



## Computational Topology - Project 2

---

# Lakes and islands

---

Žiga Lesar  
Iztok Oder  
Matija Rezar

Junij 2015

## 1 Introduction

Maps accessible via internet (notably the Google maps) allow us to quickly access a map of any region of the world. Using all possible maps it is of interest to develop tools that will recognise certain patterns of our focus.

For our project we developed a method based on persistent homology for analysing 2D images of maps. Our approach relies on detecting holes in a simplicial complex built on top of an image of a map. We use the Vietoris-Rips (VR) complex to generate a simplified model of the image, which we then analyse using standard topological tools. The method is capable of identifying connected components and circular patterns based on the connectivity in the VR complex. Preprocessing plays an important role in our method, so we tested different techniques for segmentation and morphological transformations.

For the purpose of the project we have designed an application with a GUI, which simplifies the use of our method and also enables less technologically aware people to harness its power for their own needs.

We implemented our method in Java using the JavaPlex library for homology computation and Swing for the GUI.

## 2 Method

Every analysis process consists of three main steps: preprocessing, building the VR complex and homology computation. The application is built in such a way that the workflow roughly follows these three steps: users are interested mainly in the homology information, which they obtain by building and analysing the simplicial complex on top of a segmented image. Each step is briefly described in the following sections.

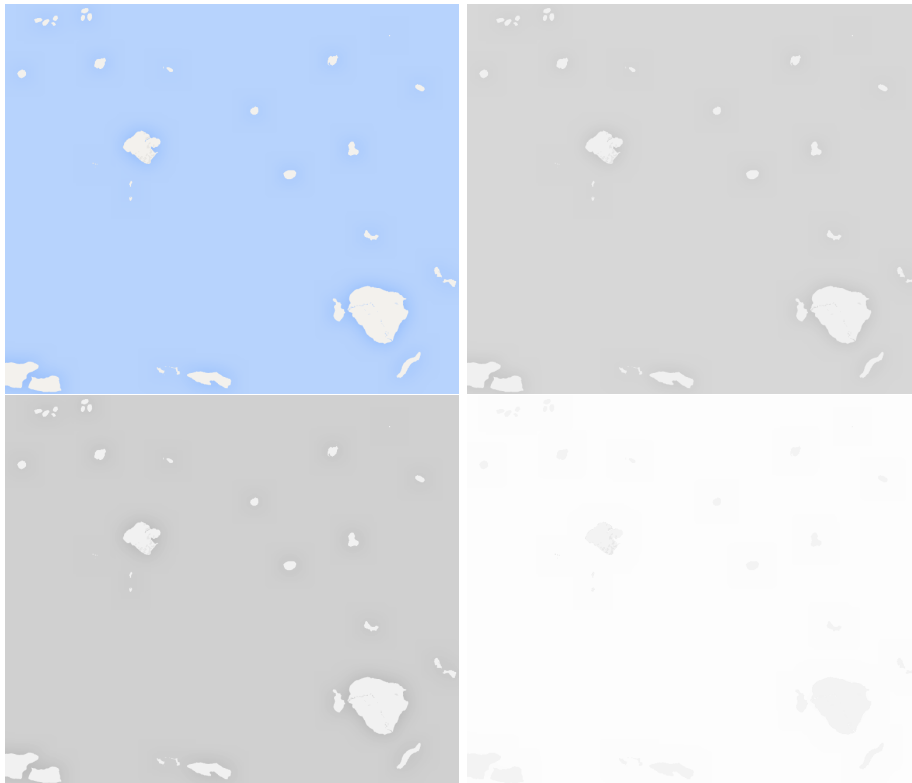


Figure 1: Conversion to grayscale. Original image (top left), conversion with intensity (top right), conversion with luma (bottom left) and conversion with value (bottom right).

## 2.1 Image preprocessing

We implemented several different preprocessing techniques frequently used in practice. Our main goal was to build a set of preprocessing elements which, when combined, would enable users to create more sophisticated functions suitable for their needs.

