Evaluation
Supplemental material #5 to
Multi-Scale Label-Map Extraction for Texture Synthesis

Yitzchak David Lockerman[1], Basile Sauvage[2], Rémi Allègre[2], Jean-Michel Dischler[2], Julie Dorsey[1], Holly Rushmeier[1]
[1] Yale University, Computer Graphics Group, New Haven, USA
[2] ICube, Université de Strasbourg, CNRS, France

Contents

1 Introduction

In this document we evaluate our algorithms and design choices. We prepared a data set consisting of large inputs (section 2). Unlike segmentation data sets, which usually contain images with individual objects to be detected, we chose inputs more suitable for texture extraction: non-stationary textures (few perspective distortions and shadows) and natural images containing textures. We drawn clustering manually which will serve as ground truth (section 2). In section 3 we explain the numerical measures we use. Then we evaluate our hierarchical SLIC algorithm compared to hierarchical segmentation (section 4), the choice of feature vectors (section 5), and our clustering algorithm compared to k-means and spectral clustering (section 6).

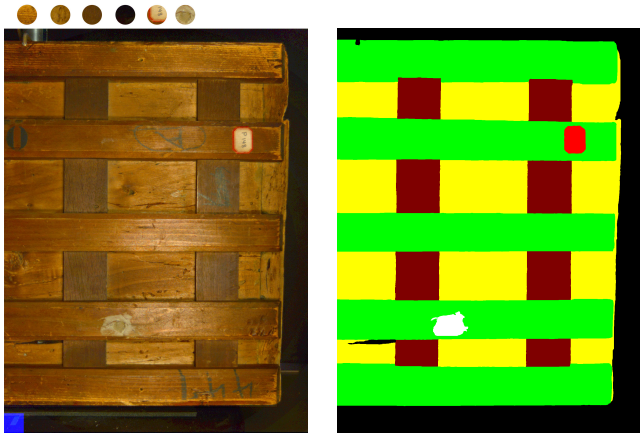
We would like to note that one of our algorithms strength is its flexibility. A number of its components can be directly swapped to apply it to different use cases. Thus, while we give justifications for our choices here, we recommend that those applying our method consider if there use cases justify different choices.

2 Datasets

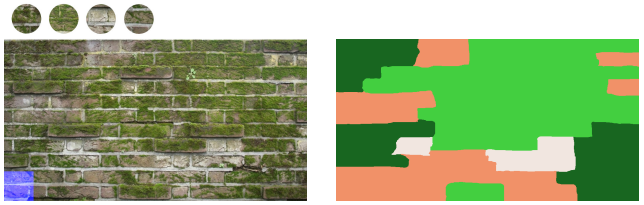
Input images with one ground truth for each of them are shown in figures 1 and 2. Table 1 provides the resolutions and an approximate scale of the patterns labeled in the shown ground truth. Users were asked to: 1) identify the patterns at a scale that they fixed themselves, e.g. by means of a square window, 2) label the pixels according to the identified patterns.

Dataset name	Resolution in pixels	Ground truth scale in pixels
Boxtop	1536×2048	100
Brick_small_dirty	1024×544	100
Bubbling_rusty_metal	2560×1920	200
Bush	1600×1200	150
City	2016×703	75
Concrete_floor_damaged	1024×636	40
Earth	1440×720	34
Flower	1600×1200	100
Moss	1024×683	200
Plaster_damaged	373×560	60
River	1616×616	100
Grass	1600×1200	300
Tower	2618×3907	400
Wall_with_door	2560×1920	300
Wall_with_pipe	2560×1920	250

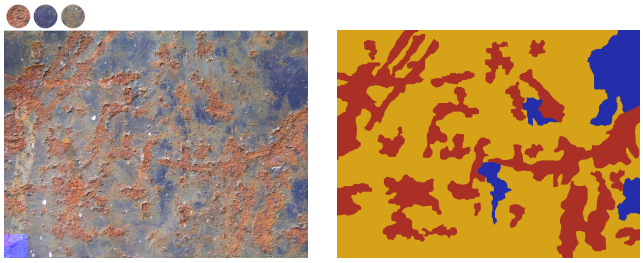
Table 1: *Resolution of our test images and pattern scale of our ground truth labeling for each test image.*



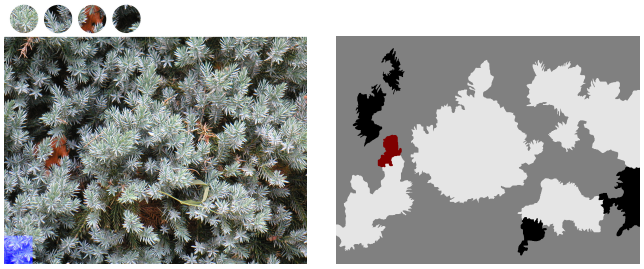
Boxtop



Brick_small_dirty

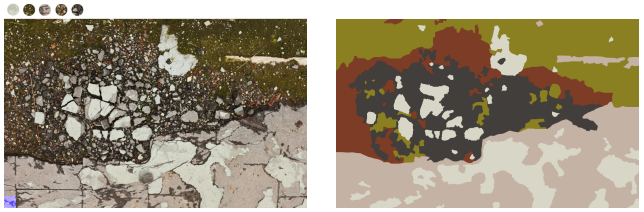


Bubbling_rusty_metal

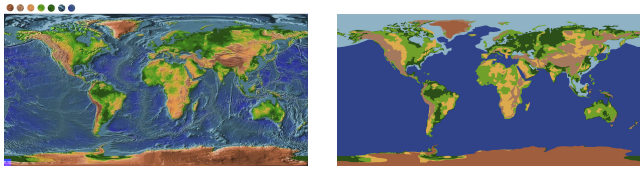


Bush

City



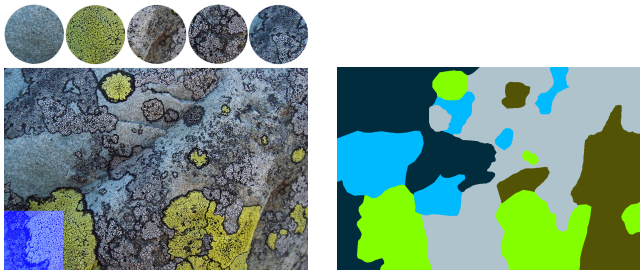
Concrete_floor_damaged



Earth

Flower

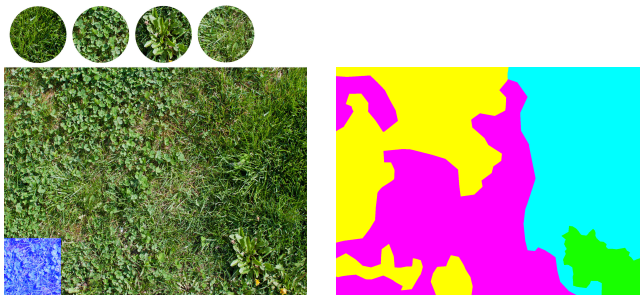
Figure 1: *Our datasets, part 1/2: input image (left) and our manually assigned texture labels (right). The superimposed blue square at the bottom-left corner of input images represents the pattern scale at which the manual clustering was performed. The image row above each input image shows the set of texture patterns that were identified by the users.*



Moss

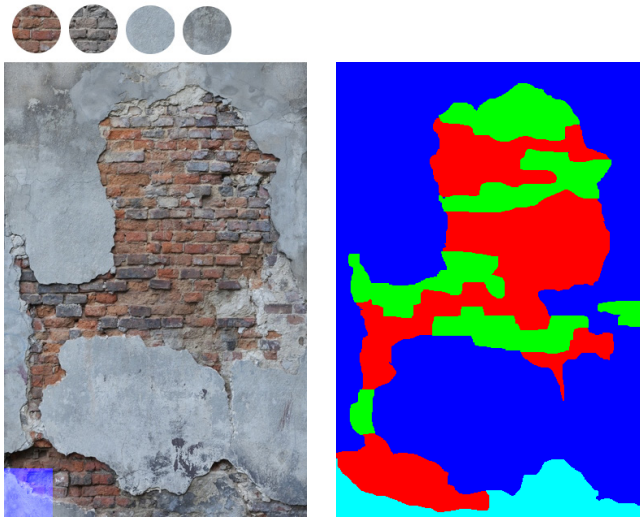
Wall_with_door

River



Grass

Tower



Plaster_damaged

Wall_with_pipe

Figure 2: *Our datasets, part 2/2: input image (left) and our manually assigned texture labels (right). The superimposed blue square at the bottom-left corner of input images represents the pattern scale at which the manual clustering was performed. The image row above each input image shows the set of texture patterns that were identified by the users.*

3 Numerical measures

Let $\mathbb{S} = \{S_1, S_2, \ldots, S_{|\mathbb{S}|}\}$ be a partition of an image with N pixels. For any pixel $\mathbf{p} \in S_i$, let denote $\mathbb{S}(\mathbf{p}) = i$ the region index. We have to compare numerically partitions of a same image, which we know a ground truth clustering $\mathbb{G}$ for.

When $\mathbb{S}$ is a clustering, each cluster S may have several regions. When $\mathbb{S}$ is a partition (or segmentation), each segment S is a single region. We use the same ground truth in both cases. However, different regions belonging to the same cluster are considered as distinct regions in a segmentation point of view.

3.1 Measures for partitions / segmentations

Superpixel partitions are a basis for later clustering which may be seen as an under-segmentation. In order to be useful each superpixel should ideally be contained in a single region. This condition can be assessed by the under-segmentation error [Achanta et al. 2012]

$$U(\mathbb{S} \to \mathbb{G}) = \frac{1}{N} \left(\sum_{G \in \mathbb{G}} \left(\sum_{S \in \mathbb{S}, \frac{|S \cap G|}{|S|} > \epsilon} |S| \right) - N \right) \quad (1)$$

where ϵ is a threshold to tolerate small errors in the ground truth [Achanta et al. 2012], which we set to $\epsilon = 1\%$. A smaller under-segmentation error indicates a set of superpixels that are more fateful to the ground truth.

3.2 Measures for clustering

Conversely to superpixels, clusters are comparable to the ground truth. Thus we make use of two standard segmentation metrics [Arbeláez et al. 2011].

The Rand Index is defined as

$$RI(\mathbb{R}, \mathbb{S}) = \frac{2}{N(N-1)} \sum_{\mathbf{p}} \sum_{\mathbf{q}>\mathbf{p}} (\mathbb{R}(\mathbf{p}) = \mathbb{R}(\mathbf{q})) \quad \& \quad (\mathbb{S}(\mathbf{p}) = \mathbb{S}(\mathbf{q})) \quad || \quad (\mathbb{R}(\mathbf{p}) \neq \mathbb{R}(\mathbf{q})) \quad \& \quad (\mathbb{S}(\mathbf{p}) \neq \mathbb{S}(\mathbf{q})) \quad (2)$$

where $\mathbf{q} > \mathbf{p}$ compares positions in lexicographic order, and the boolean expression is interpreted as 1 or 0. It measures the proportion of pairs of pixels $(\mathbf{p}, \mathbf{q})$ that are either in the same region both in $\mathbb{R}$ and in $\mathbb{S}$, or in different regions both in $\mathbb{R}$ and in $\mathbb{S}$.

The covering of $\mathbb{R}$ by $\mathbb{S}$ is an asymmetric measure defined as

$$Cov(\mathbb{R} \to \mathbb{S}) = \sum_{R \in \mathbb{R}} \frac{|R|}{N} \max_{S \in \mathbb{S}} \frac{|R \cap S|}{|R \cup S|} \quad (3)$$

Both of these measures give values between 0 (worse) and 1 (best).

3.3 Dealing with multi-scale partitions

One issue is to compare a multi-scale sequence $\{\mathbb{S}^1, \ldots, \mathbb{S}^\Gamma\}$ with a single scale ground truth $\mathbb{G}$. The idea is that we do not expect one single scale to perfectly match the ground truth but we want the hierarchy to potentially recover any ground truth. Thus we plot curves of the metric on $\mathbb{S}^\gamma$ against $|\mathbb{S}^\gamma|$. So different sequences may be compared even if abscissas $|\mathbb{S}^\gamma|$ are not equal.

4 Evaluation of the segmentation

Our hierarchical SLIC superpixels algorithm (H-SLIC) is similar to segmentation though the regions we seek have different properties. In this section we compare H-SLIC with the hierarchical segmentation (H-Seg) of Arbeláez et al. [2011]. Both H-SLIC and H-Seg produce a sequence of nested partitions $\{\mathbb{S}^1, \ldots, \mathbb{S}^\Gamma\}$, that is $\mathbb{S}^{\gamma-1}$ is a sub-partition of $\mathbb{S}^\gamma$. As explained in section 3 we plot curves $U(\mathbb{S}^\gamma \to \mathbb{G})$ against $|\mathbb{S}^\gamma|$. Numerical results are shown in figure 3 and 4. Note that the mininum and maximum numbers of regions of H-SLIC and H-Seg curves may differ for a dataset. The reason is that the H-SLIC and H-Seg data were produced independently. The comparison between the results of the two methods is thus relevant in the range were the numbers of regions are similar. Image sequences illustrating H-SLIC and H-Seg results are available as videos following this link: graphics.cs.yale.edu/HSLIC_video.html.

