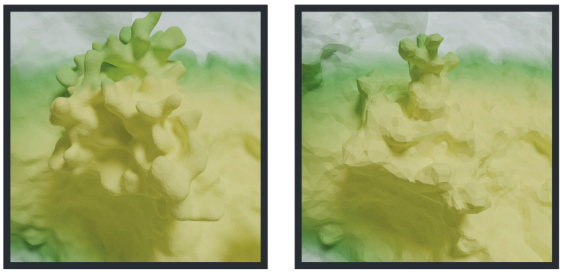

Figure 10: *A true change identified by our algorithm but missed by manual inspection. Model C2 2019 (left) to 2020 (right).*

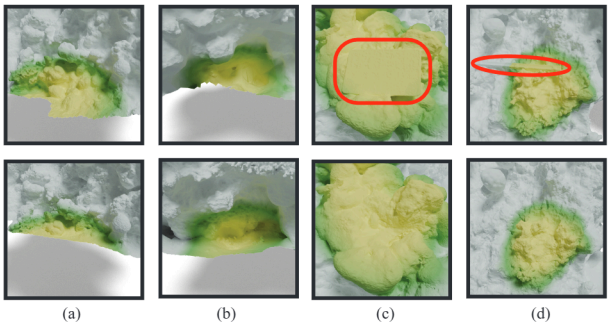

Figure 11: *Outliers: (top row) before, (bottom row) after. False positives due to boundary changes. (a) C3, 2020 to 2022 and (b) C4, 2020 to 2022. We also identify elements that are not part of the structure of the coral reef (circled in red) such as (c) color chart in C5, 2019 to 2022 and (d) scale bar in C2, 2019 to 2022.*

[HSA*23] HARTWIG F., SASSEN J., AZENCOT O., RUMPF M., BEN-CHEN M.: An elastic basis for spectral shape correspondence. In *ACM SIGGRAPH 2023 Conference Proceedings* (2023), pp. 1–11. 10

[JLS94] JONES C. G., LAWTON J. H., SHACHAK M.: Organisms as ecosystem engineers. *Ecosystem management* (1994), 130–147. 2

[LFM*24] LECHENE M. A., FIGUEIRA W. F., MURRAY N. J., ASTON E. A., GORDON S. E., FERRARI R.: Evaluating error sources to improve precision in the co-registration of underwater 3d models. *Ecological Informatics* (2024), 102632. 2

[LGC*23] LIU H.-T. D., GILLESPIE M., CHISLETT B., SHARP N., JACOBSON A., CRANE K.: Surface simplification using intrinsic error metrics. *ACM Transactions on Graphics 42*, 4 (2023). 3

[LJO19] LIU H.-T. D., JACOBSON A., OVSJANIKOV M.: Spectral coarsening of geometric operators. *ACM Trans. Graph. 38*, 4 (July 2019). URL: https://doi.org/10.1145/3306346.3322953, doi:10.1145/3306346.3322953. 3

[LLT*20] LESCOAT T., LIU H.-T. D., THIERY J.-M., JACOBSON A., BOUBEKEUR T., OVSJANIKOV M.: Spectral mesh simplification. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 315–324. 3

[LMHMM*22] LANGE I. D., MOLINA-HERNÁNDEZ A., MEDELLÍN-MALDONADO F., PERRY C. T., ÁLVAREZ-FILIP L.: Structure-from-motion photogrammetry demonstrates variability in coral growth within colonies and across habitats. *PLoS One 17*, 11 (2022), e0277546. 2, 3

[LP20] LANGE I. D., PERRY C. T.: A quick, easy and non-invasive

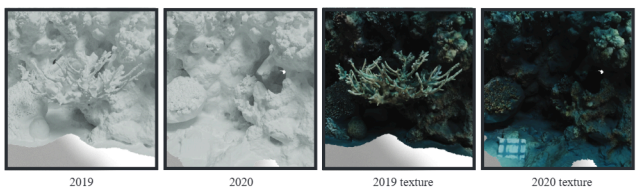

Figure 12: *Undetected "fall" change (false negative) in Kza5m mesh between 2019 to 2020 with $k = 30$. The left images show the area that was not highlighted by our algorithm, and the right images displays it with texture. The detection of the change was successful for $k > 30$.*

