4.2 Our experimental results

We used the set of images provided by the skimage library from python and the python library Gudhi [source ? ?] to produce simplicial complexes and persistence diagrams. The main procedure was as followed :

- Take a sample of points from the image. We took a uniform subsample of the image.
- Compute the Rips-Vietoris complex from those points for a radius ϵ
- Set a value for edges : the distance between the two vertices. Value of vertices is set to 0.
- Apply an age filter to produce a filtration of nested complexes over the R-V complex.
- Compute the persistent pairs for homology groups for dimensions 0 and 1 and for 1 and 2. For dimensions 0 and 1 these are pairs (c, e) where c is a non zero element of a group $H_0(K_\epsilon)$ for some ϵ and e is an edge such that the add of e to $K_{\epsilon'}$ for $\epsilon' \geq \epsilon$ makes c vanish, i.e. c is a zero for $H_0(K_{\epsilon'})$.
- With the set of all edges for persistent pairs of dimensions 0-1 we can compute the covering tree over our cloud data point with minimum value.
  - Then we can remove the $n_c - 1$ edges which vanish the $n_c - 1$ most persistent component connexes of our filtration. This gives us the n_c most persistent connexe components.
  - Or we can add edges which gave birth to the most persistent cycles through the filtration. Then find loops with a traversal algorithm.
- The most persistent connexe components and the most persistent cycles give each a segmentation of our images.

We computed the most persistent connexe components for images of our data set (see 4.3). We first downsampled our image taking approximately 5000 points that is $\approx 2\%$ of our pixels. The more points we take the more chances are there to take points on edges or very irregular regions and these points may be very isolated in space. Also this segmentation is very sensible to illumination it tends to select dark regions of shadow or very bright regions. We noticed that segments obtained tend to select either large homogene regions (for instance wood from a table, a person) or small regions of interest (for instance a hand or an eye). Points on edges are very specific and therefore will more likely form persistent connexe components. These regions are not always of interest therefore one could apply a density filter on cloud data points, it forces the algorithm to produce larger region. We removed 10% of our points to get the results given below using a gaussian kernel density filter. In the same vein we applied a small gaussian blur it also avoids too specific small number points and makes our data points more homogene with the pixels of the same superpixel.

On 4.3 we note that we could not separate the head of the astronaut from the background since there is only a slight difference of color. That is one drawback of our method, which is not based on boundaries. One notices though that we achieved good results for natural images (cameraman 4.3) having the small superpixels for small details (hand, head, camera) and very large ones for the man and the background.

We then computed the most persistent cycles for images of our data set (see **??**, **??**, **??**, **??**, **??**, **??**). We remark that for each image the most persistent 1 cycle produces the most interesting segmentation. One drawback we can see is that since these are projections of 1-manifold in 3D or 5D into 2D, therefore it can lead to irregular shapes ? ?. Also like in the case of persistent connexe components all persistent 1-cycles are not always relevant.

if we does have an explicit control over the amount of superpixels we do not control their compactness. One would need a balance between color proximity and space proximity.

best parameters for 0-homology groups segmentation : - nb of data points : 5000 - no blur - no density filtration - minimum number of pixels per segments (postprocessing) : 10 - epsilon 120 - n events 500 best parameters for 1-homology groups segmentation : - nb of data points : 400 - no blur - no density filtration - epsilon 180 - n events 6

rho 0.1 -> 0.2 100 -> 250 -> 500 gauss 2.5 -> 5.

Procedure 0 :

- ♣ First we divise our image I of N_0 pixels into a set of N_1 square superpixels of size ${s_1}^2$. Each superpixel is represented by his center.

- ♣ We apply a small gaussian blur $G_{s_1/16}$ of standard deviation $s_1/16$ to I. Each superpixel whose center is at position (x, y) is given the value of $G_{s_1/16} * I(x, y)$. Then we obtain a cloud data point with N_1 points from $\mathbb{R}^3$ if I is a gray or N_1 points from $\mathbb{R}^5$ if I is a RGB image.