We draw the following conclusions.

- Compared to H-Seg, H-SLIC performs:
  - often better (Bush, Bubbling_rusty_metal, Moss, tower, Wall_with_door, Wall_with_pipe);
  - sometimes equivalent (Brick_small_dirty, Concrete_floor_damaged, Earth, Flower, Plaster_damaged, River);
  - rarely worse (Boxtop).
- H-SLIC performs better on both non-stationary textures (e.g. Bush, Bubbling_rusty_metal) and "textured images" (e.g. Wall_with_door).
- H-SLIC may struggle with region boundaries with low contrast differences (Boxtop).
- H-Seg performs well when region boundaries are sharp.

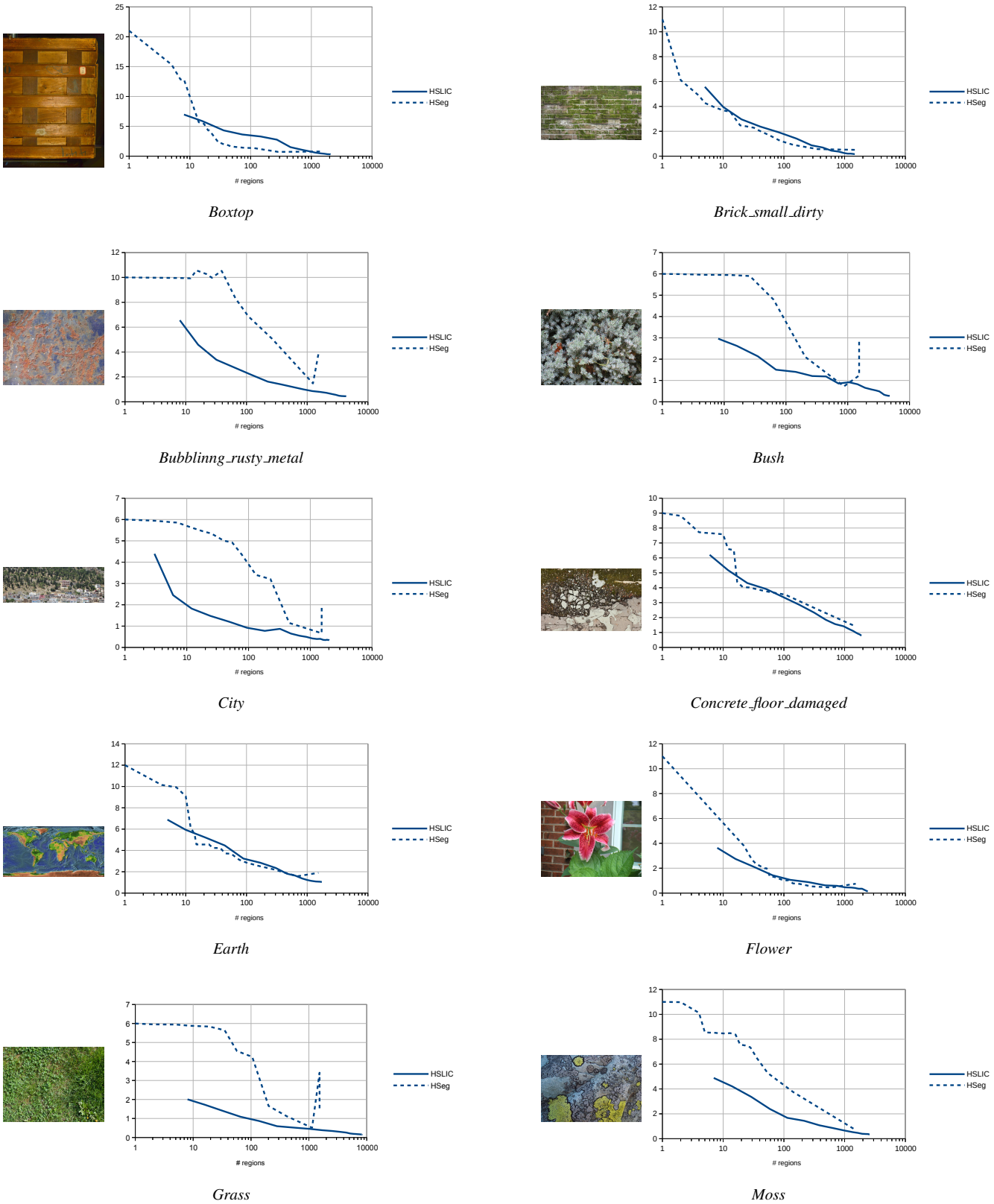
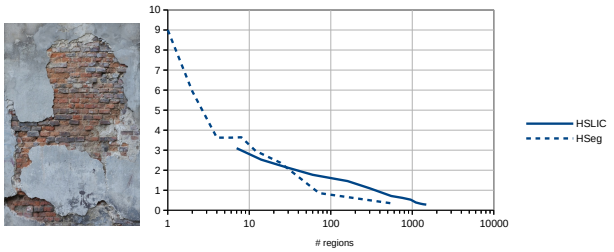
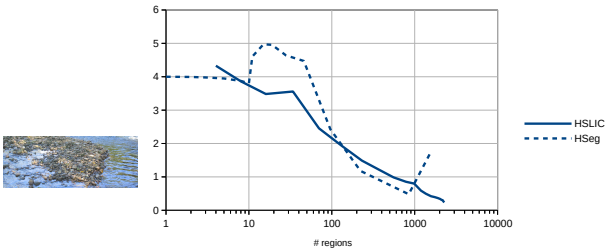


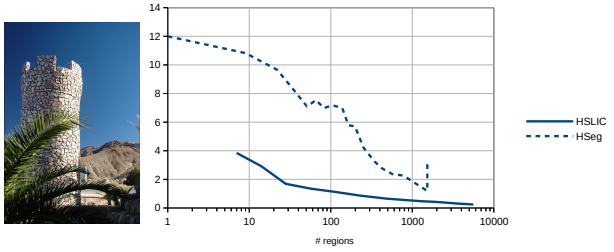
Figure 3: *H-SLIC (solid line) versus H-Seg (dashed line), part 1/2. Lower values indicate superpixels are more faithful to ground truth.*



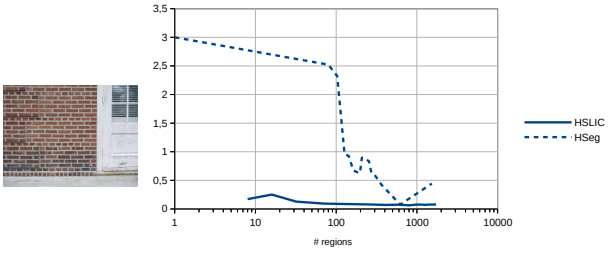
Plaster_damaged



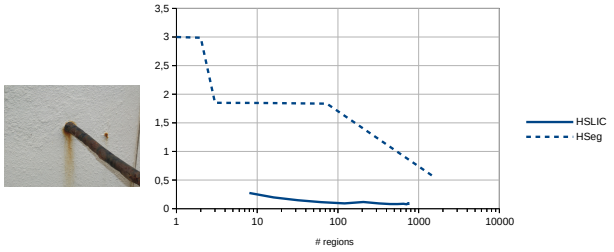
River



Tower



Wall_with_door



Wall_with_pipe

Figure 4: *H-SLIC (solid line) versus H-Seg (dashed line), part 2/2. Lower values indicate superpixels are more faithful to ground truth.*

5 Evaluation of the feature vectors

Our system requiters a method of comparing superpixels. For algorithmic simplicity and speed, we assign each superpixel a feature in a feature space, and then approximate k-nearest-neighbors [Mount 2010] to efficiently perform the comparison. Direct comparison of superpixels is left to future work. We tested the following feature vectors to describe a region:

Moments. The 15 dimensional vector includes, for each of the 3 Lab color channels, the first 5 centralized moments. That is it includes the mean, standard deviation, and 3 higher moments [Lockerman et al. 2013].

Gabor filter. We compute filters in frequency space as defined by Equation 31 of Movellan [2002]. A single octave is used with a central frequency of $\frac{20}{\sigma}$, where σ is the scale of level. Six angles are selected with 30 degrees angular resolution. For each filter we place the sum of square values of both the real and imaginary components. This leads to a 36 dimensional feature vector (3 color channels $*$ 1 octave $*$ 6 angles $*$ 2 components).

We chose a filter size to be $\frac{2}{\max(a,b)}$, where a and b are defined by Movellan [2002]. Care must be taken due to the non-rectangular shape of superpixels. We use binary erosion with a square neighborhood of half the size of the filter to remove points whose values suffer from boundary effects.

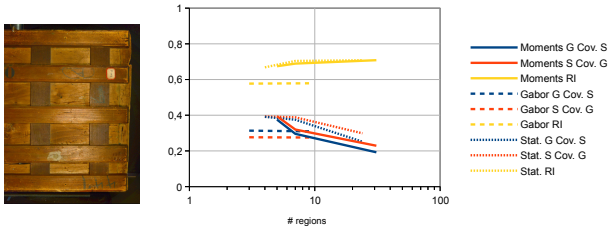
It should be noted that this feature space requires superpixels to be large enough to perform the convolution. As such, we bound the base scale to 25 pixels and requere each superpixel to contain at least 100 pixels.

Local statistics. The 63 dimensional vector includes, for each of the 3 Lab color channels, the means over cells of a quadtree: the whole region; 4 quadrants weighted $1/4$; 16 sub-quadrants weighted $1/16$. We split the quadtree at the mean row and column of each set of points.

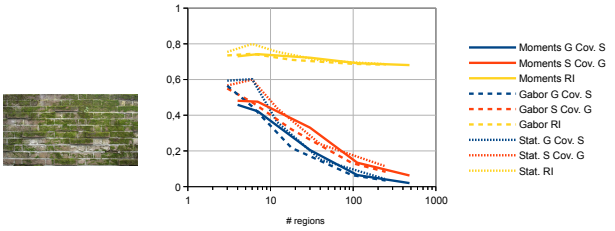
Several qualitative comparisons are shown in figures 7 to 21. We show the clustering C^l for several levels l.

Quantitative comparisons, obtained as explained in section 3, are provided in figures 5 and 6.

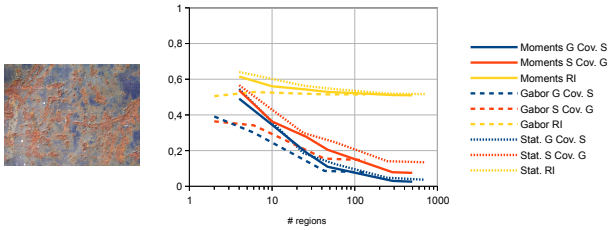
In conclusion, there is qualitative differences in the resulting clusters. However none of the 3 feature vectors performs better universally. The moments vector has the advantage of being the simplest computationally, so it sounds a good compromise. Alternatives are available and usable depending on the needs of the user.