The preprocessing elements are divided into three categories: grayscale conversion, segmentation, and morphological transformations.

Some segmentation methods require the image to be in grayscale, so color images have to be converted before the segmentation step. This can be done in various ways - we chose three: *intensity*, *value* and *luma*. Intensity (the value  $I$  in the HSI color space) can be expressed as  $(R + G + B)/3$ , value (the value  $V$  in the HSV color space) is the maximum component and can be expressed as  $\max\{R, G, B\}$ , and luma is a certain linear combination of RGB values, arising from a physical quantity *luminance*, which is computed as  $0.2126R + 0.7152G + 0.0722B$ . The three grayscale conversion methods can be seen in figure 1.

When the image is converted to grayscale, we can employ a segmentation method to separate the foreground (e.g. lakes, islands) from the background (e.g. sea, land). Segmentation represents an enormous field of research in com-

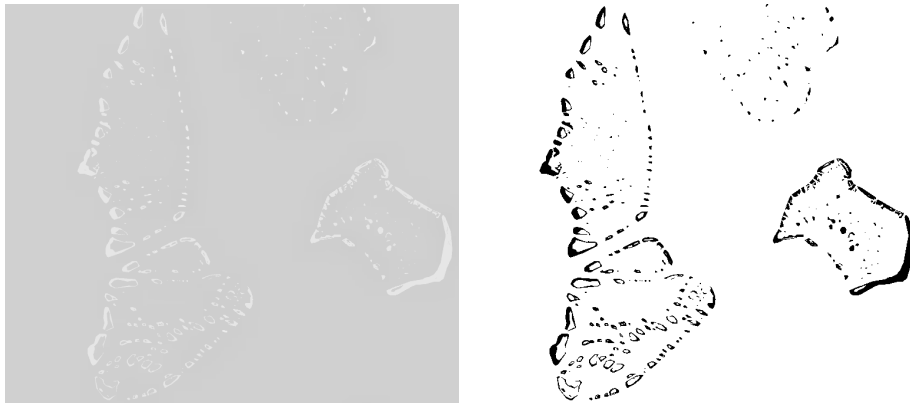


Figure 2: An image before and after thresholding. In this example the image was converted to grayscale based on luma and Otsu’s method was used for thresholding. Also, the resulting image was inverted, so the foreground is now correctly represented with black pixels.

puter vision and image processing, but since we don’t need such accurate results, we implemented two basic methods: simple thresholding and Otsu thresholding [1]. In the former case we compute the average value across the whole image and then set the threshold at that value. In the latter case finding the thresholding value is a bit more sophisticated. Otsu’s method finds a threshold value that maximizes the inter-class variance between foreground and background pixels. An example of thresholding is shown in figure 2

We can enhance the segmented image using a set of morphological transformations, namely dilation, erosion, opening and closing. These transformations can be used as a simple noise reduction step before or after computing the homology. An application of morphological opening to determining region connectivity can be seen in figure 3.

We also implemented a simple technique for removing noisy features that cannot be removed with morphological transformations without seriously affecting other parts of the image. The technique finds all homogeneous regions in the image with their respective surface areas and then removes regions with a small enough surface area, which are probably due to noise in the input image. This threshold is also user-controlled to enable users to control the amount of noise reduction.

## 2.2 Building the Vietoris-Rips complex

A properly preprocessed and segmented image is ready to be used as a basis for VR complex construction. A straightforward approach would be to put a vertex on every foreground pixel of the input image, but it turns out that this is too computationally expensive and thus intractable in practice. Putting a vertex on every  $k$ -th pixel effectively reduces the amount of computation required to capture relevant features in the image. In our application the user can control the parameter  $k$ .

