Case study: Understanding topology of small image patches

Tao Hou, University of Oregon

Understanding topology of small image patches

This application is documented in the following papers

- G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, On the local behavior of spaces of natural images, International Journal of Computer Vision, (76), 1, 2008, pp. 1-12.
- Carlsson, Gunnar. "Topology and data." Bulletin of the American Mathematical Society 46.2 (2009): 255-308.

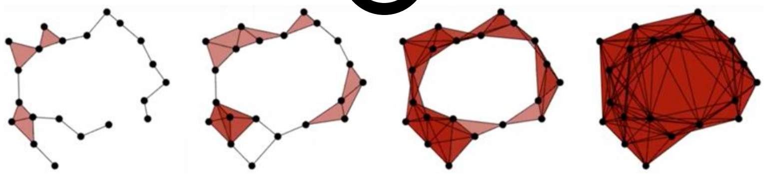
Persistent homology pipeline

1

Point cloud
(image patches)

Build Filtration

2



Čech/Rips Filtration

Persistent Homology

3

barcode

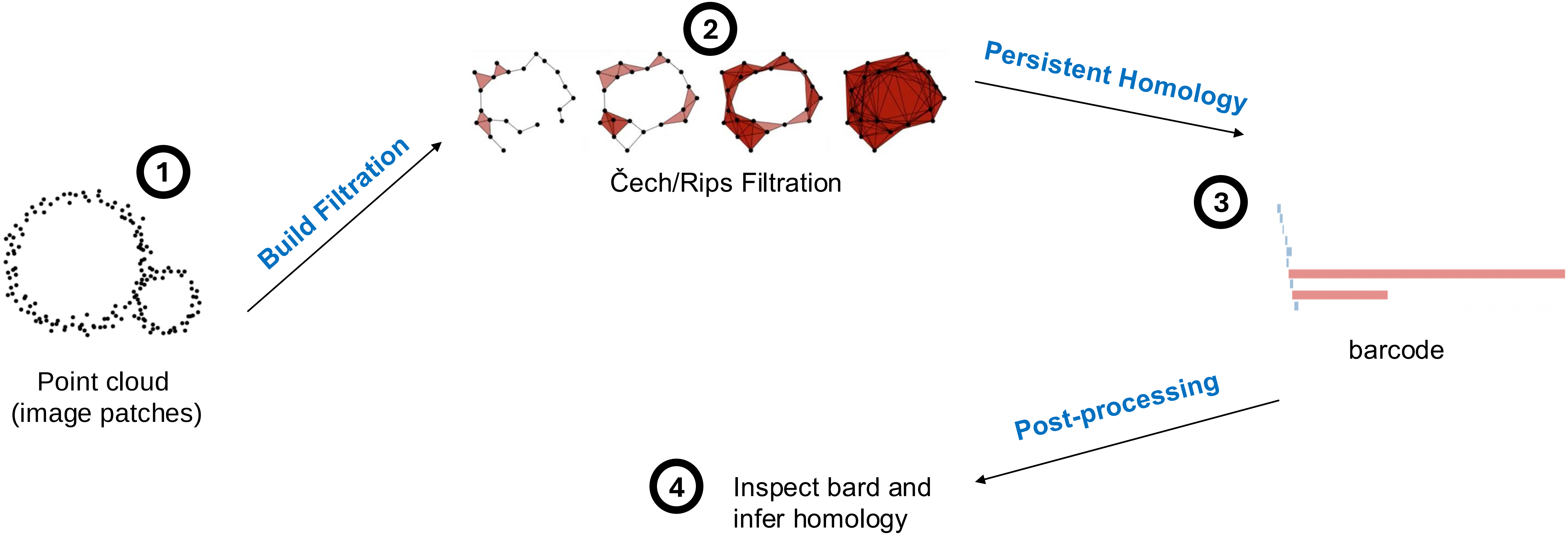
Post-processing

4

Inspect bard and
infer homology

Some img from: AATRN

Persistent homology pipeline



This paper indeed adopts an “iterative” process, aka they repeat the above processes several times to refine their inference

Some img from: AATRN

Background

- An image (assuming gray-scale) taken with a digital camera consists of a number (0-255) attached to each pixel
- It can be considered as a point in $\mathbb{R}^p$, where p is the number of pixels (e.g., $p = 16 \times 16 = 256$)

Background

- An image (assuming gray-scale) taken with a digital camera consists of a number (0-255) attached to each pixel
- It can be considered as a point in $\mathbb{R}^p$, where p is the number of pixels (e.g., $p = 16 \times 16 = 256$)
- Now an image is a point in $\mathbb{R}^p$. What about the collection of all possible natural images lying in $\mathbb{R}^p$? What does it look like?

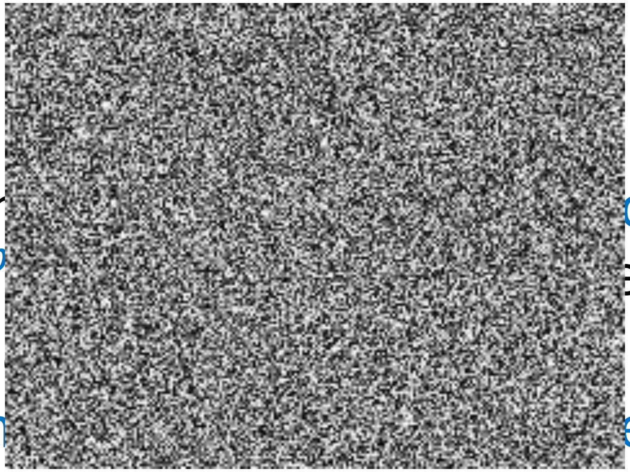
Background

- An image (assuming gray-scale) taken with a digital camera consists of a number (0-255) attached to each pixel
- It can be considered as a point in $\mathbb{R}^p$, where p is the number of pixels (e.g., $p = 16 \times 16 = 256$)
- Now an image is a point in $\mathbb{R}^p$. What about the collection of all possible natural images lying in $\mathbb{R}^p$? What does it look like?
    1. On one hand , one could conjecture that the collection of all natural images is a very high dimensional subspace in $\mathbb{R}^p$, since images are capable of expressing a very wide variety of scenes

Background

- An image (assuming gray-scale) taken with a digital camera consists of a number (0-255) attached to each pixel
- It can be considered as a point in $\mathbb{R}^p$, where p is the number of pixels (e.g., $p = 16 \times 16 = 256$)
- Now an image is a point in $\mathbb{R}^p$. What about the collection of all possible natural images lying in $\mathbb{R}^p$? What does it look like?
  1. On one hand , one could conjecture that the collection of all natural images is a very high dimensional subspace in $\mathbb{R}^p$, since images are capable of expressing a very wide variety of scenes
  2. On the other hand it could also be possible that the collection of all natural images only occupy a very low dimensional subspace in $\mathbb{R}^p$ (aka. a very tiny portion of it), since random pixel arrays (e.g., “white noise”) typically cannot come close to a natural image

Background

- An image (assuming gray-scale) taken with a digital camera consists of a number (0-255) attached to each pixel
- It can be considered as a  the number of pixels (e.g., $p = 16 \times 16 = 256$)
- Now an image is a point i ollection of all possible natural images lying in $\mathbb{R}^p$ e?
    1. On one hand , one c e collection of all natural images is a very high e in $\mathbb{R}^p$, since images are capable of expressing a very wide variety of scenes
    2. On the other hand it could also be possible that the collection of all natural images only occupy a very low dimensional subspace in $\mathbb{R}^p$ (aka. a very tiny portion of it), since random pixel arrays (e.g., “white noise”) typically cannot come close to a natural image

Img from: Wikipedia

Dataset

- Trying to testify the above hypothesis is difficult since it’s hard to have a database sampling nearly all possible natural images

Dataset

- Trying to testify the above hypothesis is difficult since it's hard to have a database sampling nearly all possible natural images
- The task instead looks at small 3×3 patches of images, for which we could reasonably have a nice sample of all the possibilities

Dataset

- Trying to testify the above hypothesis is difficult since it's hard to have a database sampling nearly all possible natural images
- The task instead looks at small 3×3 patches of images, for which we could reasonably have a nice sample of all the possibilities
- They began with a database of black and white images taken by:
  - J. H. van Hateren and A. van der Schaaf, Independent component filters of natural images compared with simple cells in primary visual cortex