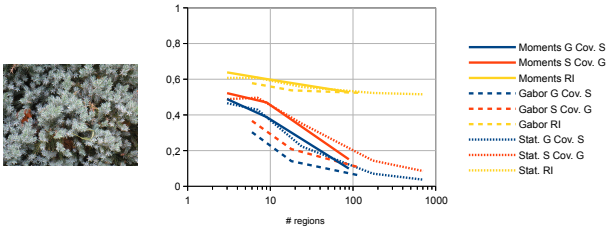
Boxtop



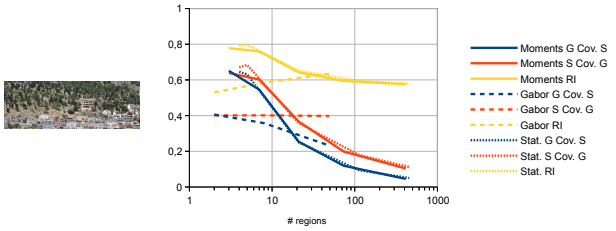
Brick_small_dirty



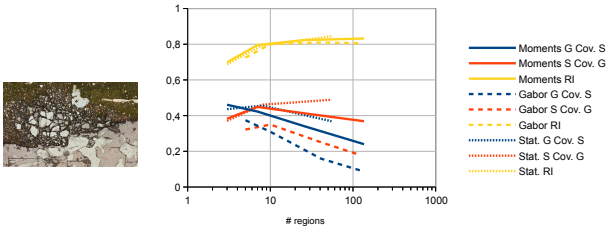
Bubblinng_rusty_metal



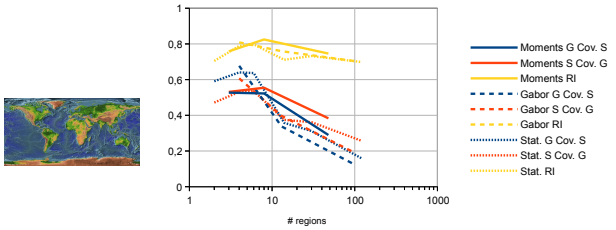
Bush



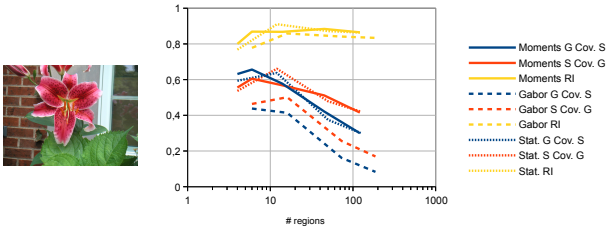
City



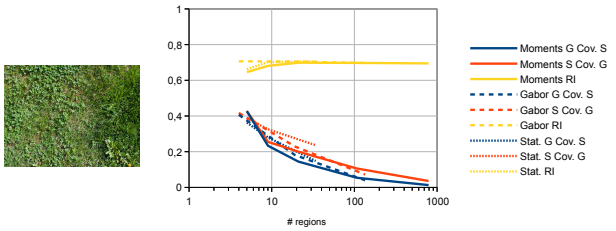
Concrete_floor_damaged



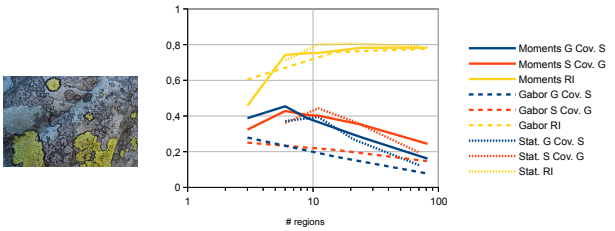
Earth



Flower

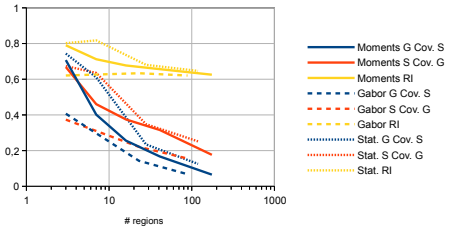


Grass

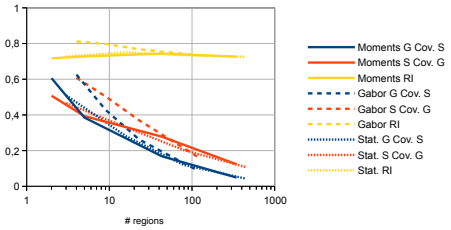


Moss

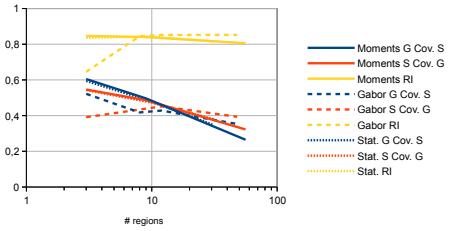
Figure 5: *Comparison of feature vectors, part 1/2. Note that each color represents a different quantitative measure that is not comparable to the other colors. Higher values represents better agreement between algorithm result and ground truth.*



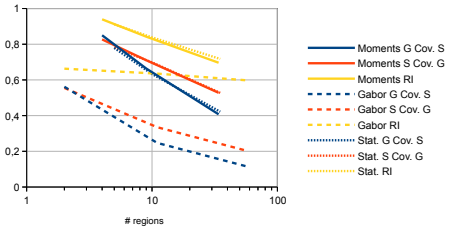
Plaster_damaged



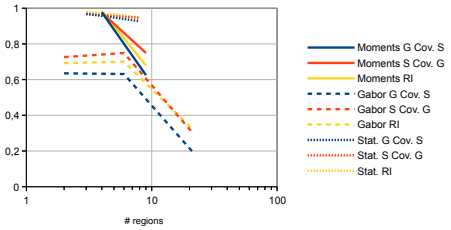
River



Tower



Wall_with_door



Wall_with_pipe

Figure 6: *Comparison of feature vectors, part 2/2. Note that each color represents a different quantitative measure that is not comparable to the other colors. Higher values represents better agreement between algorithm result and ground truth.*

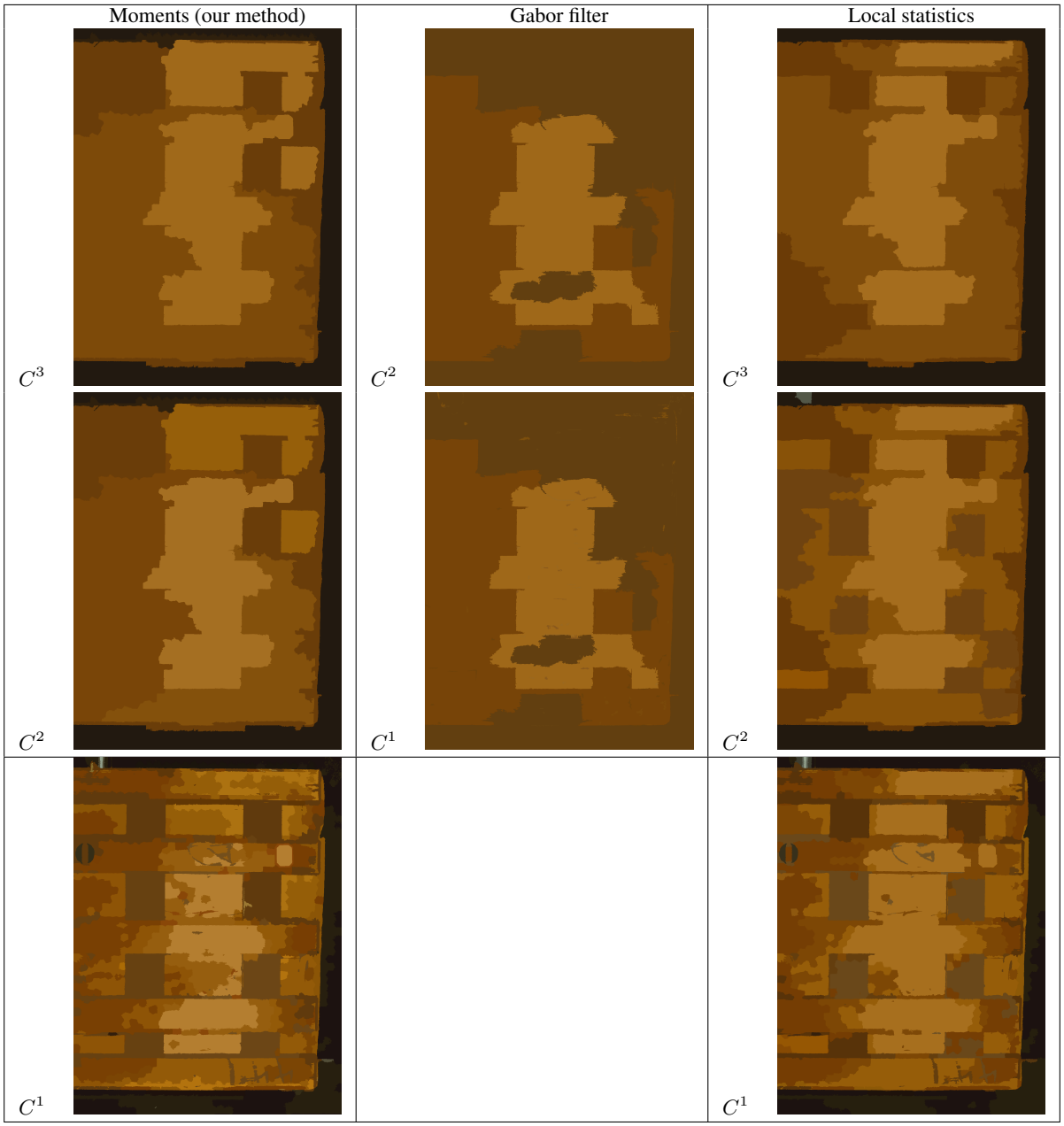


Figure 7: *Clustering results with different feature vectors for image Boxtop.*

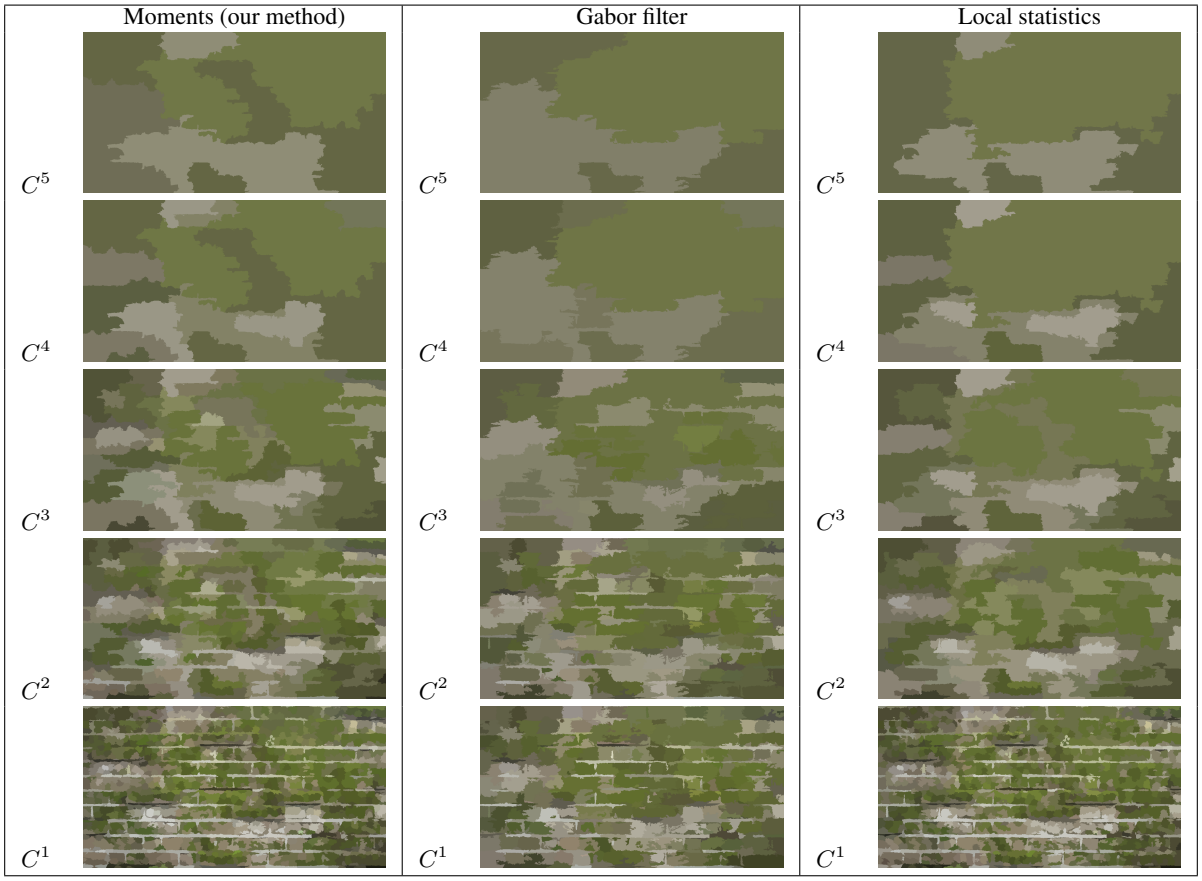


Figure 8: *Clustering results with different feature vectors for image Brick_small_dirty.*

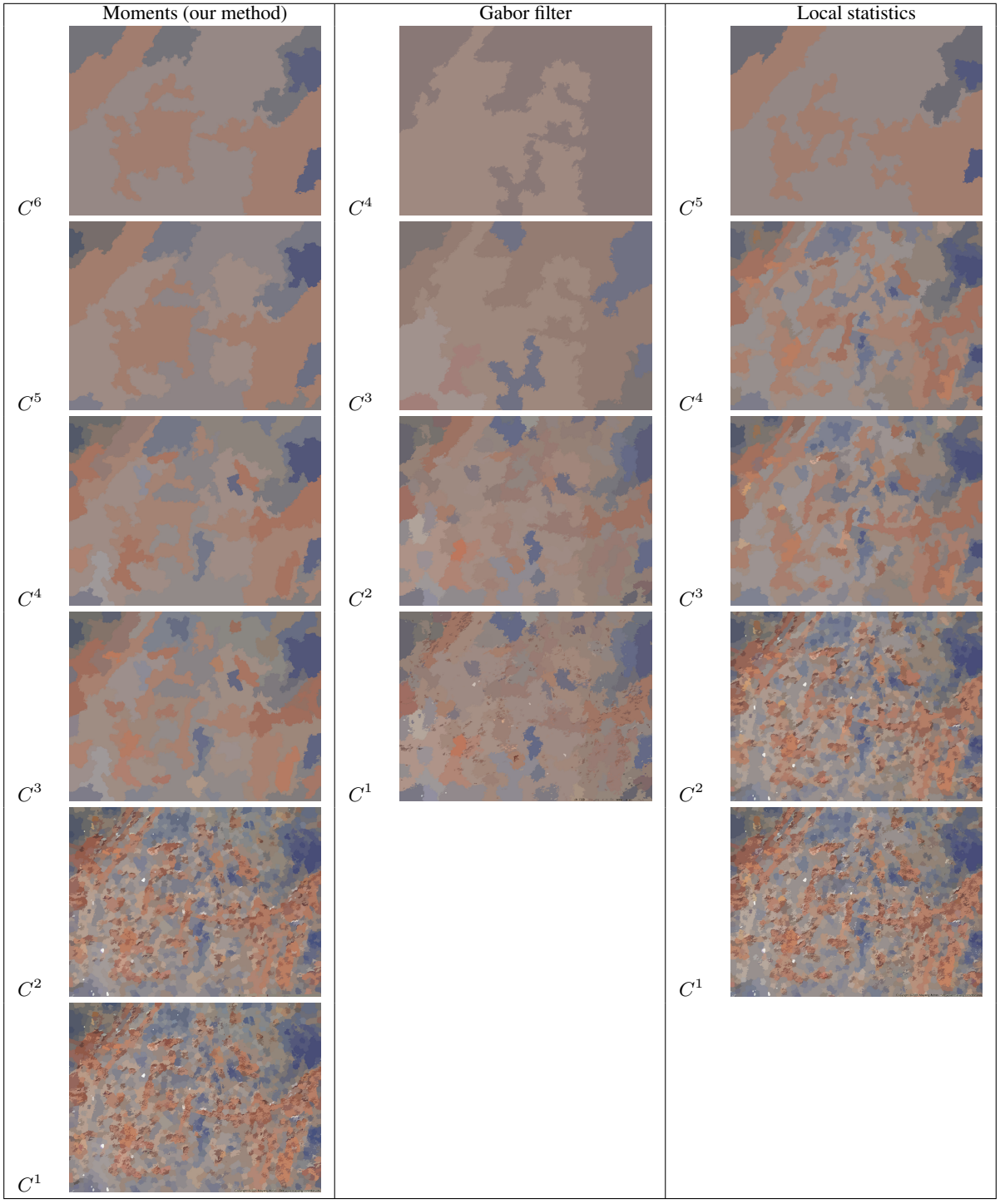


Figure 9: *Clustering results with different feature vectors for image Bubbling_rusty_metal.*