The resulting complex is used to compute a filtration based on the parameter  $\delta$  of the VR complex. The user provides a parameter which controls the

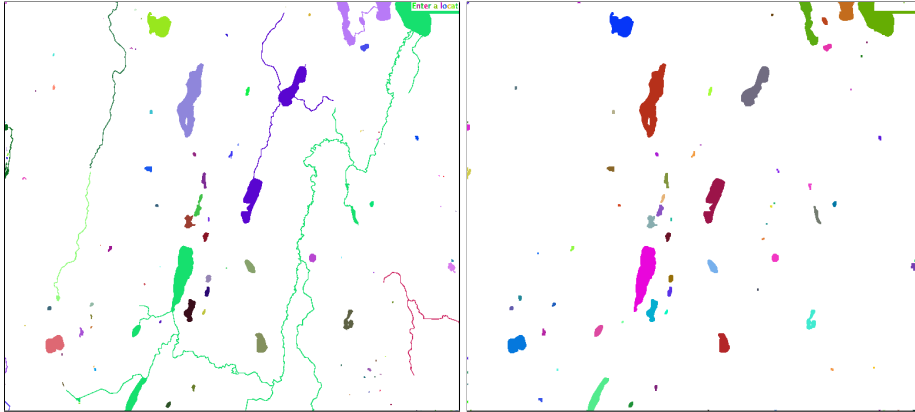


Figure 3: An image before and after morphological opening. As shown by this example morphological transformations can be used to remove unwanted features as well as extracting useful information (the difference in region connectivity indicates that some regions were connected prior to this transformation).

maximum  $\delta$  for the filtration. If a low enough value is used, the parameter  $k$  can be made smaller. When for example  $\delta = \sqrt{2}$  only neighbouring pixels are connected in the filtration - this, in combination with  $k = 1$  is used to accurately count islands and lakes. In other situations, where we also need to take into consideration neighbouring islands,  $\delta$  must be larger, but to reduce computation time  $k$  must also be larger.

We can extract relevant information out of the computed filtration. In particular, we are interested in the generators of homology groups in dimensions 0 and 1, since those represent connected components and cycles. Only generators with a long lifetime are considered, since those with a short lifetime are probably due to noise.

### 3 Results

We tested our methods on a set of images with lakes and islands. Our goal was to count the elements (lakes or islands) in the image and detect circular patterns formed by some of the elements. While counting the elements could be done with simpler methods, we were hoping to get similar results with topological methods. Apart from just looking at generators of the 0-th homology group we were also interested in generators of the 1-st homology group, since this one carries information about such circular patterns we were looking for. The images with extracted information are shown in the following figures.

### 4 Conclusion

In this report we have presented an approach to image analysis based on persistent homology. We have designed an application with a GUI to be used as a basic analysis tool in a streamlined fashion. Our methods perform well and are also highly customizable and thus can also be used on low-end hardware.

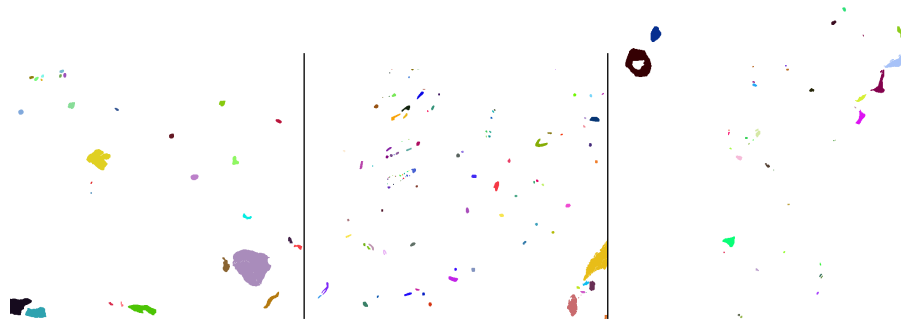


Figure 4: Counting the islands by examining the 0+th homology group. Every island is colored differently. The left image contains 29 islands, the middle image contains 115 islands and the right image contains 41 islands.