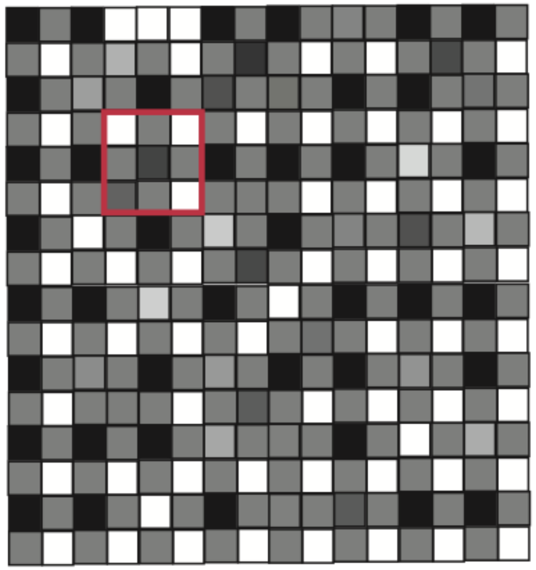
Dataset

- Trying to testify the above hypothesis is difficult since it’s hard to have a database sampling nearly all possible natural images
- The task instead looks at small 3×3 patches of images, for which we could reasonably have a nice sample of all the possibilities
- They began with a database of black and white images taken by:
  - J. H. van Hateren and A. van der Schaaf, Independent component filters of natural images compared with simple cells in primary visual cortex
- The database consisted of images taken around Groningen, Holland, in town and in the surrounding countryside

Dataset

Dataset

- From images from the previous database, they collect 4 millions of 3×3 patches



Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel

   People often compute logarithms of values when processing data because it allows them to transform skewed data into a more normal distribution, making it easier to analyze using statistical methods

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel
   b) Subtract the average of all coordinates from each coordinate. This produces a new vector (patch) whose average is always 0

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel
   b) Subtract the average of all coordinates from each coordinate. This produces a new vector (patch) whose average is always 0

Doing the above equates two patches ***which only differ by the brightness*** *aka. if a patch is obtained from another patch by adding a constant value, i.e. "turning up the brightness", then the two patches will be regarded as the same*

Data preprocessing

Images of the same scene with different brightness

Image from: https://theailearner.com/2019/01/30/what-is-contrast-in-image-processing/

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel
   b) Subtract the average of all coordinates from each coordinate. This produces a new vector (patch) whose average is always 0
   c) Compute "D-norm" of the patch, which measure the contrast

Data preprocessing

Image contrast is the difference in brightness between light and dark parts

Image from: https://pippin.gimp.org/image-processing/chap_point.html

Data preprocessing

Image contrast is the difference in brightness between light and dark parts

Image from: https://pippin.gimp.org/image-processing/chap_point.html

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel
   b) Subtract the average of all coordinates from each coordinate. This produces a new vector (patch) whose average is always 0
   c) Compute "D-norm" of the patch, which measure the contrast
   d) Keep this patch if its D-norm is among the top 20% of all patches taken from the image

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel
   b) Subtract the average of all coordinates from each coordinate. This produces a new vector (patch) whose average is always 0
   c) Compute "D-norm" of the patch, which measure the contrast
   d) Keep this patch if its D-norm is among the top 20% of all patches taken from the image
   e) If kept, normalize the patch by dividing by its D-norm

Data preprocessing

Given each image from the previous database, they do the following:

1. Extract at random 5000 3×3 patches from the image. Regard each patch as a point (vector) in $\mathbb{R}^9$
2. For each patch do the following:
   a) Compute the logarithm of intensity at each pixel
   b) Subtract the average of all coordinates from each coordinate. This produces a new vector (patch) whose average is always 0
   c) Compute "D-norm" of the patch, which measure the contrast
   d) Keep this patch if its D-norm is among the top 20% of all patches taken from the image
   e) If kept, normalize the patch by dividing by its D-norm

A "norm" is a generalization of the ***length*** *of a vector. Divide the vector (patch) by its "length" (D-norm) make it of length 1, so the contrast of all images is always at the same level*

Data preprocessing

- At the end of the process, they get 4 millions of 3×3 patches of high contrast with the same brightness, denoted by M

Data preprocessing

- At the end of the process, they get 4 millions of 3×3 patches of high contrast with the same brightness, denoted by M
- Some further observations:
  - The previous “equating brightness” process (making each patch vector average to 1) essentially “reduces” the dimension of the patched to 8-dimensional

Data preprocessing

- At the end of the process, they get 4 millions of 3×3 patches of high contrast with the same brightness, denoted by M
- Some further observations:
  - The previous "equating brightness" process (making each patch vector average to 1) essentially "reduces" the dimension of the patched to 8-dimensional
  - The previous "normalization process" based on the D-norms make the patches to reside in a "7-dimensional sphere" S^7 within the 8-dimensional space

Data preprocessing

- At the end of the process, they get 4 millions of 3×3 patches of high contrast with the same brightness, denoted by M
- Some further observations:
  - The previous "equating brightness" process (making each patch vector average to 1) essentially "reduces" the dimension of the patched to 8-dimensional
  - The previous "normalization process" based on the D-norms make the patches to reside in a "7-dimensional sphere" S^7 within the 8-dimensional space

For reference, a "1-dimensional sphere" S^1 within the 2-dimensional space is the just the boundary of unit ball (consisting of all points with length 1). Reasoning the above in detail is beyond the scope

Data preprocessing

- At the end of the process, they get 4 millions of 3×3 patches of high contrast with the same brightness, denoted by M
- Some further observations:
  - The previous “equating brightness” process (making each patch vector average to 1) essentially “reduces” the dimension of the patched to 8-dimensional
  - The previous “normalization process” based on the D-norms make the patches to reside in a “7-dimensional sphere” S^7 within the 8-dimensional space
- They then observe that all patches are scattered throughout the 7-sphere, in the sense that no point on the 7-sphere is very far from the each other

Data sampling

- 4 millions points is a large number and build filtrations (say, Rips) and computing PD for it is too costly
- A sampling of the 4 millions points is then necessary
- A straightforward sampling is just to select points randomly.

Data sampling

- 4 millions points is a large number and build filtrations (say, Rips) and computing PD for it is too costly
- A sampling of the 4 millions points is then necessary
- A straightforward sampling is just to select points randomly.
- But this may not be the most efficient way: a more efficient way is to consider **density** of points:

Data sampling

- 4 millions points is a large number and build filtrations (say, Rips) and computing PD for it is too costly
- A sampling of the 4 millions points is then necessary
- A straightforward sampling is just to select points randomly.
- But this may not be the most efficient way: a more efficient way is to consider **density** of points:
  - Points from dense areas tend to have a lot of similar peers and so we should sample less of such points

Data sampling

- 4 millions points is a large number and build filtrations (say, Rips) and computing PD for it is too costly
- A sampling of the 4 millions points is then necessary
- A straightforward sampling is just to select points randomly.
- But this may not be the most efficient way: a more efficient way is to consider **density** of points:
  - Points from dense areas tend to have a lot of similar peers and so we should sample less of such points
  - Points from sparse areas should have a higher precedence because of its scarcity