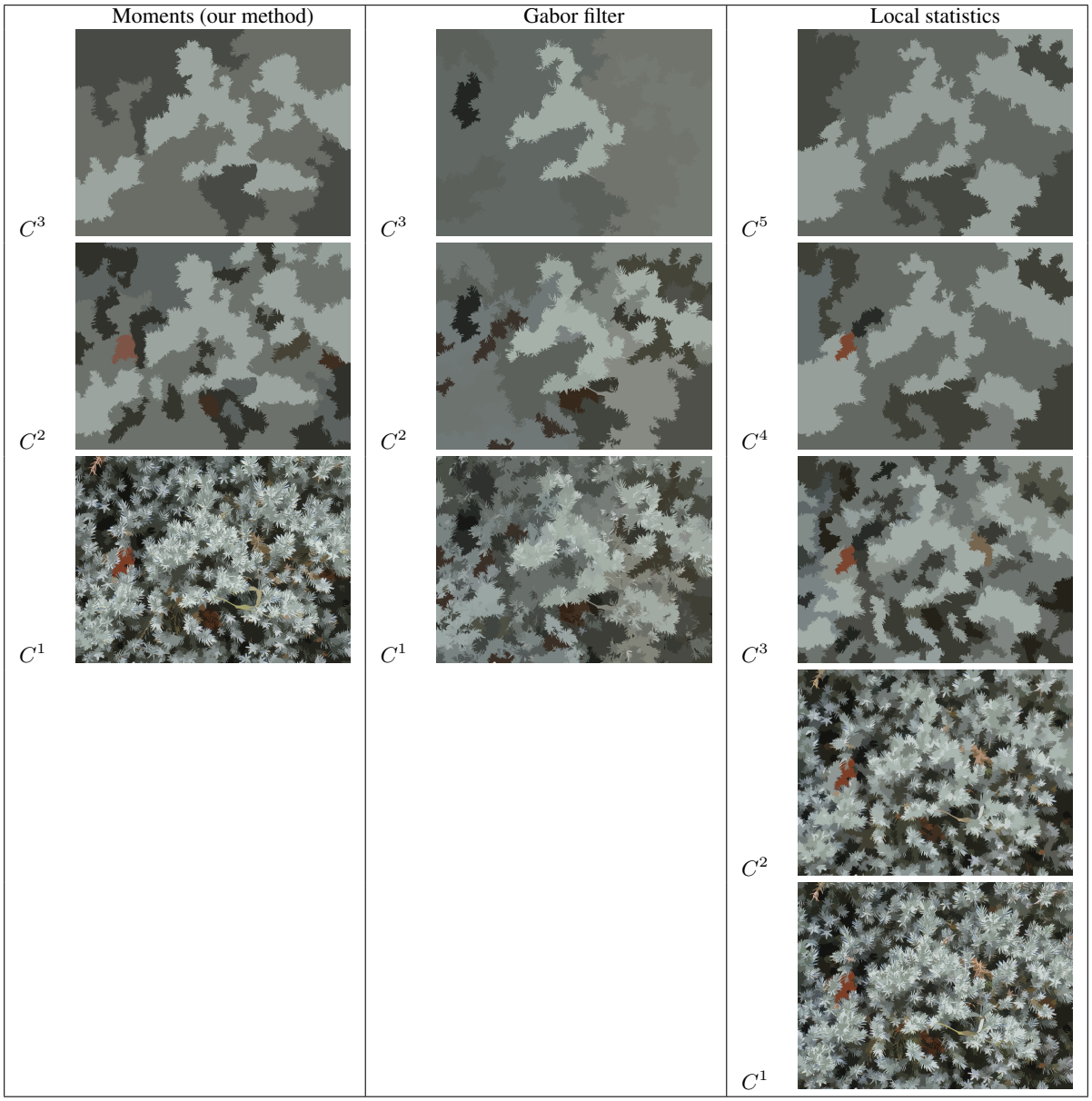


Figure 10: *Clustering results with different feature vectors for image Bush.*

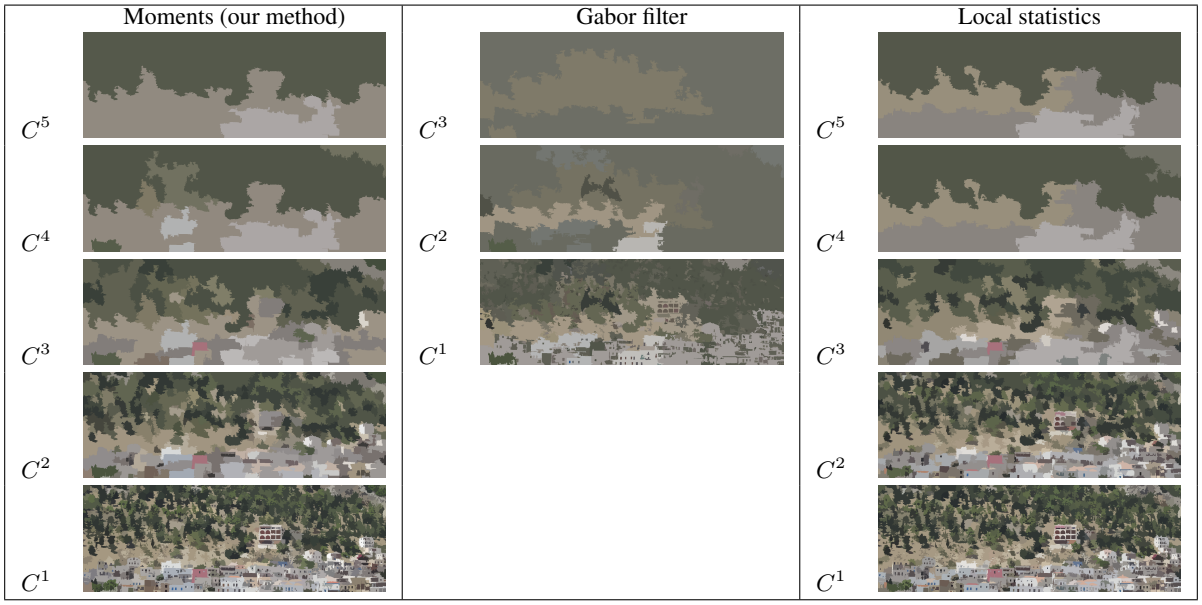


Figure 11: *Clustering results with different feature vectors for image City.*

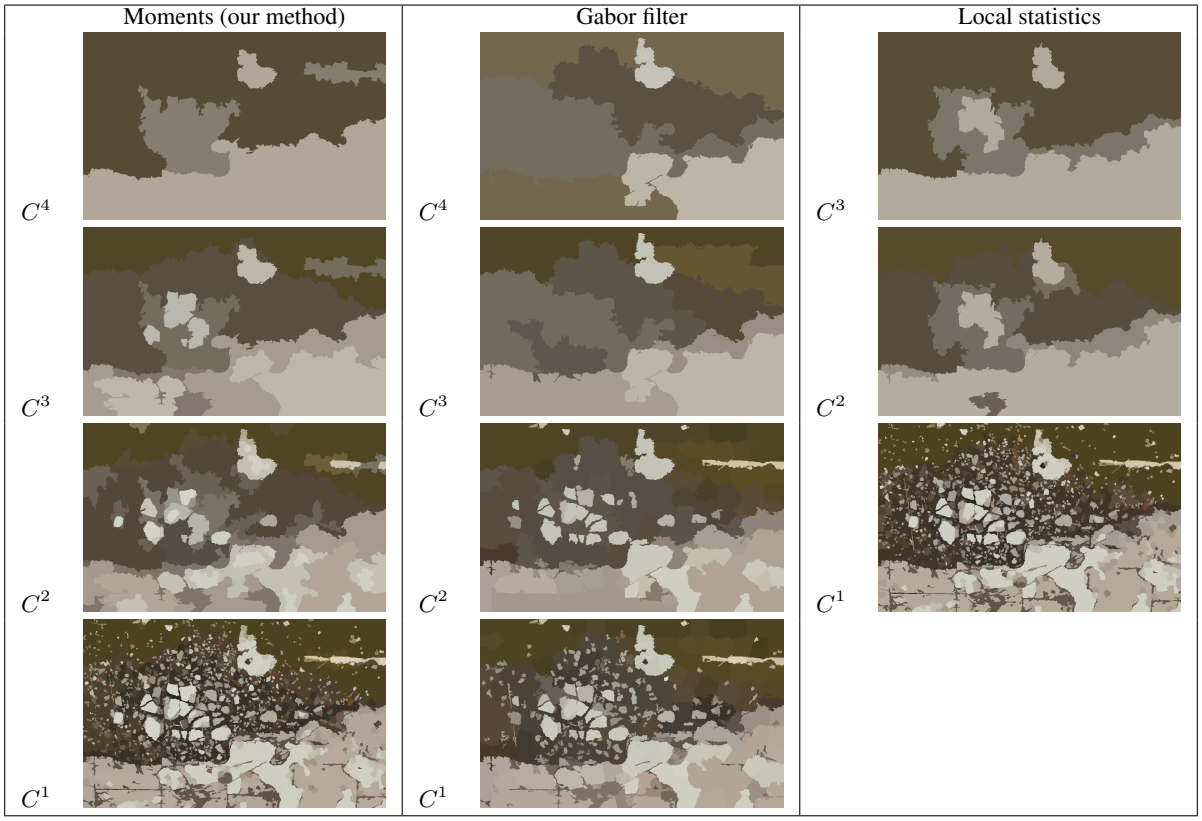


Figure 12: *Clustering results with different feature vectors for image Concrete_floor_damaged.*