- ♣ In order to get a filtered simplifical complex we compute the Vietoris-Rips complex VR_{ϵ_∞} over our cloud data point. Using the following filtration $f : VR_{\epsilon_\infty} \to \mathbb{R}$ where :

$$f([v_1, \ \dots \ , \ v_n]) = \max_{[v_i, v_j] \in VR_{\epsilon_\infty}, \ 1 \leq i,j \leq n} \|v_i - v_j\|_2 \tag{4.2.1}$$

  we deduce the filtered simplicial complex $(f^{-1}(\epsilon))_{0 \leq \epsilon \leq \epsilon_\infty}$, where VR_0 is the set of all superpixels with no edges.

- ♣ We then compute 0-homology groups $H_0(f^{-1}(\epsilon))$ for $0 \leq \epsilon \leq \epsilon_\infty$, that is, connected components. We have $\beta_0 = H_0(VR_{\epsilon_\infty})$. In practice we uses persistence pairs $(v_i, [v_j, v_k])$ where the add of the edge $[v_j, v_k]$ corresponds to the death of the component $[v_i]$.

- ♣ Finally we take the subgraph G of VR_{ϵ_∞} keeping only edges from persistence pairs. G has β_0 nodes and $N_1 - \beta_0$ edges. If $\beta_0 = 1$ it gives us in fact the covering tree of minimal weight over our cloud point data. In order to get the $\beta_0 + N_c$ most persistent 0-homology groups, we remove N_c edges from G.

Here is one example of image segmentation produced using the previous procedure.

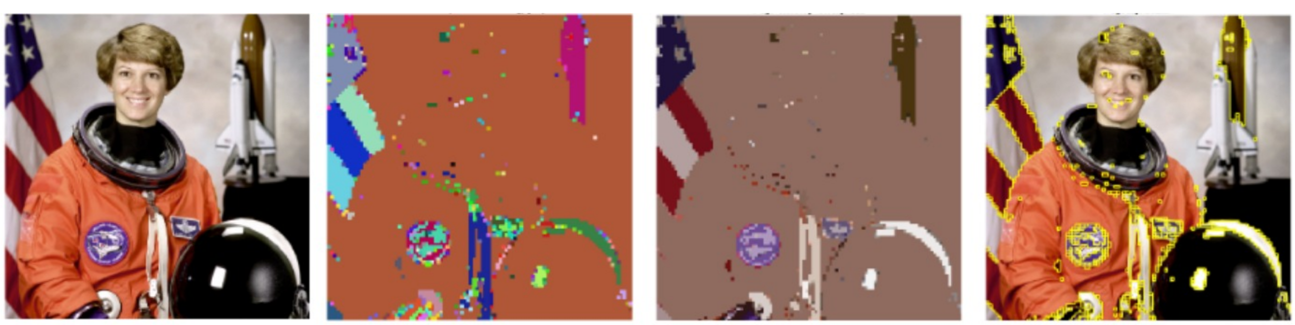

FIGURE 4.1 – image segmentation using the 250 most persistent 0-homology groups

One can make few observations :

⟶ image segmentation is not parsimonious : most of the segments are made of very few pixels.

⟶ very small segments are located along edges.

⟶ large areas of original image are not recovered.

To tackle the issue of having too small segments we propose to add the following steps to the procedure :

- ♣ Among the $\beta_0 + N_c$ most persistent connected components we only keep those made of more than c_1 elements, i.e. superpixels. We note I_l the image of same size as I where $I_{\mathcal{L}}(x, y)$ is the label of pixel at position (x, y) for labelled pixels and is set to zero for removed pixels. Labels are integers $l \in \mathcal{L} = \{1, ..., N_{\mathcal{L}}\}$.