Data sampling

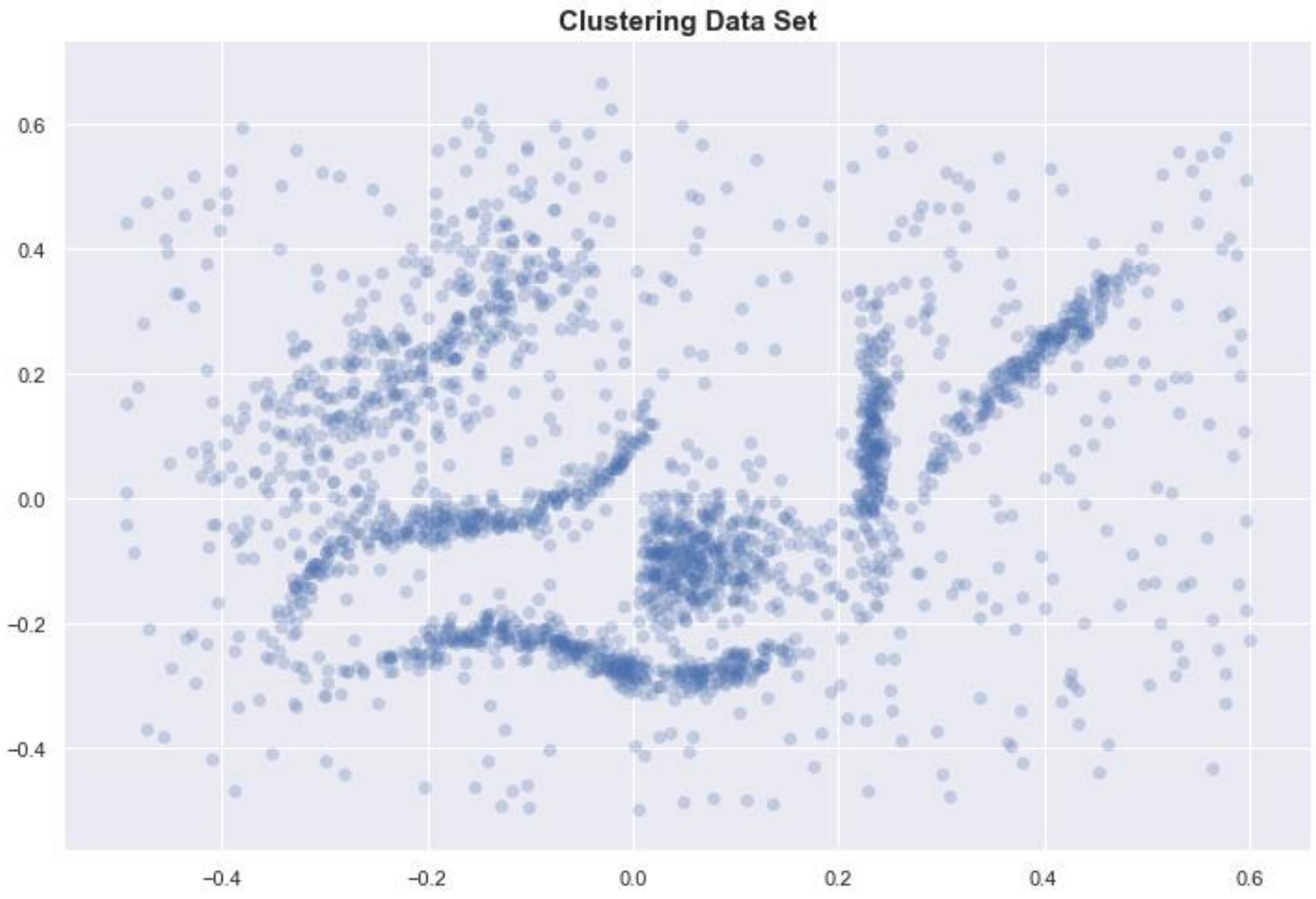


Image from: https://pberba.github.io/stats/2020/07/08/intro-hdbscan

Data sampling

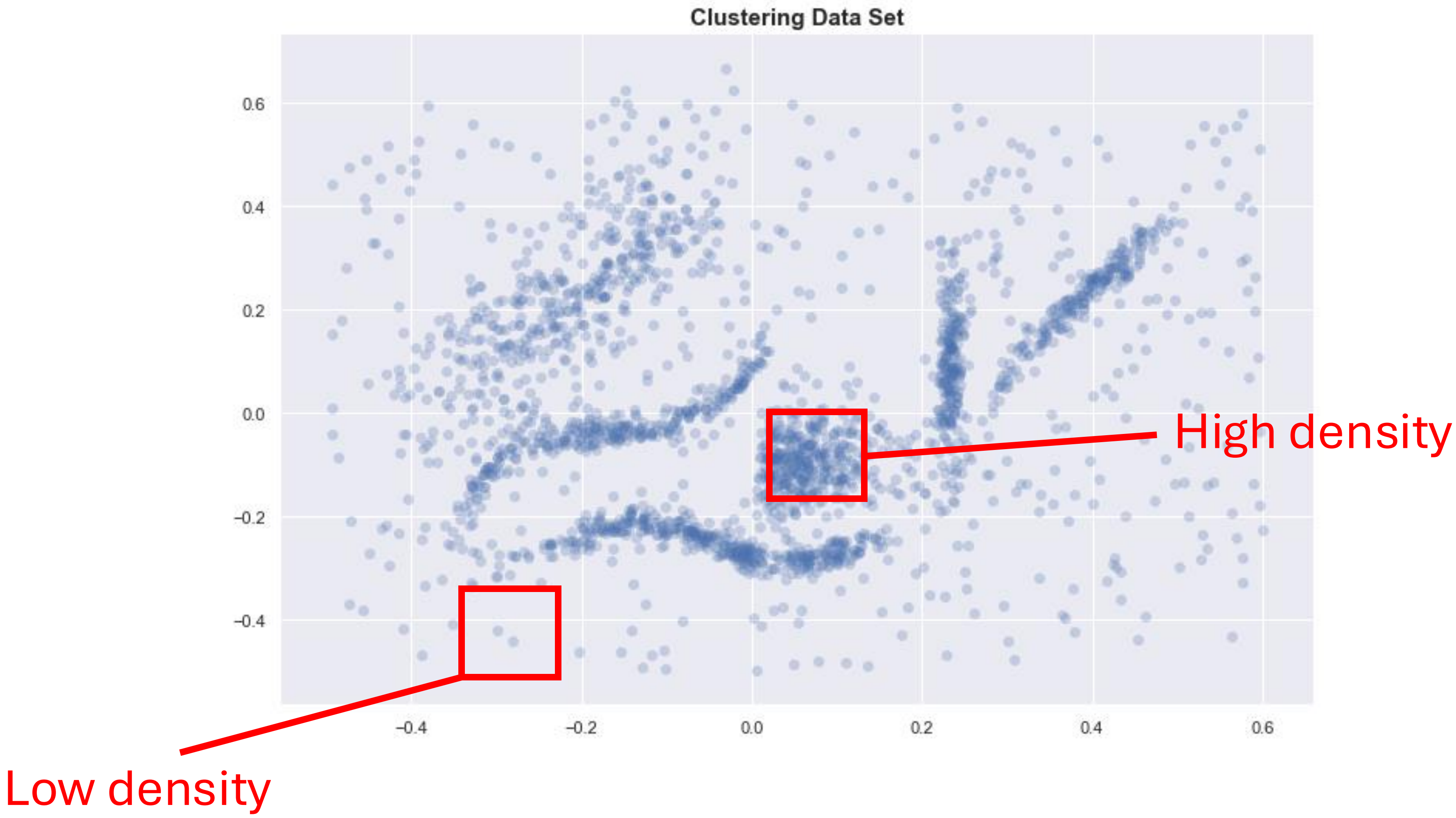


Image from: https://pberba.github.io/stats/2020/07/08/intro-hdbscan

Data sampling

- A common density measure of points is to use the "distance to the k-th nearest neighbor" denoted $\delta_k(p)$ for a point p

Data sampling

- A common density measure of points is to use the "distance to the k-th nearest neighbor" denoted $\delta_k(p)$ for a point p
- They then define a subset $M[k, t]$ of M as:

$$M[k, T] = \{p \in M \mid \delta_k(p) \text{ lies in the } T\% \text{ lowest values among } M\}$$

Data sampling

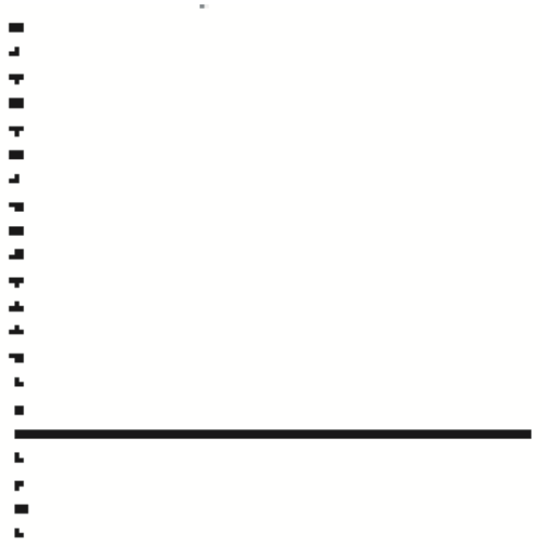
- A common density measure of points is to use the "distance to the k-th nearest neighbor" denoted $\delta_k(p)$ for a point p
- They then define a subset $M[k, t]$ of M as:

$$M[k, T] = \{p \in M \mid \delta_k(p) \text{ lies in the } T\% \text{ lowest values among } M\}$$

- They randomly select a certain number of points (called "landmarks") from a certain $M[k, T]$
- From the landmark points, they then build the Čech/Rips Filtration and compute the persistence barcode for the it

First attempt

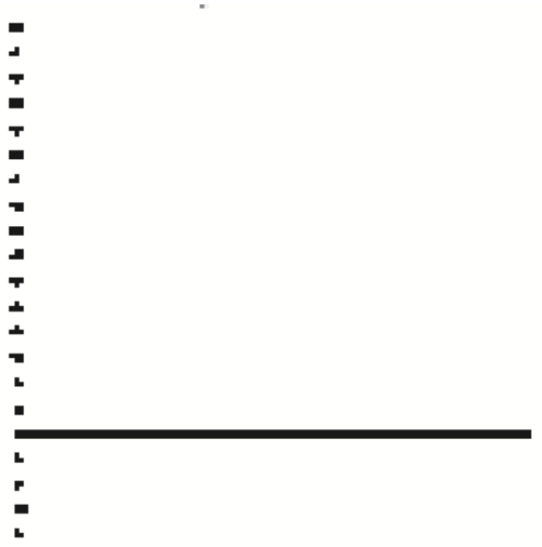
- 1d barcode for 50 landmarks from $M[300,30]$:



$k = 300, T = 30\%$

First attempt

- 1d barcode for 50 landmarks from $M[300{,}30]$:



$k = 300, T = 30\%$

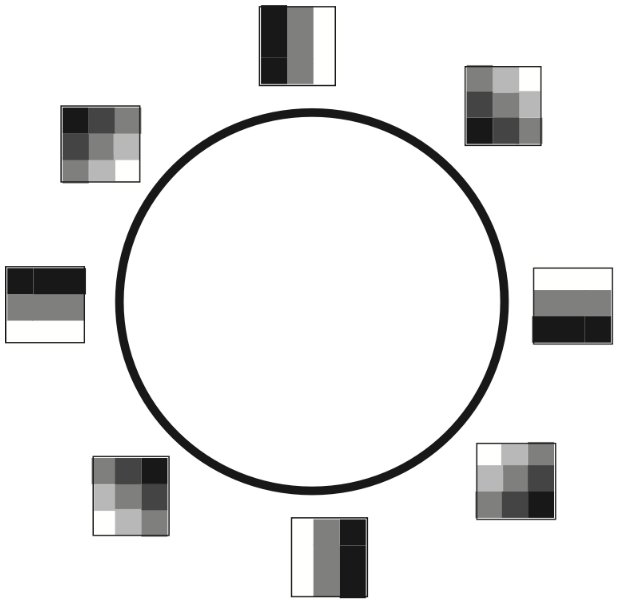
- There are a number of short lines, and one long one
- This suggests that the first Betti number should be estimated to be one, aka. there is a single 1-d cycle (hole) across all patches (points)

First attempt

- The barcode is stable, in the sense that it appears repeatedly in different rounds of sampling
- The simplest possible explanation for this barcode is that the underlying space should be a circle

First attempt

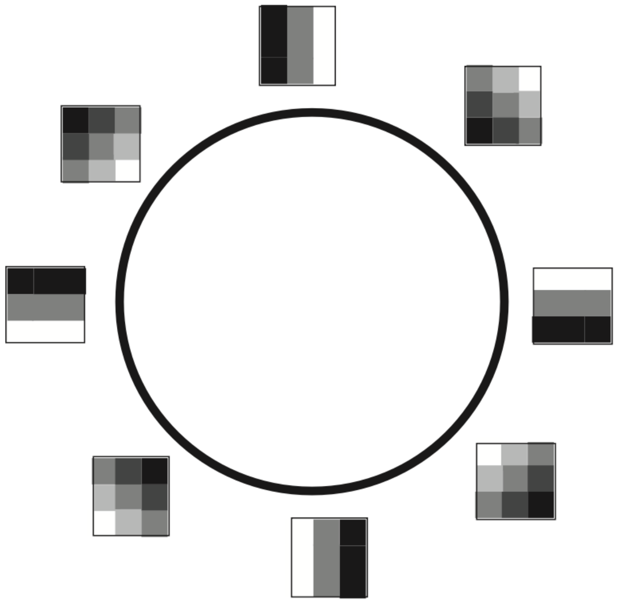
- The barcode is stable, in the sense that it appears repeatedly in different rounds of sampling
- The simplest possible explanation for this barcode is that the underlying space should be a circle



Primary circle

First attempt

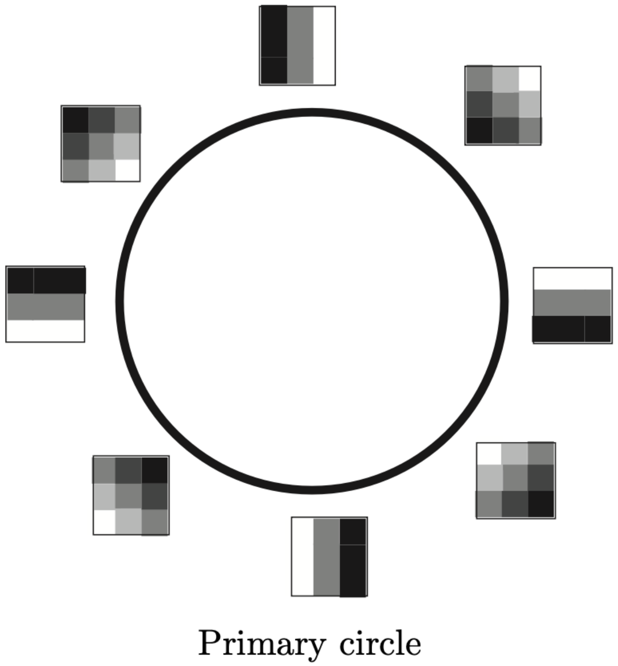
- An explanation for this is that the patches are "variation across a single direction" where the direction rotates which form a cycle

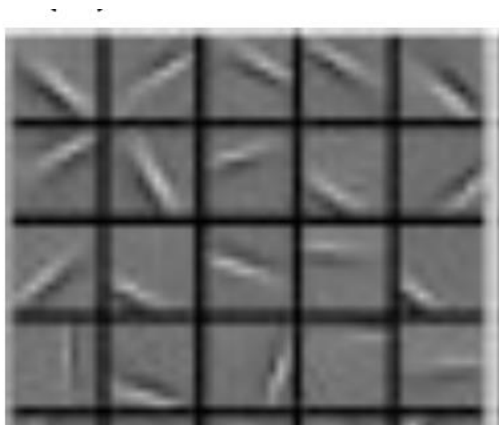


Primary circle

First attempt

- An explanation for this is that the patches are "variation across a single direction" where the direction rotates which form a cycle

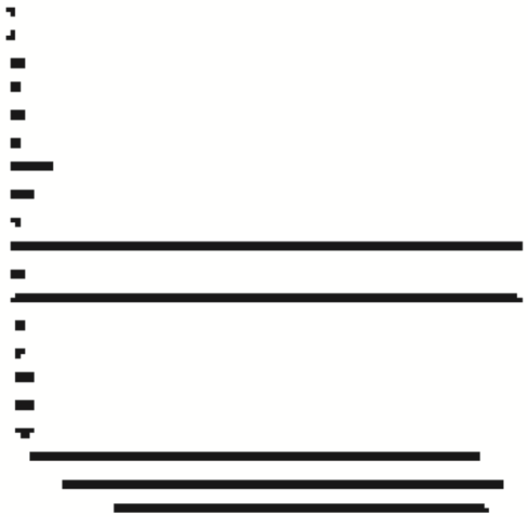




Right image from: van Hateren and van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex

Second attempt

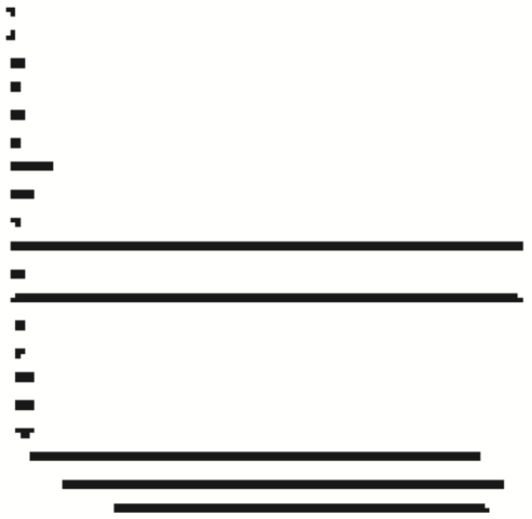
- 1d barcode for 50 landmarks from $M[15,30]$:



$k = 15, T = 30\%$

Second attempt

- 1d barcode for 50 landmarks from $M[15,30]$:

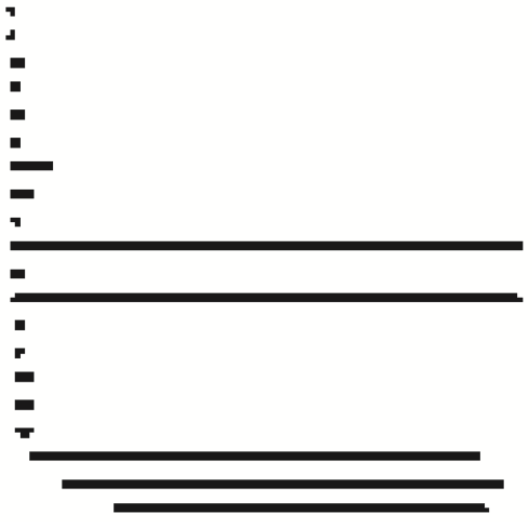


$k = 15, T = 30\%$

- Notice: $\delta_k(p)$ for large k computes density using points in large neighborhoods of p, and for small k uses small neighborhoods
- So, δ_k for large k corresponds to a smoothed out notion of density, and for small k carries more of the detailed structure of the data set