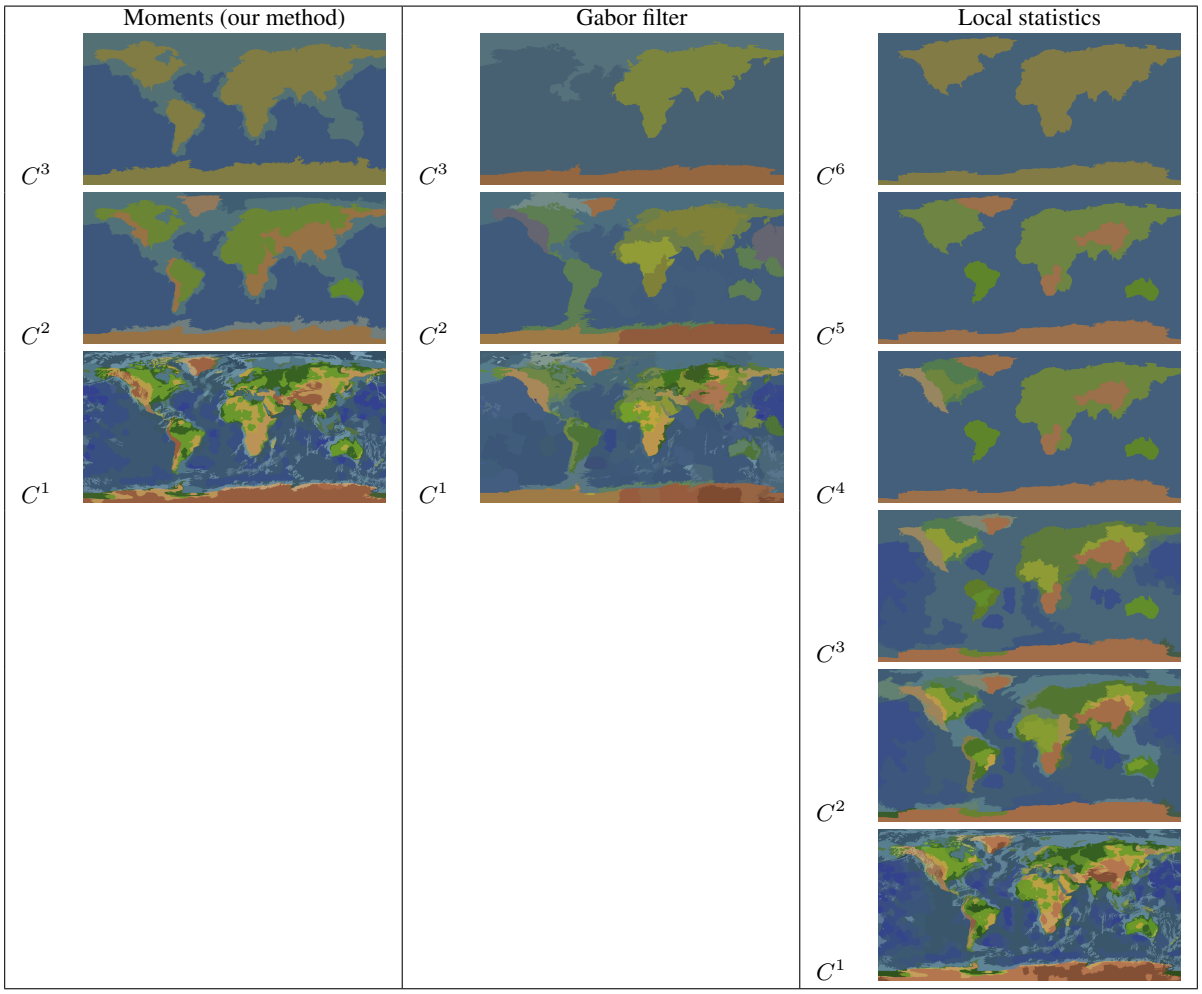


Figure 13: *Clustering results with different feature vectors for image Earth.*

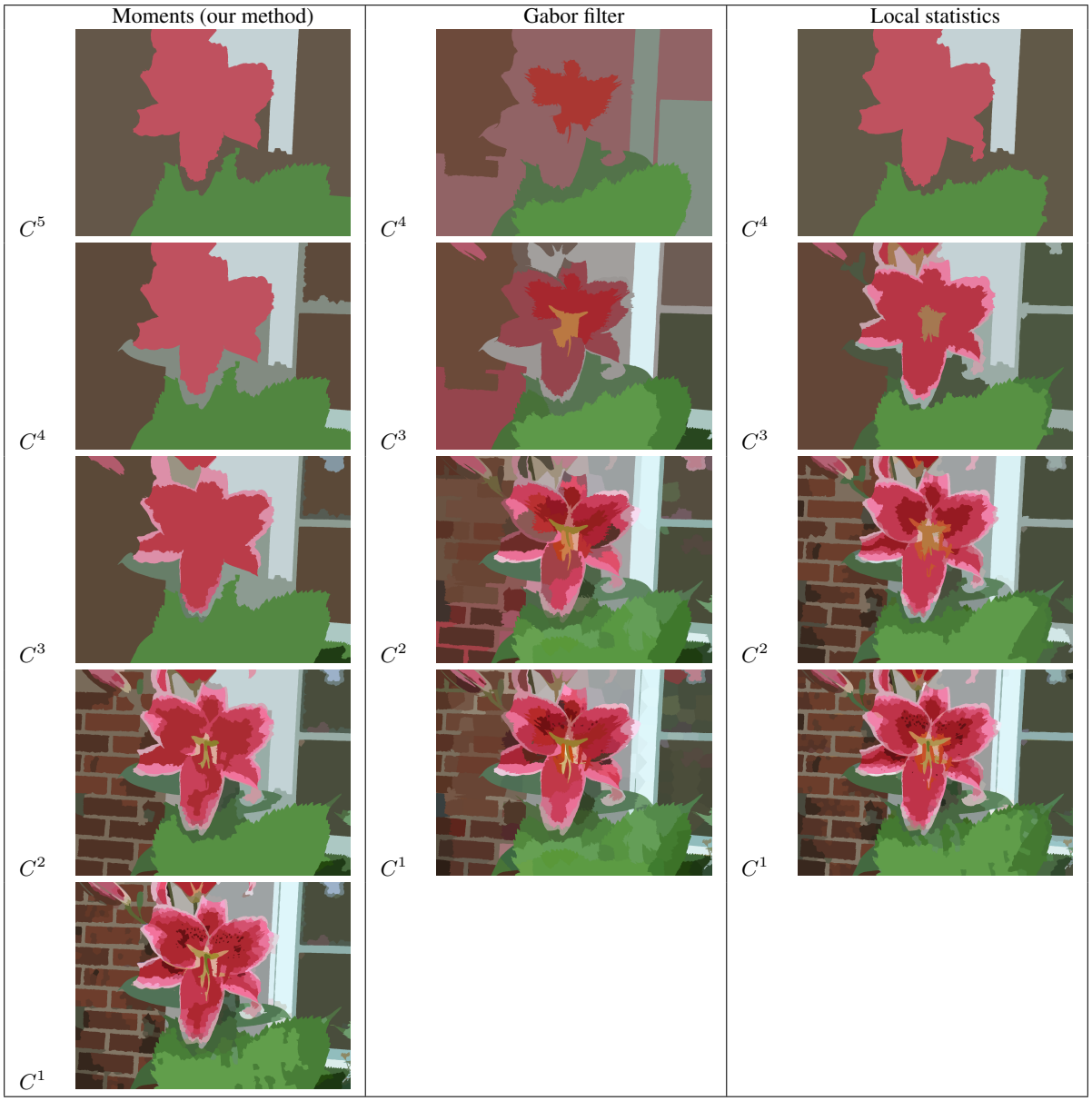


Figure 14: *Clustering results with different feature vectors for image Flower.*

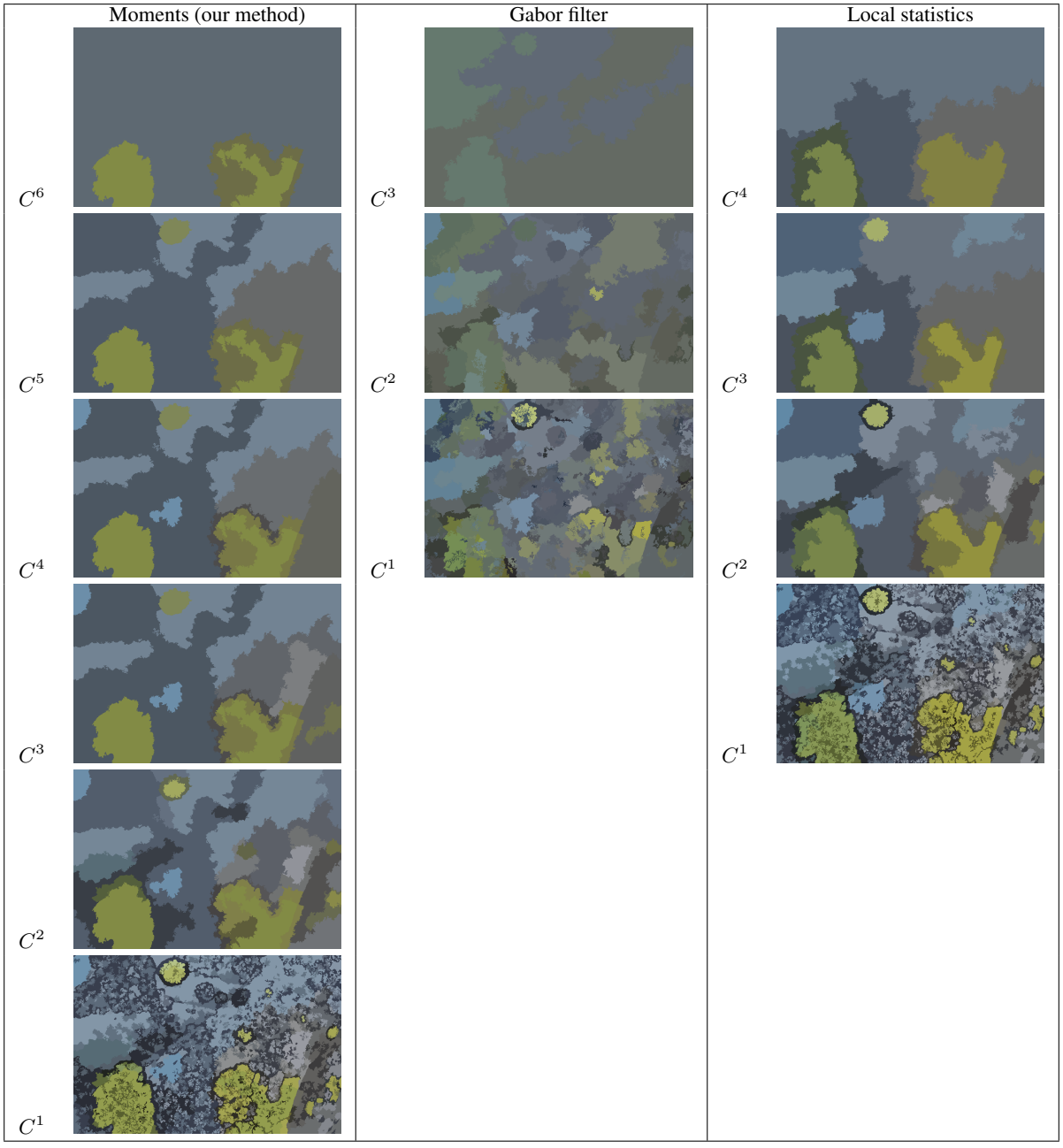


Figure 15: *Clustering results with different feature vectors for image Moss.*

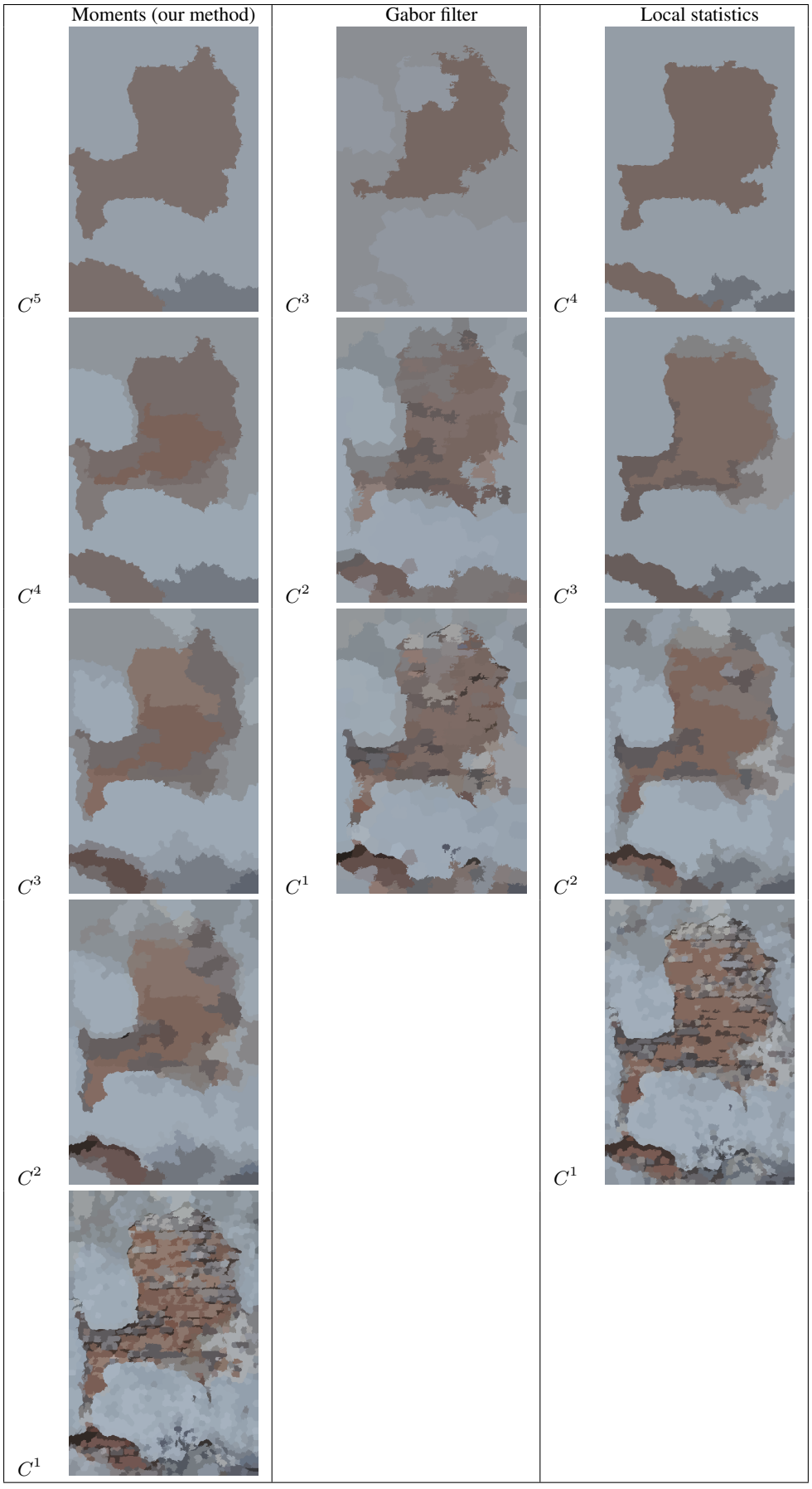


Figure 16: *Clustering results with different feature vectors for image Plaster_damaged.*