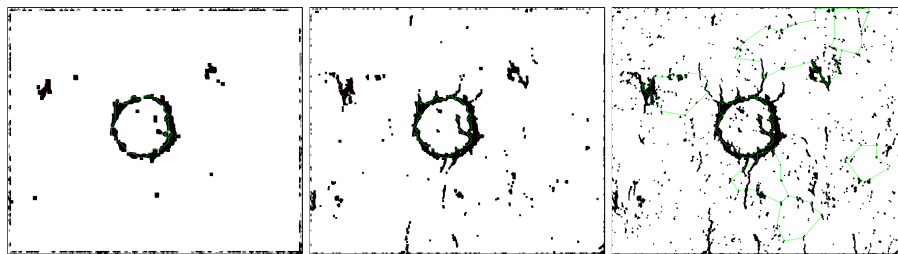


Figure 5: An example showing the impact of preprocessing on the final result. In this example the same image was preprocessed with three different sizes of the morphological opening kernel: large (left), medium (middle), and small (right). Since this significantly affects connectivity in the VR complex we also get very different homology groups. Of the above images only the left one produces expected results.

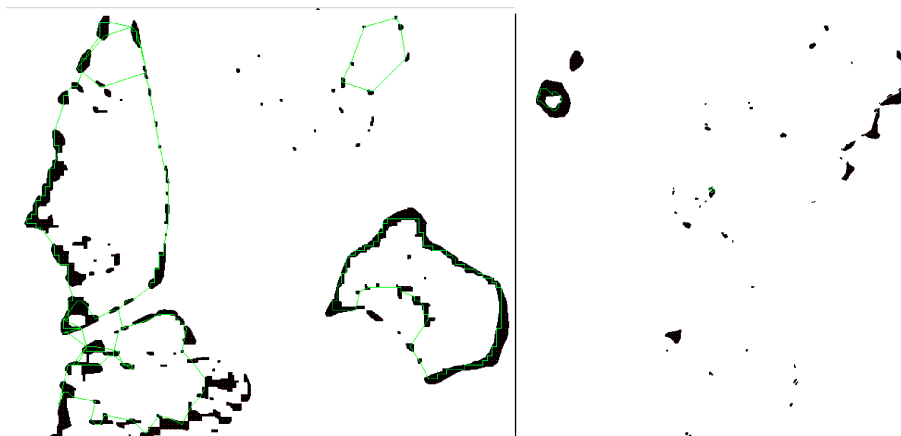


Figure 6: The first homology group contains information about circular patterns that we're interested in. The generators of homology are shown in green.



Figure 7: By looking at the components to which the vertices of the cycles belong we can also detect lakes on islands: if the vertices belong to the same component, the cycle is a lake. The generators of homology are shown in green.

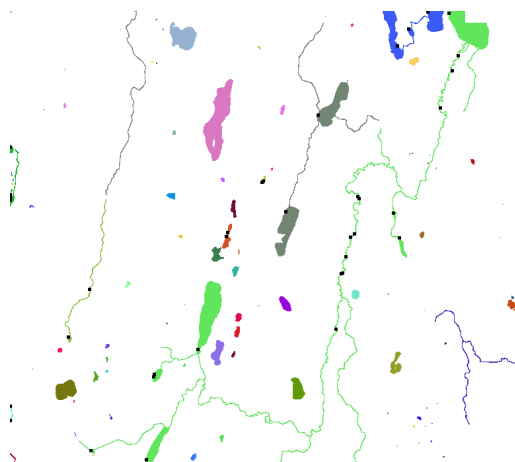


Figure 8: Detecting lake systems and lakes connected by rivers. In this example there are 97 lake systems and 26 lakes in those systems are part of a bigger system composed of several lakes connected by rivers.

Our methods are capable of topologically analysing structures in images, such as lakes, islands and their connectivity. Using persistent homology has turned out to be a good option giving expected results within a short amount of time. However, since the results are highly dependent on the preprocessing, the whole analysis would require enormous amounts of work to automate. With a careful consideration of possible input images and fine tuning of the parameters in our method we could automate the process to a certain degree, but probably never to an extent at which it would be usable in practice.

## A Division of work

Žiga Lesar developed the preprocessing code and the GUI, Iztok Oder and Matija Rezar dealt with homology computation.

## References

- [1] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.