Second attempt

- 1d barcode for 50 landmarks from $M[15,30]$:

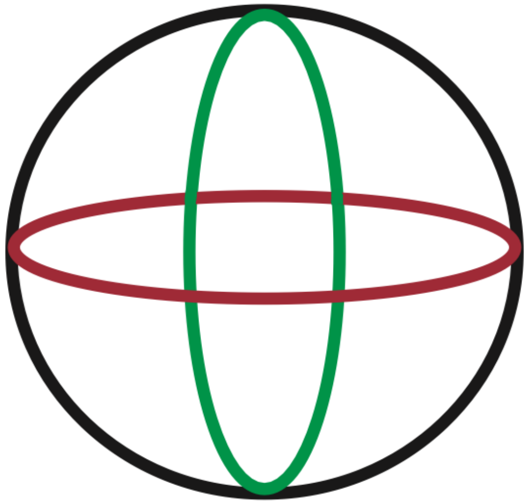


$k = 15, T = 30\%$

- There are a number of short lines, and five long ones, which is stable across different samples
- This suggests that the first Betti number should be estimated to be five

Second attempt

- The most probably space for a first Betti number of five is a space of three cycles (we will not really touch on why)



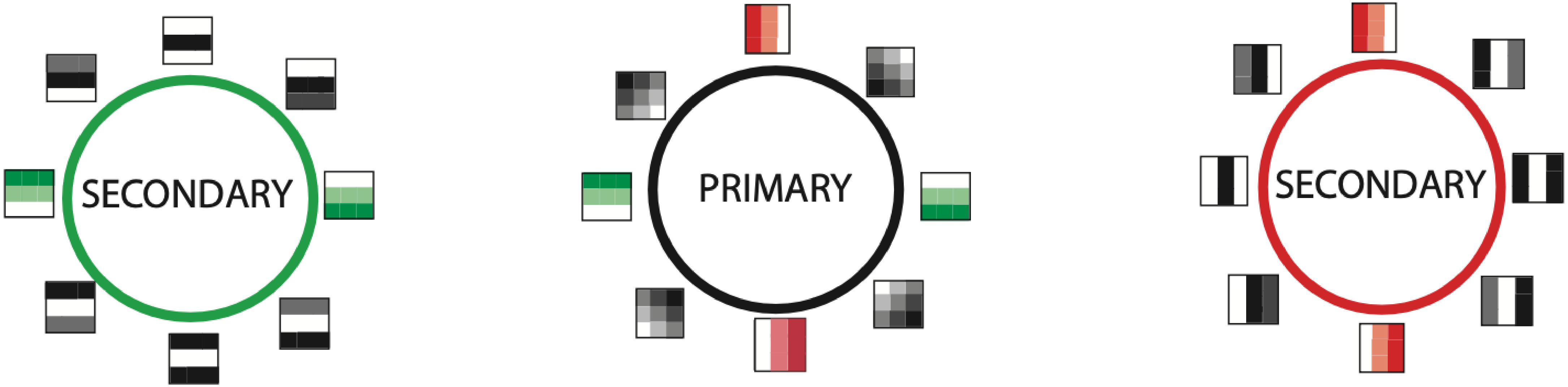
Three circle model

- Notice that the primary black cycle touches both the two secondary cycles (red and green), but the two red and green cycles do not actually touch each other

Second attempt

An explanation:

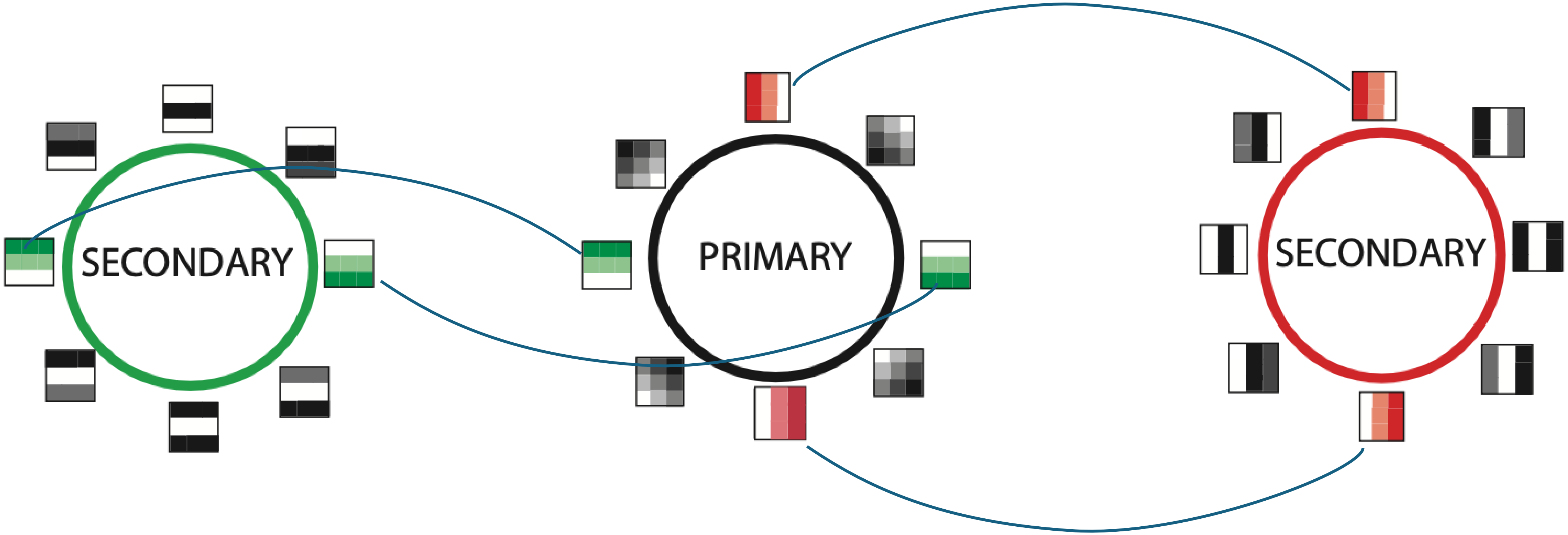
- The primary cycle still corresponds to “variation across a single direction” where the direction rotates
- The two secondary cycles capture vertical or horizonal variations of patches where the variation direction changes horizontally or vertically



Second attempt

An explanation:

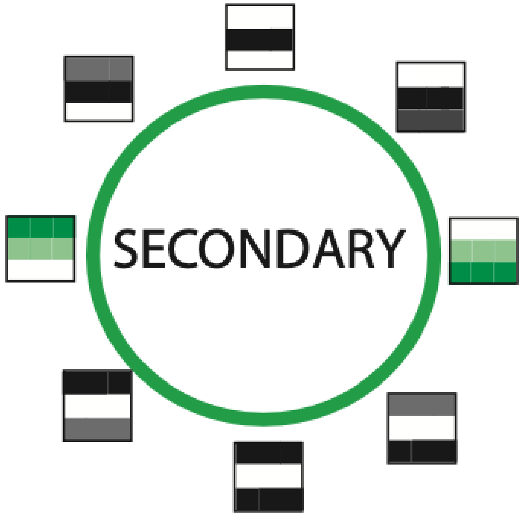
- The primary cycle still corresponds to “variation across a single direction” where the direction rotates
- The two secondary cycles capture vertical or horizonal variations of patches where the variation direction changes horizontally or vertically

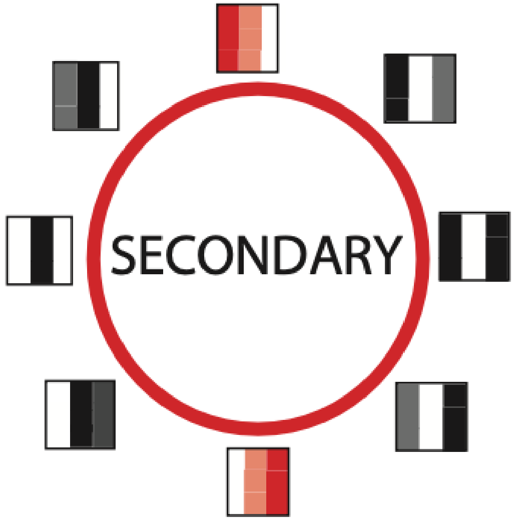


Second attempt

Explanation for vertical and horizontal variations:

1. Nature has this bias, since for example objects aligned in a vertical direction are more stable than those aligned at a 45 degree angle