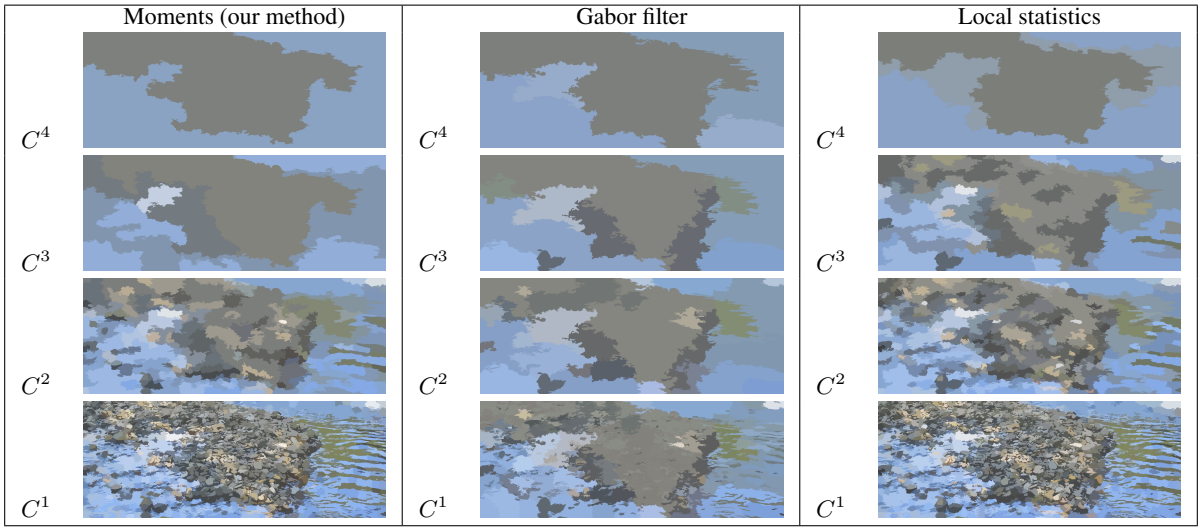


Figure 17: *Clustering results with different feature vectors for image River.*

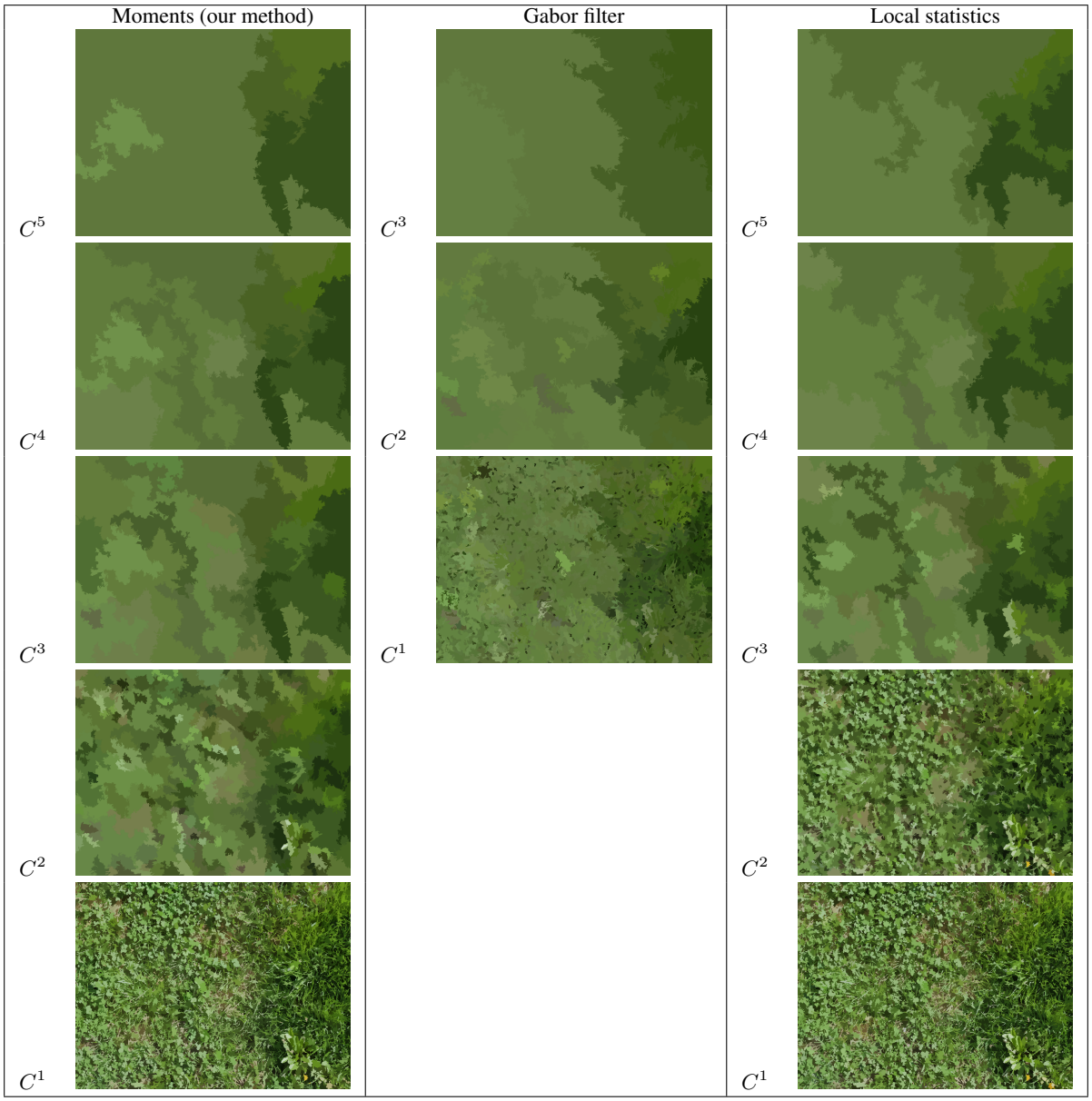


Figure 18: *Clustering results with different feature vectors for image Grass.*

Figure 19: *Clustering results with different feature vectors for image Tower.*

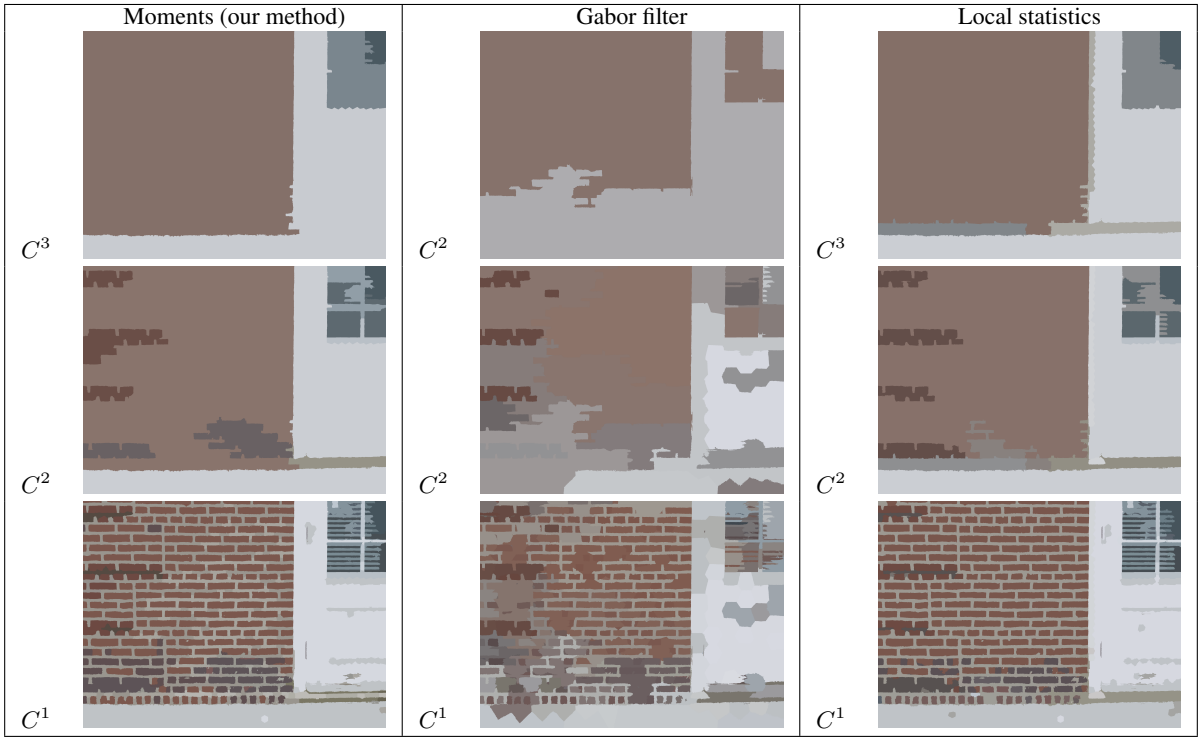


Figure 20: *Clustering results with different feature vectors for image Wall_with_door.*

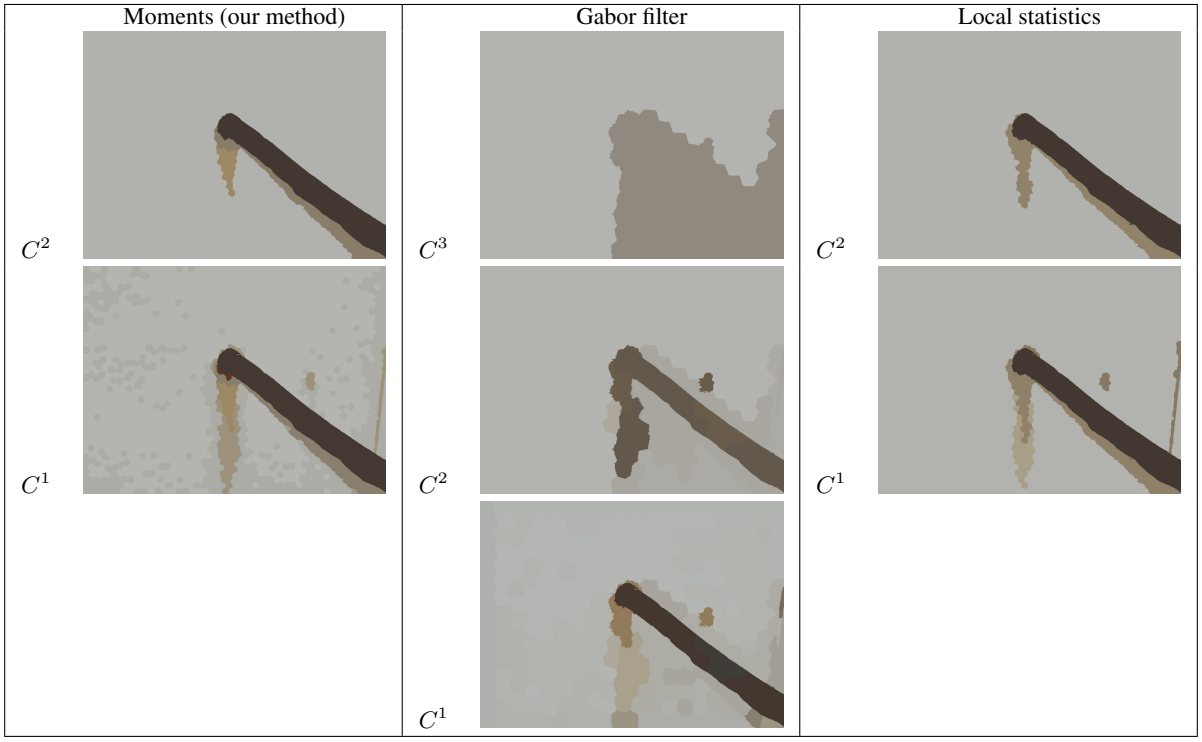


Figure 21: *Clustering results with different feature vectors for image Wall_with_pipe.*

6 Evaluation of the clustering

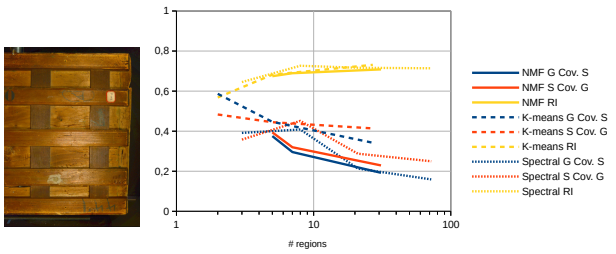
We compared our clustering based on nonnegative matrix factorization (NMF) with K-means and spectral clustering. However, these algorithms require the number of clusters to be supplied as inputs. We therefore extend them by using our NMF inspired heuristic to calculate the number of clusters.