- ♣ In order to infer labels for each segment of size $\leq c_1$ we look for neighbors in the image grid which have been labelled. If none is found in 8-connexity we look for such neighbors in 16-connexity ect... until we have at least found one. Let's note V_P the set of superpixels in the neighborhood of P which have already been labelled. Also we write μ the current distribution of labels in the image s.t. $\mu = \sum_{l=1}^{N_{\mathcal{L}}} \frac{card(\{I_{\mathcal{L}}=l\})}{card(\{I_{\mathcal{L}} \in \mathcal{L}\})} \, \delta_l$.
From μ we derive $\tilde{\mu}_{V_P} = \sum_{l=1}^{N_{\mathcal{L}}} \big(\sum_{Q \in V_P} \mathbb{1}_{I_{\mathcal{L}}(Q)=l} \big) \, \mu(l)^{-\alpha} \, \delta_l$ and let μ_{V_P} be the distribution deduced from $\tilde{\mu}_{V_P}$.

Here is the result using the proposed method :

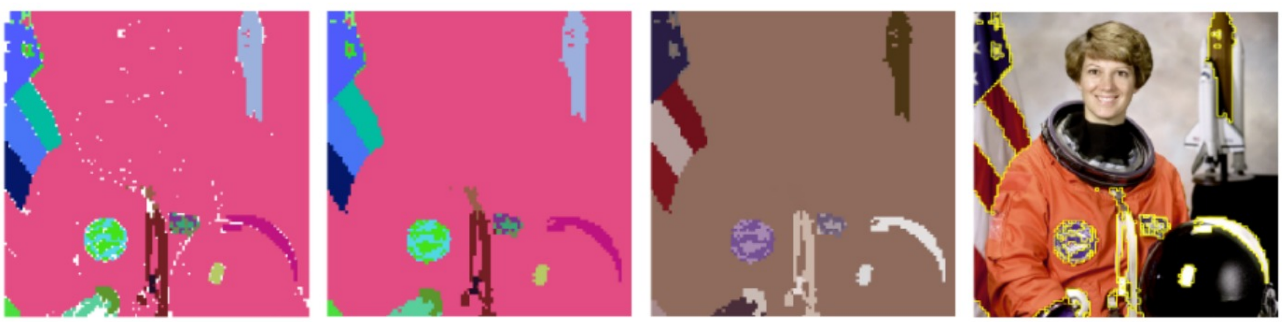

FIGURE 4.2 – image segmentation using the 21 most persistent 0-homology groups of size at least 10.

One can make few observations :

$\longrightarrow$ image segmentation is more parsimonious : we got only 21 segments instead of 250.

$\longrightarrow$ obviously we have no more small segments.

$\longrightarrow$ large areas of original image are still not recovered.