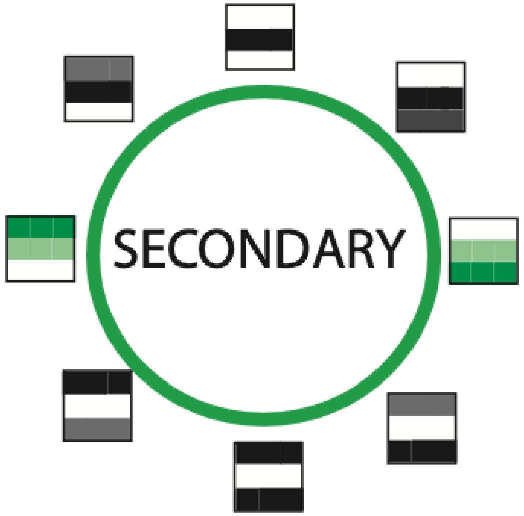


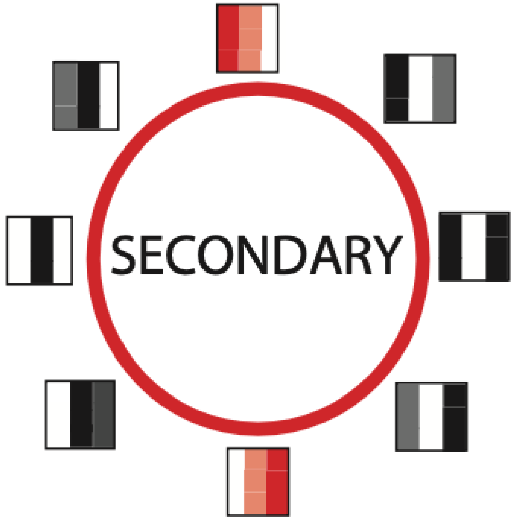


Second attempt

Explanation for vertical and horizontal variations:

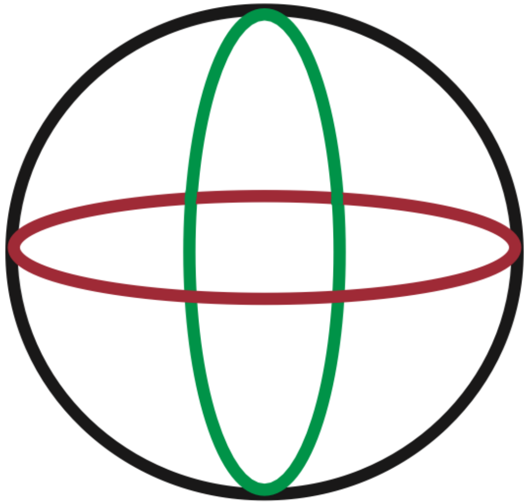
1. Nature has this bias, since for example objects aligned in a vertical direction are more stable than those aligned at a 45 degree angle
2. Another explanation is that this phenomenon is related to the technology of the camera, since the rectangular pixel arrays in the camera have the potential to bias the patches in favor of the vertical and horizontal directions





A further quest

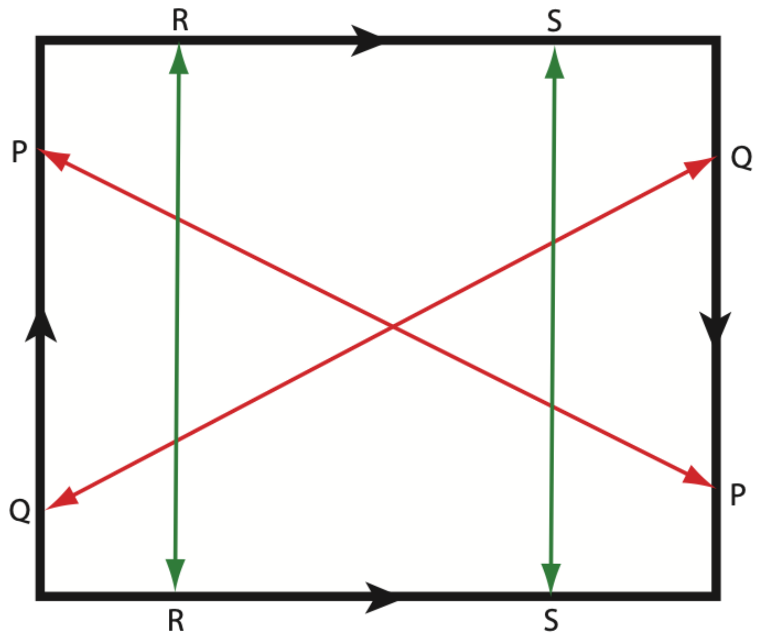
- A question to ask as an extension is that where is the three-cycle-with-two-intersections space lie in?
- Aka. if you fill out the “missing pieces” of the points, what kind of space woud you get?



Three circle model

A further quest

- A natural guess is the famous space in topology world called the "Klein bottle"
- It's a space derived by identifying sides of the following square with
  - Horizontal sides identified normally
  - Vertical sides identified **reversely**



A further quest

- Klein bottle can only be visualized in 4D world but we are in 3D
- The following is a 3D presentation which is not perfect (has self intersections)

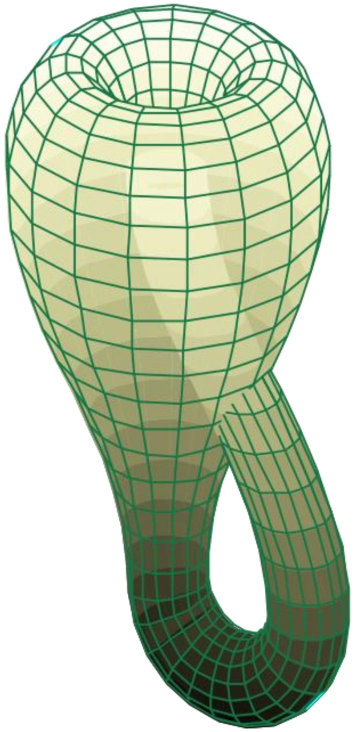


Image from: Wikipedia

A further quest

- The following is an illustration for how to identify the sides of the square to from the bottle (see also: https://plus.maths.org/content/introducing-klein-bottle)