Several qualitative comparisons are shown in figures 24 to 38. We show the clusters C^l for several levels l.

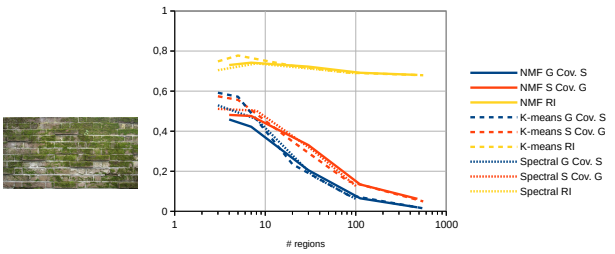
Quantitative comparisons, obtained as explained in section 3, are provided in figures 22 and 23.

We draw the following conclusions.

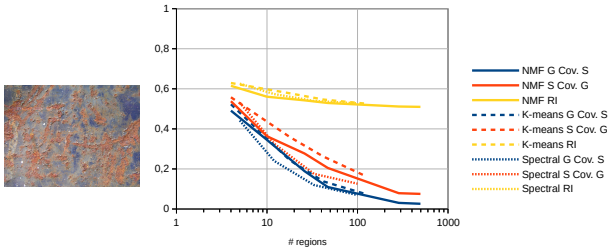
- In many cases, the different clustering methods perform similarly.
- In order to use the other methods, we still need to perform part of the NMF based algorithm. The NMF algorithm also gives us additional information (the F and G matrices) that is not evaluated here, or given by other methods. We feel this justifies the choice of NMF for our algorithm.
- Using NMF introduces a new tool without limiting users still able to use the other algorithms depending on the situation and constraints. For example, a user concerned with speed, but who does not need F and G, might choose K-means or spectral clustering.
- Finally, the novel NMF algorithm allows for texture metrics that don't use feature spaces.



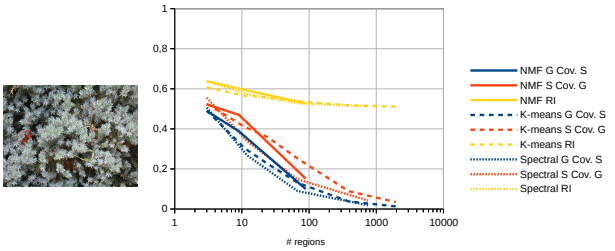
Boxtop



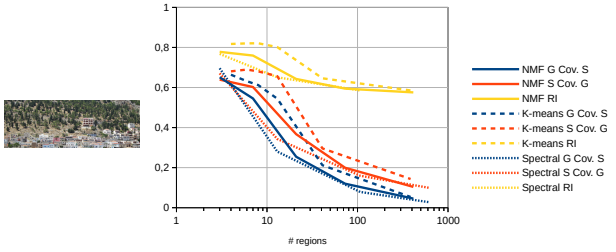
Brick_small_dirty



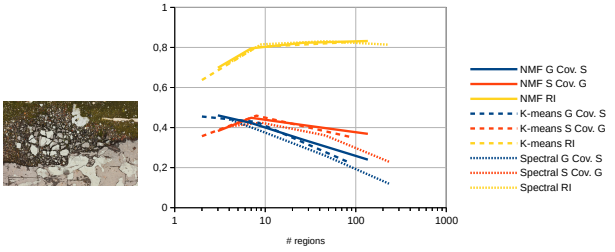
Bubblinng_rusty_metal



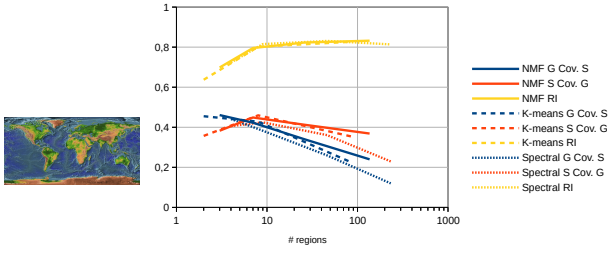
Bush



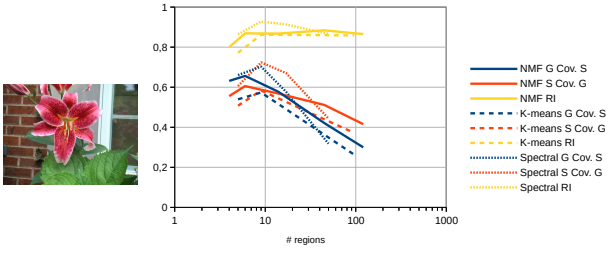
City



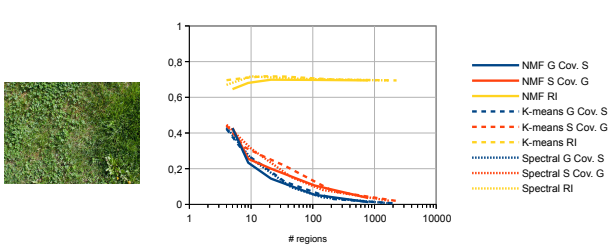
Concrete_floor_damaged



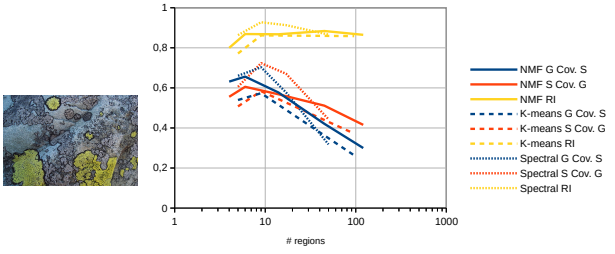
Earth



Flower

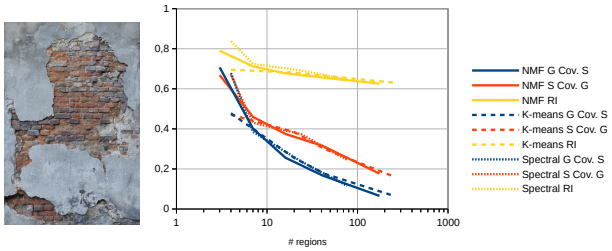


Grass

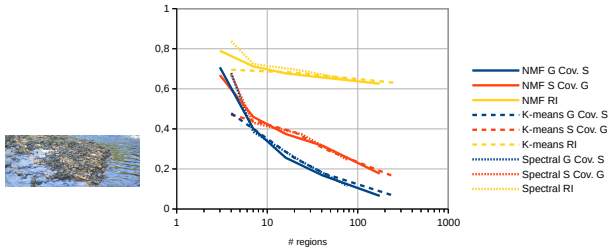


Moss

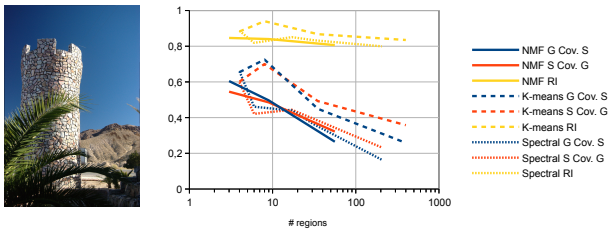
Figure 22: *Comparison of clustering methods, part 1/2. Note that each color represents a different quantitative measure that is not comparable to the other colors. Higher values represents better agreement between algorithm result and ground truth.*



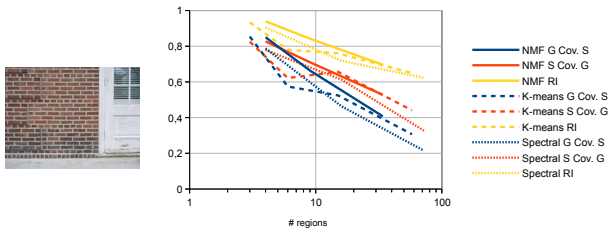
Plaster_damaged



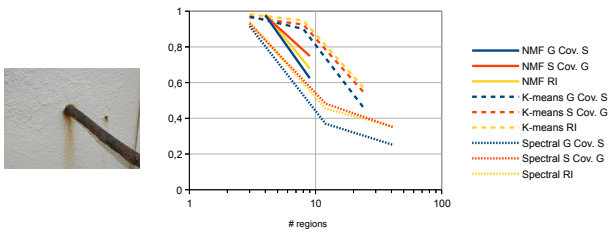
River



Tower



Wall_with_door



Wall_with_pipe

Figure 23: *Comparison of clustering methods, part 2/2. Note that each color represents a different quantitative measure that is not comparable to the other colors. Higher values represents better agreement between algorithm result and ground truth.*