PROBLEM WITH TAKING SEG FROM ALL SIZES!! SHOW EX WITH MIN SIZE 0

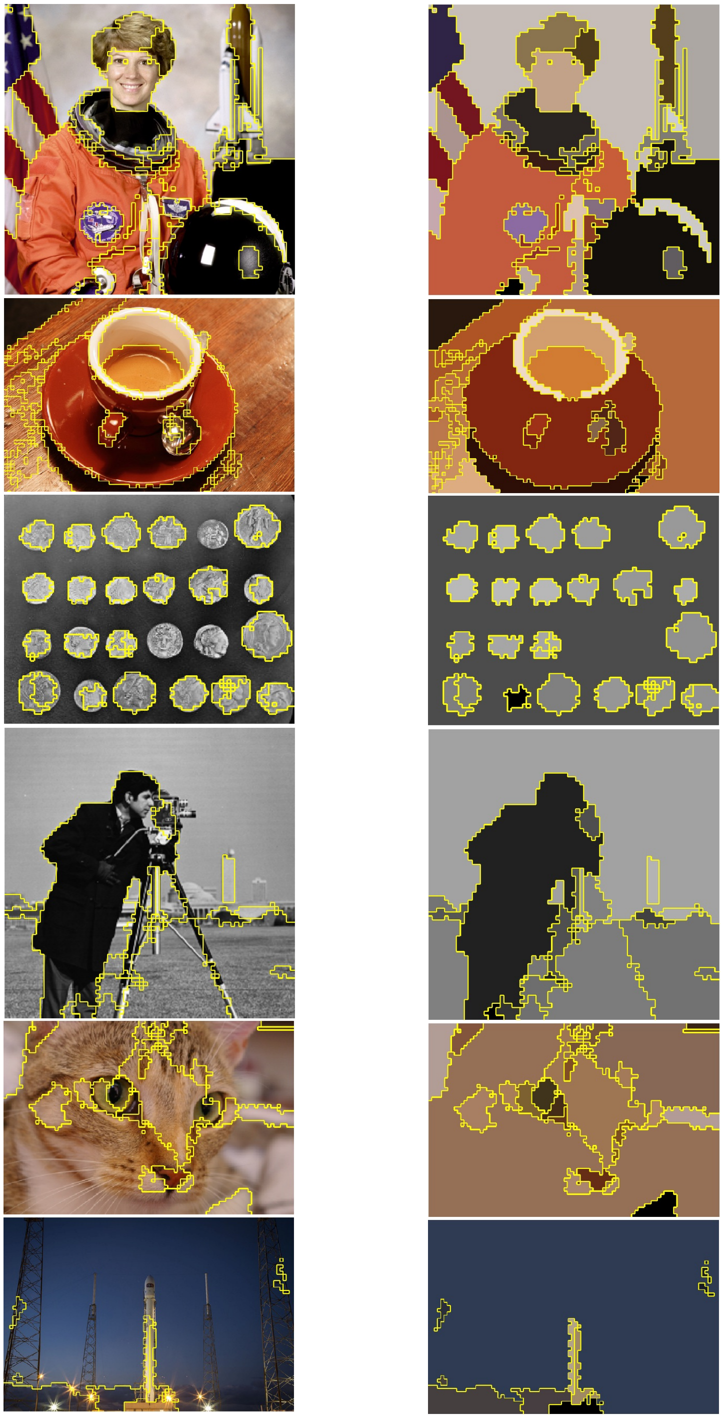

FIGURE 4.3 – Segmentation computed using persistent 0-homology groups on a set of images with the exact same parameters for each. 33, 24, 27, 22, 10, 7

fefezfe dezdfezzfe dezfergflj-clùsdchNC
JMFNNEMB HSBVL
JDBV QVBLDHBQJ ; QMCB-QSBCL
JHSQ Mhrkfherfebfrf erfrerfrfe