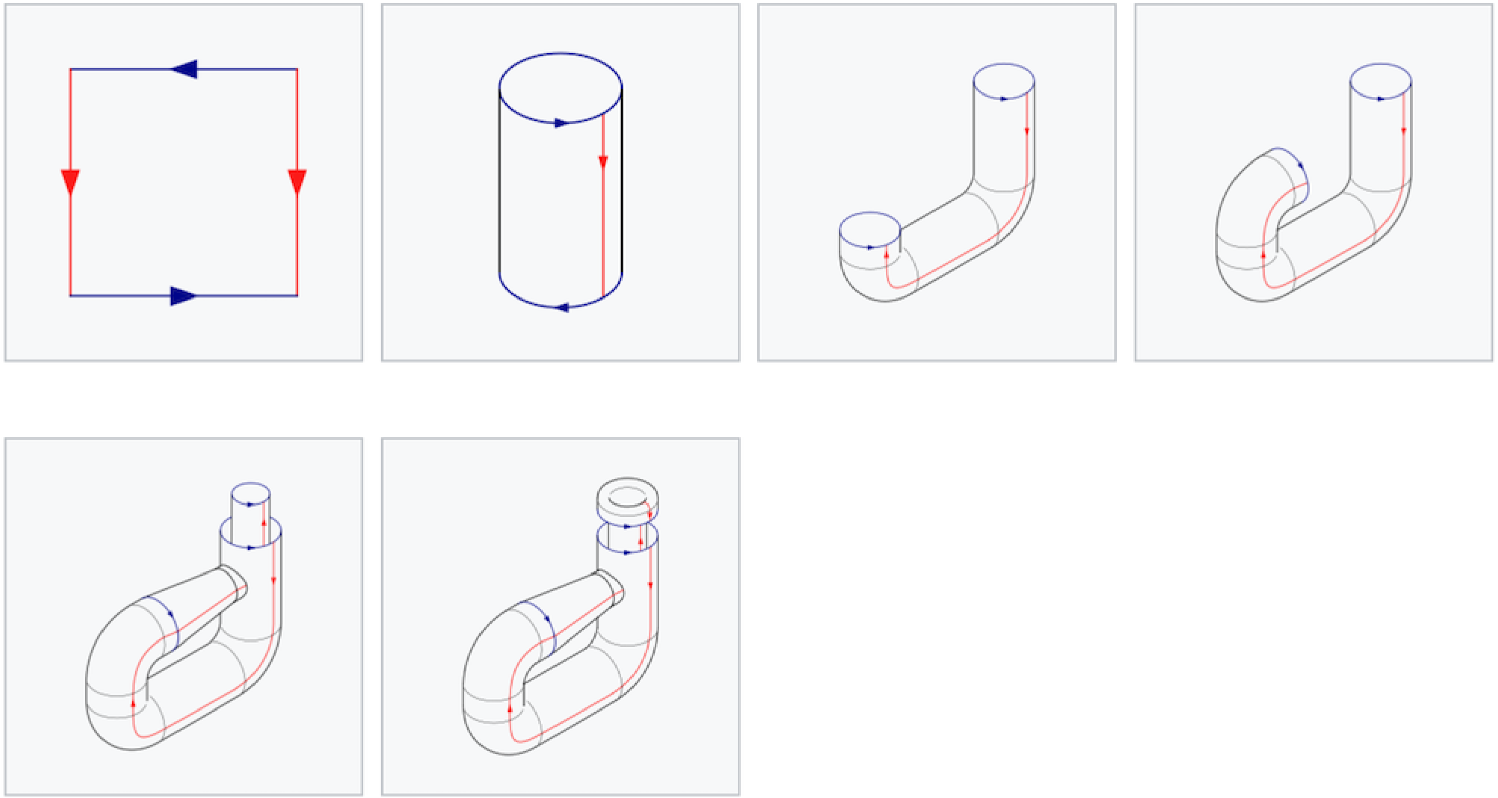
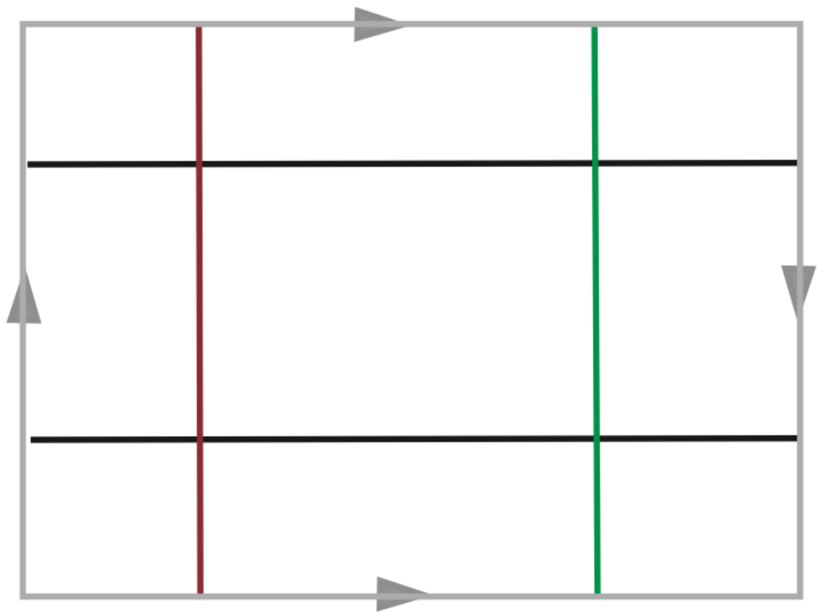


Image from: Wikipedia

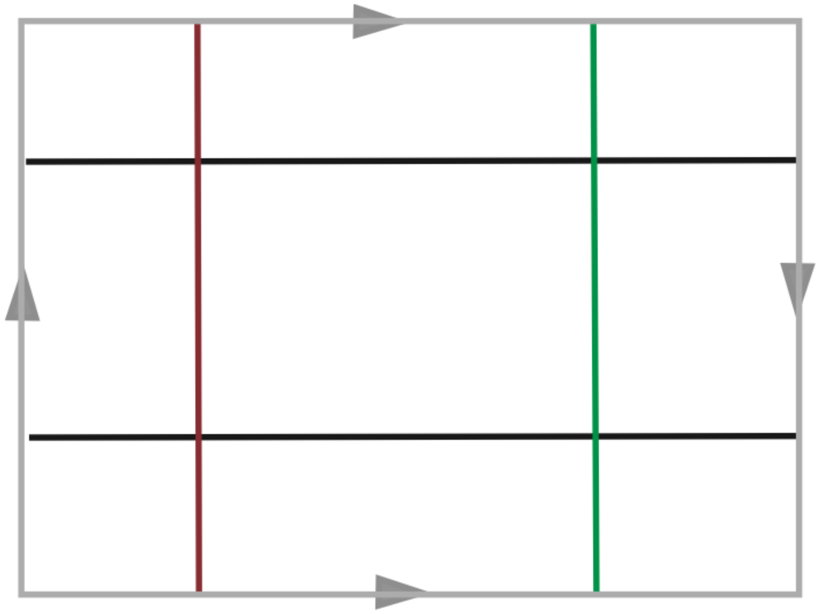
A further quest

- By the identification of the sides, we notice that the the following black lines form a single cycle (the primary one)



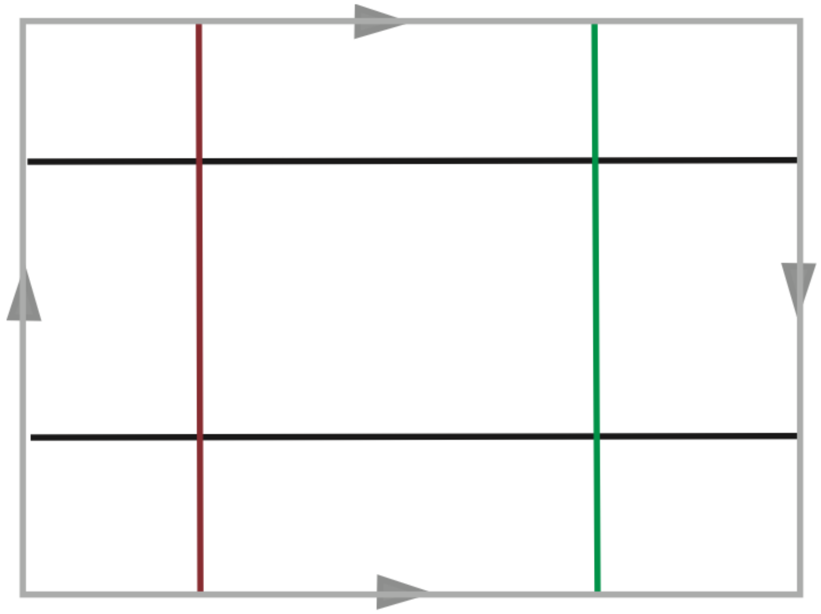
A further quest

- By the identification of the sides, we notice that the the following black lines form a single cycle (the primary one)
- The two red and green lines form two separate cycles



A further quest

- By the identification of the sides, we notice that the the following black lines form a single cycle (the primary one)
- The two red and green lines form two separate cycles
- We also have that black line intersects red and green lines twice while red and green do not intersect (which suits our assumption)



A further quest

- The authors then move on to see whether the could actually “construct” the Klein bottle by the following fact:
  - 0th Betti number: 1
  - 1st Betti number: 2
  - 2nd Betti number: 1