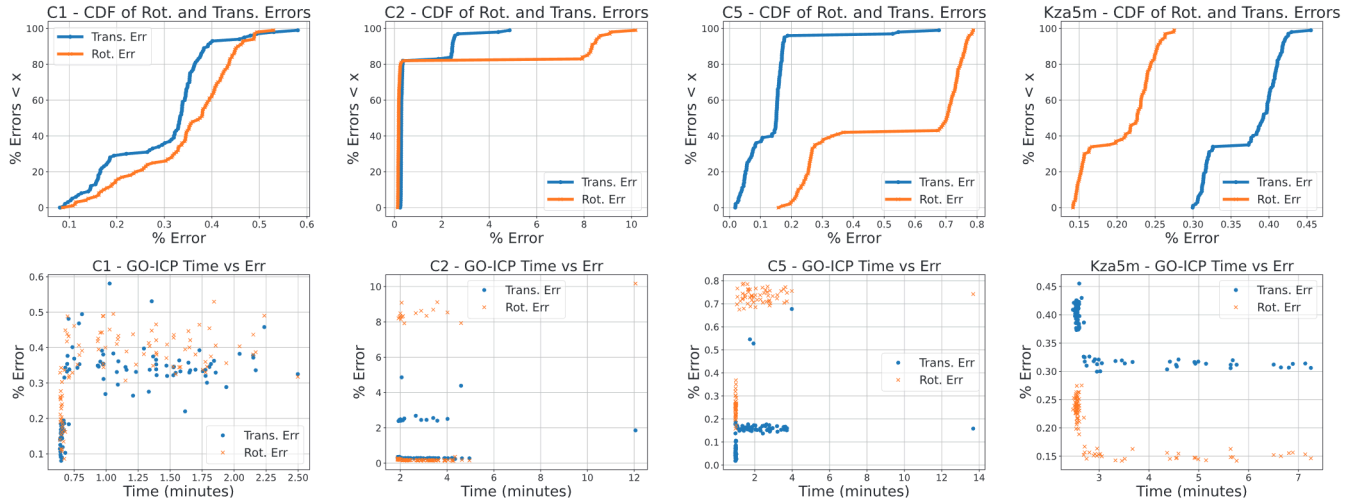

Figure 13: *Comparison of 100 runs on each GT model - C1,C2,C5 and Kza5m. First row is the cumulative distribution of rotation and translation errors, second row the correlation of GO-ICP timing and the error percentage ([M_1 bbx diagonal] for translation error and [%π] for rotation).*

method to quantify coral growth rates using photogrammetry and 3d model comparisons. *Methods in Ecology and Evolution 11*, 6 (2020), 714–726. 2, 3

[MAM14] MELLADO N., AIGER D., MITRA N. J.: Super 4pcs fast global pointcloud registration via smart indexing. In *Computer graphics forum* (2014), vol. 33, Wiley Online Library, pp. 205–215. 2

[MBRM25] MAGGIOLI F., BAIERI D., RODOLÀ E., MELZI S.: Rematching: Low-resolution representations for scalable shape correspondence. In *European Conference on Computer Vision* (2025), Springer, pp. 183–200. 3

[MCPM25] MARIN R., CORONA E., PONS-MOLL G.: Nicp: neural icp for 3d human registration at scale. In *European Conference on Computer Vision* (2025), Springer, pp. 265–285. 2

[MO23] MAGNET R., OVSJANIKOV M.: Scalable and efficient functional map computations on dense meshes. In *Computer Graphics Forum* (2023), vol. 42, Wiley Online Library, pp. 89–101. 3, 6

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: spectral upsampling for efficient shape correspondence. *ACM Trans. Graph. 38*, 6 (Nov. 2019). URL: https://doi.org/10.1145/3355089.3356524, doi:10.1145/3355089.3356524. 6

[MSPS22] MCCARTHY O. S., SMITH J. E., PETROVIC V., SANDIN S. A.: Identifying the drivers of structural complexity on hawaiian coral reefs. *Marine Ecology Progress Series 702* (2022), 71–86. 2

[NBH18] NASIKUN A., BRANDT C., HILDEBRANDT K.: Fast approximation of laplace-beltrami eigenproblems. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 121–134. 3, 4, 5

[OBCCG13] OVSJANIKOV M., BEN-CHEN M., CHAZAL F., GUIBAS L.: Analysis and visualization of maps between shapes. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 135–145. 1, 2, 3, 4, 5, 7

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG) 31*, 4 (2012), 1–11. 2, 3

[PCS*15] POMERLEAU F., COLAS F., SIEGWART R., ET AL.: A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics 4*, 1 (2015), 1–104. 2

[RBSW17] REICHERT J., BACKES A. R., SCHUBERT P., WILKE T.: The power of 3d fractal dimensions for comparative shape and structural complexity analyses of irregularly shaped organisms. *Methods in Ecology and Evolution 8*, 12 (2017), 1650–1658. 2

[ROA*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 1–12. 3, 4

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace–beltrami spectra as 'shape-dna'of surfaces and solids. *Computer-Aided Design 38*, 4 (2006), 342–366. 3

[TCL*12] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics 19*, 7 (2012), 1199–1217. 2

[VL08] VALLET B., LÉVY B.: Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 251–260. 3

[WBCPS18] WANG Y., BEN-CHEN M., POLTEROVICH I., SOLOMON J.: Steklov spectral geometry for extrinsic shape analysis. *ACM Transactions on Graphics (TOG) 38*, 1 (2018), 1–21. 10

[YDRE17] YOUNG G., DEY S., ROGERS A., EXTON D.: Cost and time-effective method for multi-scale measures of rugosity, fractal dimension, and vector dispersion from coral reef 3d models. *PloS one 12*, 4 (2017), e0175341. 2

[YLCJ15] YANG J., LI H., CAMPBELL D., JIA Y.: Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence 38*, 11 (2015), 2241–2254. 1, 2, 3, 6