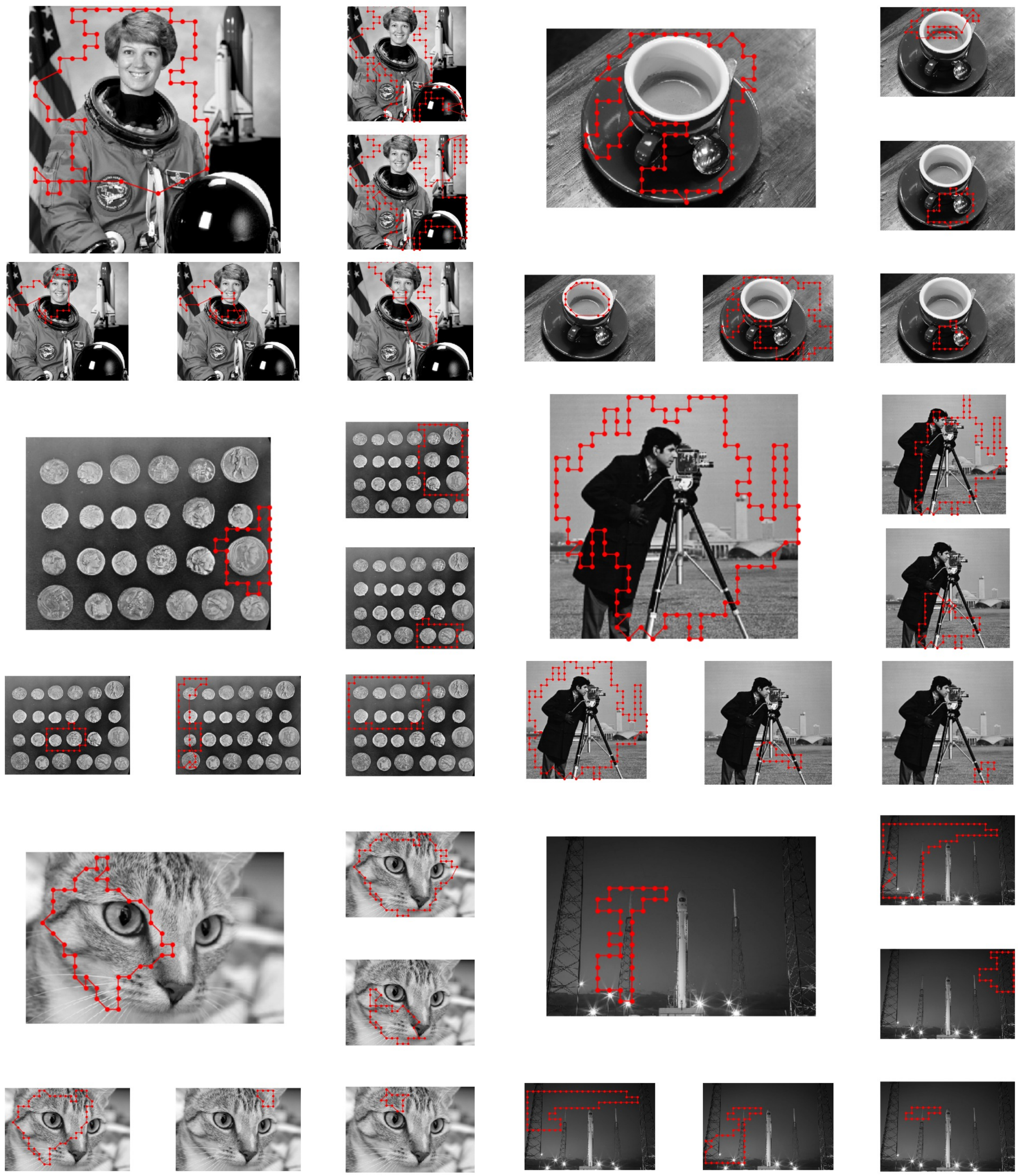

FIGURE 4.4 – The 6 most persistent 1-homology groups for a set of images using the exact same parameters.

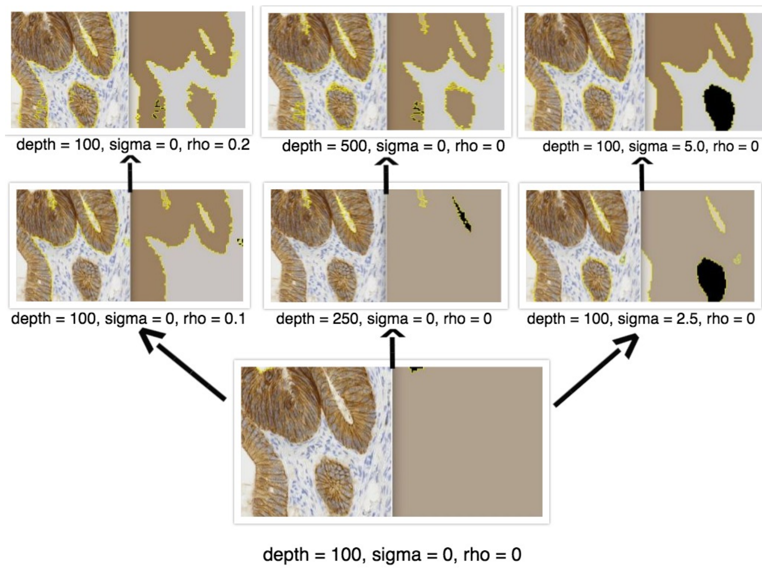

FIGURE 4.5 – Link between depth of search of persistent 0-homology groups, blurring with a coefficient sigma, and filtering with a gaussian density kernel removing rho % of points.