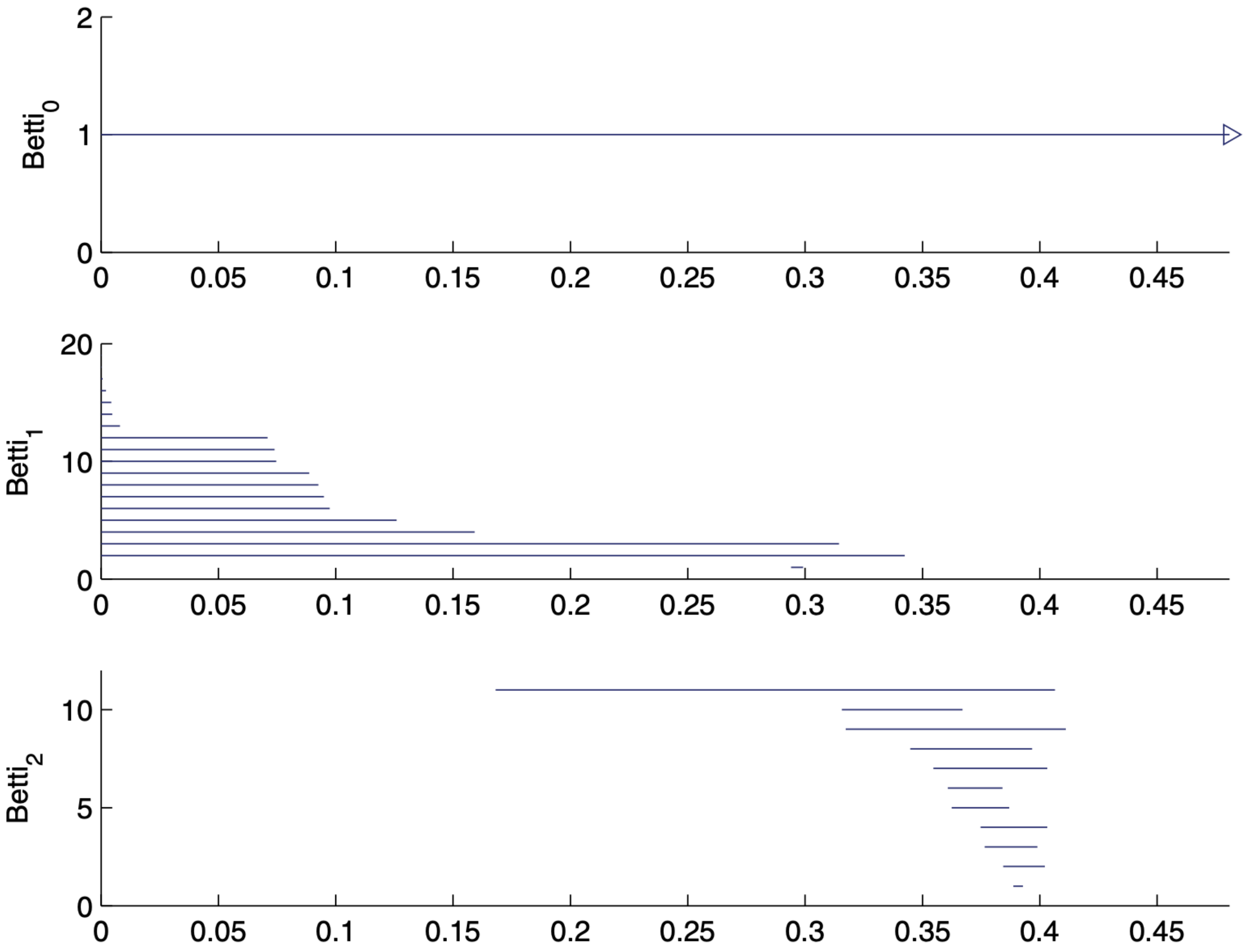
A further quest

- The authors then move on to see whether the could actually “construct” the Klein bottle by the following fact:
  - 0th Betti number: 1
  - 1st Betti number: 2
  - 2nd Betti number: 1
- This means that they must construct landmarks whose barcode reflect these Betti numbers
- However, after several attempts, they could not do this

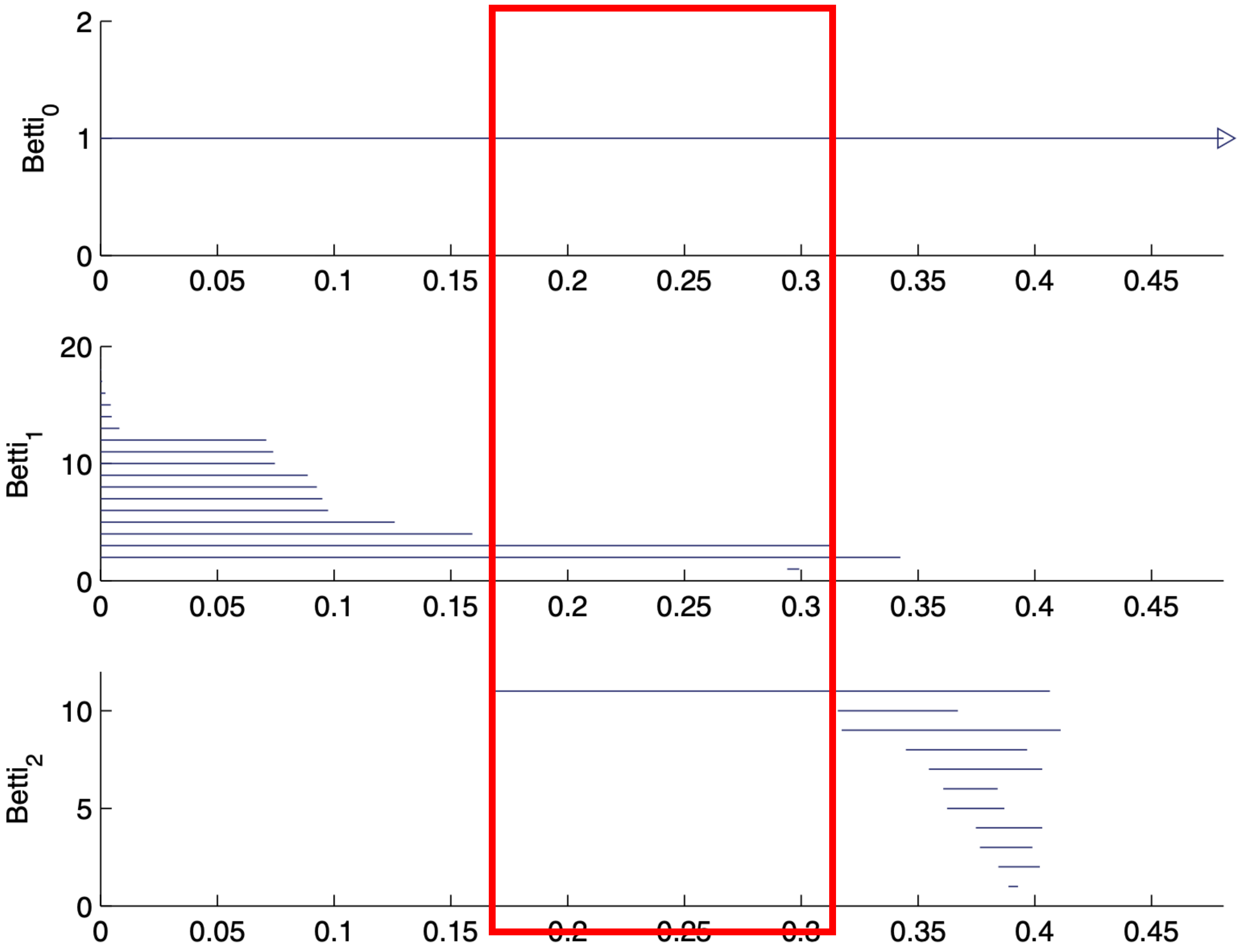
A further quest

- The authors then move on to see whether the could actually “construct” the Klein bottle by the following fact:
  - 0th Betti number: 1
  - 1st Betti number: 2
  - 2nd Betti number: 1
- This means that they must construct landmarks whose barcode reflect these Betti numbers
- However, after several attempts, they could not do this
- The authors suspects there are certain types of patches of missing which make them fail to “construct” the Klein bottle
- After some analysis of the nature of the patches, they find the missing patches.
- After adding them, the barcode of the landmarks finally exhibit the desired property

A further quest



A further quest



Remark

- A question is then: what is the usage of this seemingly wild finding?

Remark

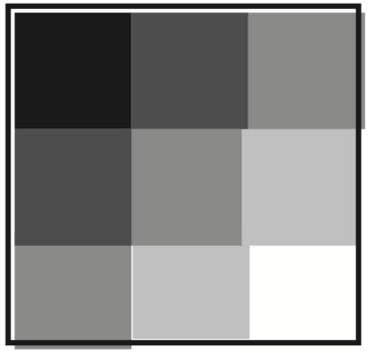
- A question is then: what is the usage of this seemingly wild finding?
- A possible answer is it could be used to compress images

Remark

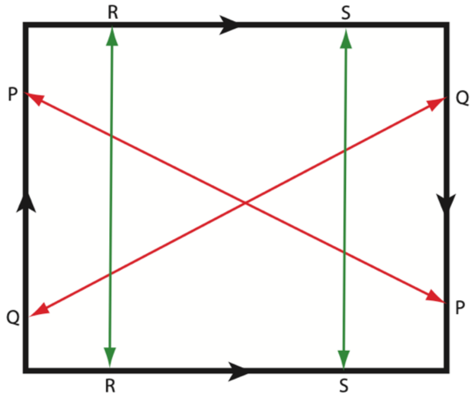
- A question is then: what is the usage of this seemingly wild finding?
- A possible answer is it could be used to compress images
- Each original patch lives in a 9-dimensional space (encodable by 9 numbers), but now we find that it lives in a 2-dimensional space (encodable by 2 numbers)

Remark

- A question is then: what is the usage of this seemingly wild finding?
- A possible answer is it could be used to compress images
- Each original patch lives in a 9-dimensional space (encodable by 9 numbers), but now we find that it lives in a 2-dimensional space (encodable by 2 numbers)



$\mathbb{R}^9$



$\mathbb{R}^2$

Further Remark

- The demonstrated case study is a almost “textbook” direct application of persistent homology
- The take a dataset (point cloud in this case), do some preprocessing and sampling, and then build the filtration and compute PD/barcode
- The whole process contains a lot of back-and-forth, trial-and-error
- They also try to refine their hypothesis or further test their findings based on initial findings