[YPT*23] YUVAL M., PEARL N., TCHERNOV D., MARTINEZ S., LOYA Y., BAR-MASSADA A., TREIBITZ T.: Assessment of storm impact on coral reef structural complexity. *Science of The Total Environment 891* (2023), 164493. 2, 3, 7

[YS23] YAZGAN M., SAHILLIOĞLU Y.: A partition based method for spectrum-preserving mesh simplification. *IEEE Transactions on Visualization and Computer Graphics* (2023). 3

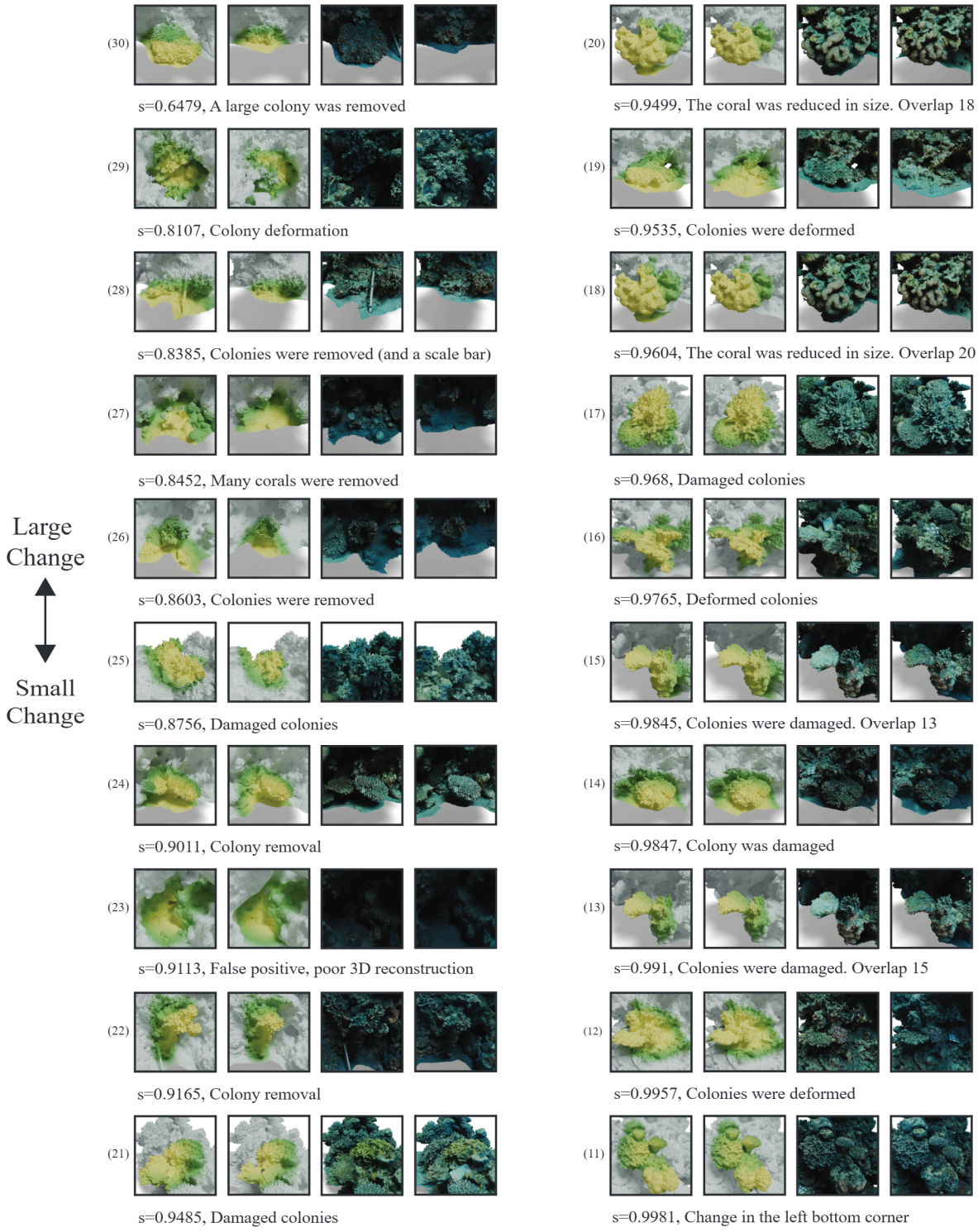

Figure 14: *Evaluation of 'fall' detections in the Kza5m coral mesh. We assessed* 20 *'fall' detections (when* $s < 1$*), arranging the images from the detection with the largest singular value (detection* 30*) indicating the most significant decay change, to the smallest (detection* 11*) where s is very close to 1. For each detection, the display starts with the detection number, followed by four images (in a row): the first two images are from 2019 and 2020, respectively, with highlights indicating detected changes, and the subsequent two images show the original texture from 2019 and 2020. Below each set of images, the singular value and a biologist's evaluation of the detection are displayed. We can see that the algorithm detected* 19 *out of* 20 *changes correctly with only one false positive that was caused by the 3D reconstruction.*