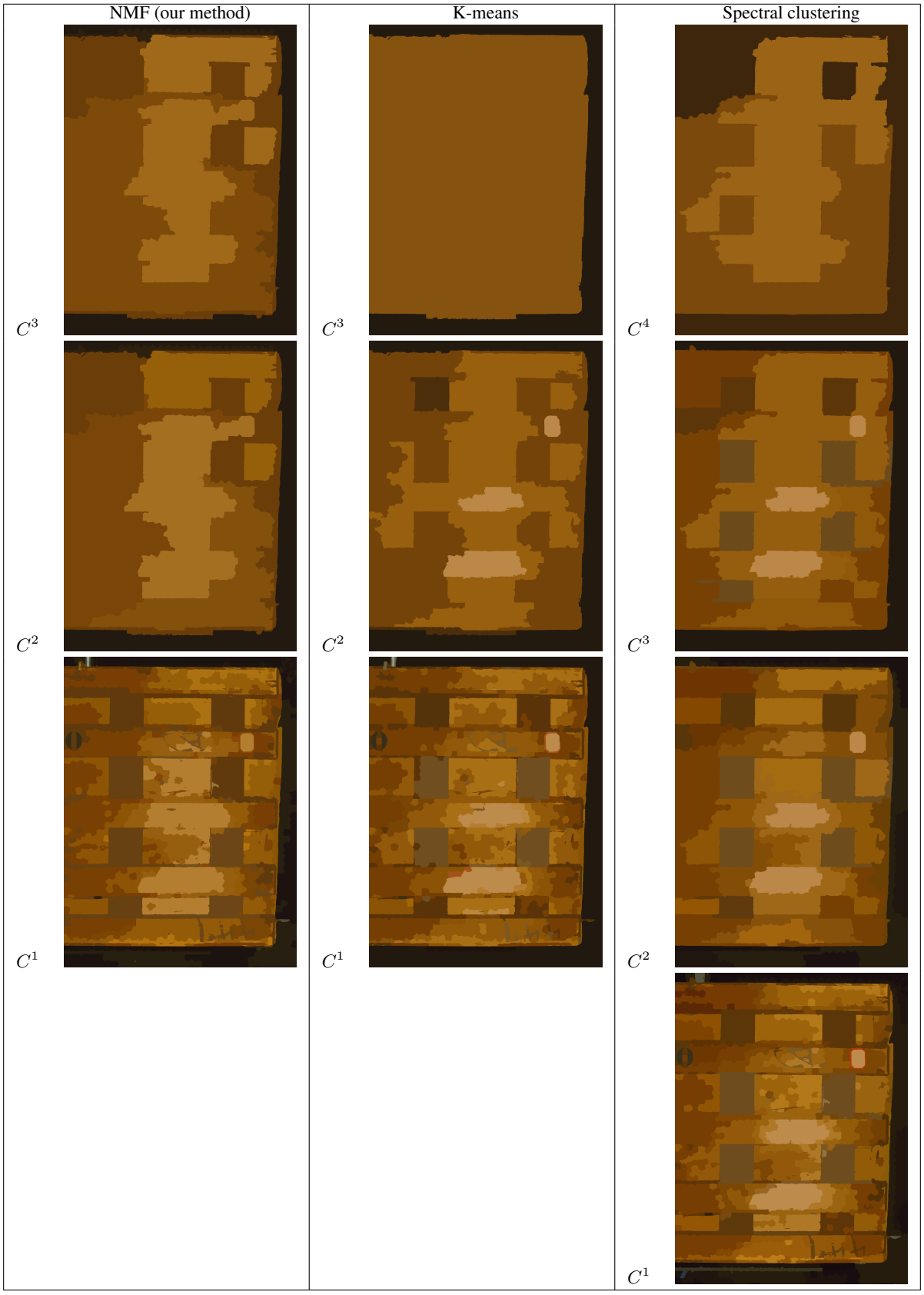


Figure 24: *Results with different clustering methods for image Boxtop.*

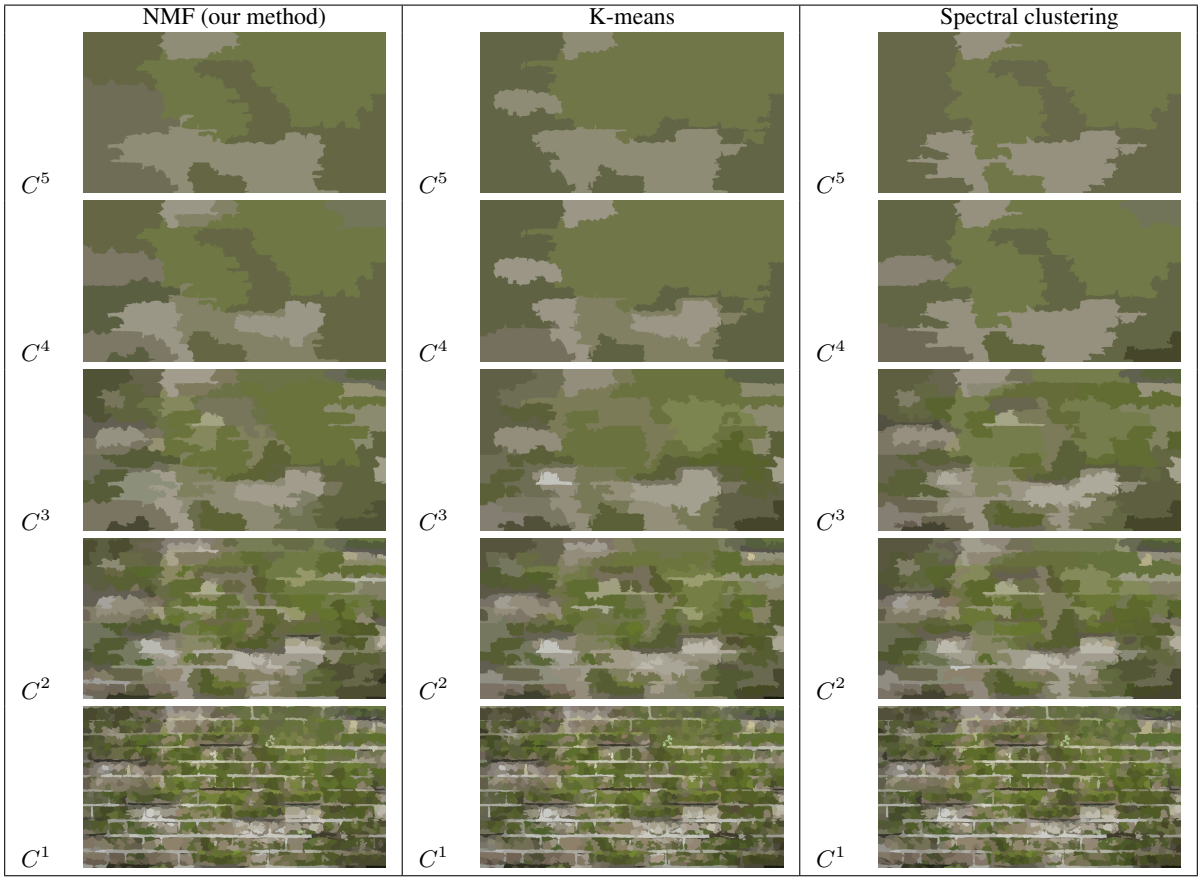


Figure 25: *Results with different clustering methods for image Brick_small_dirty.*

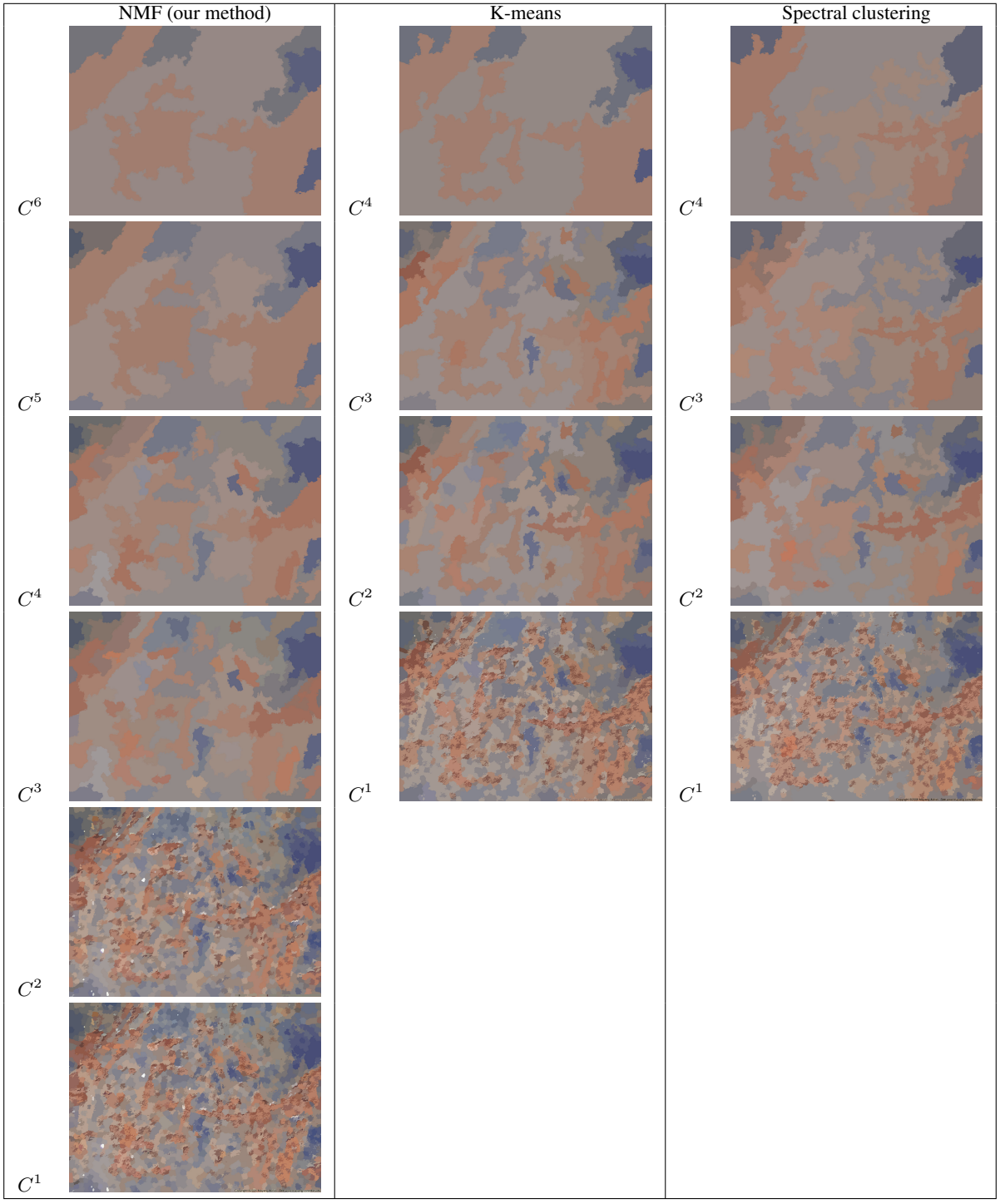


Figure 26: *Results with different clustering methods for image Bubbling_rusty_metal.*

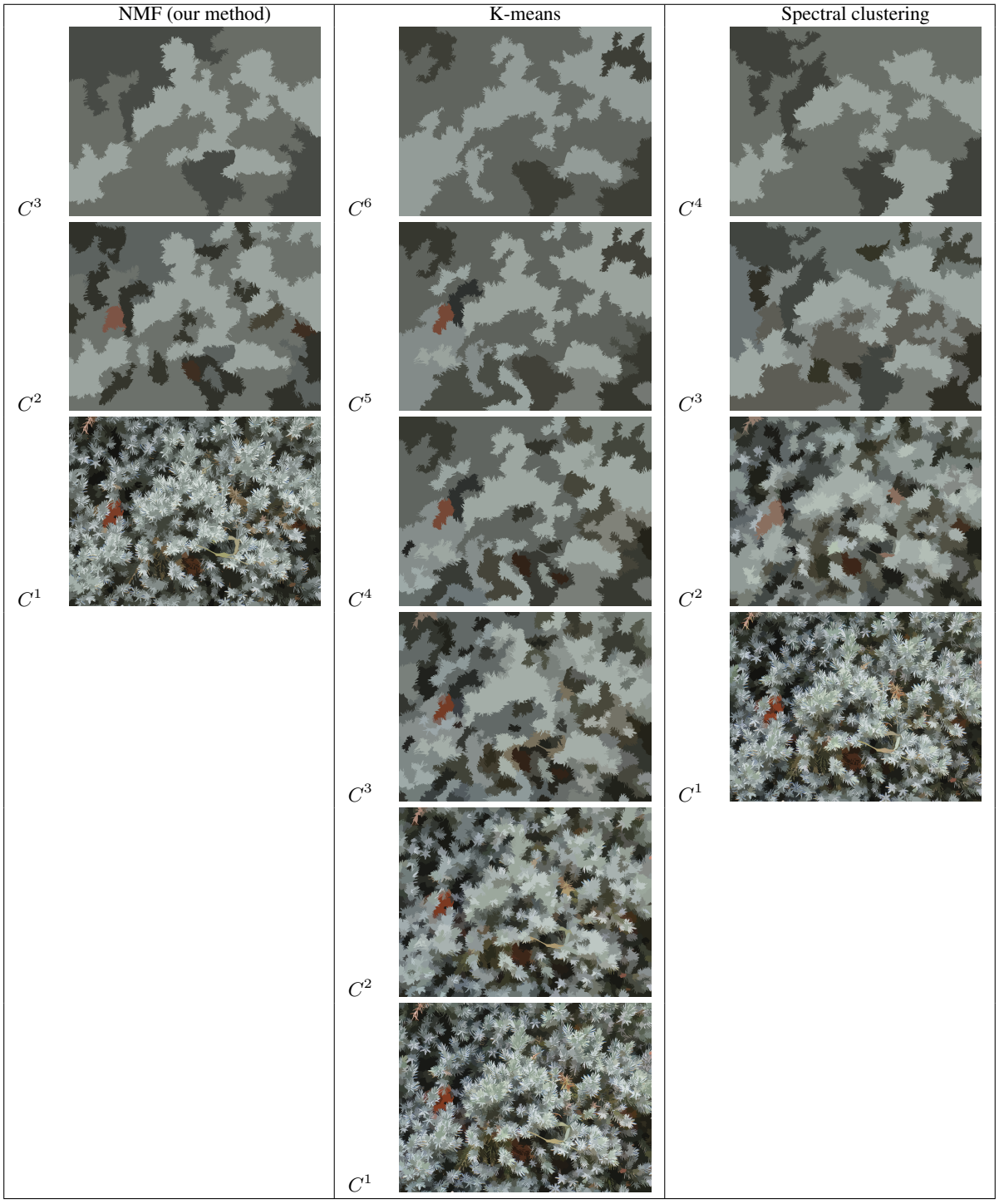


Figure 27: *Results with different clustering methods for image Bush.*

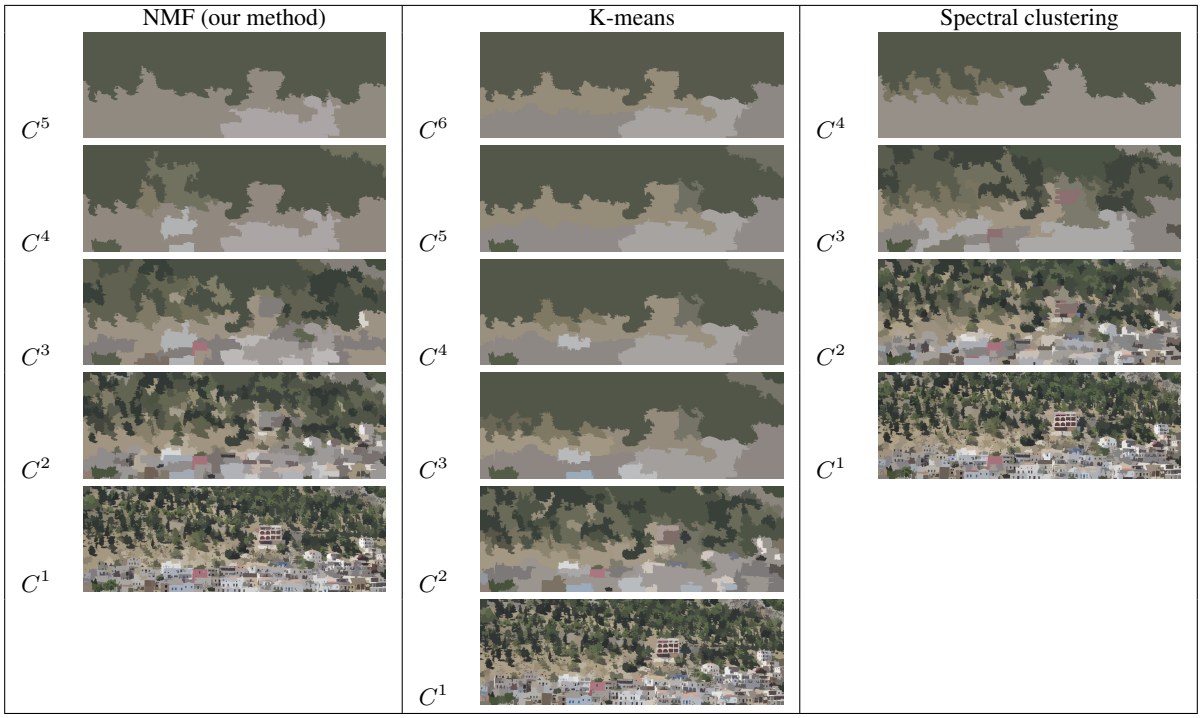


Figure 28: *Results with different clustering methods for image City.*

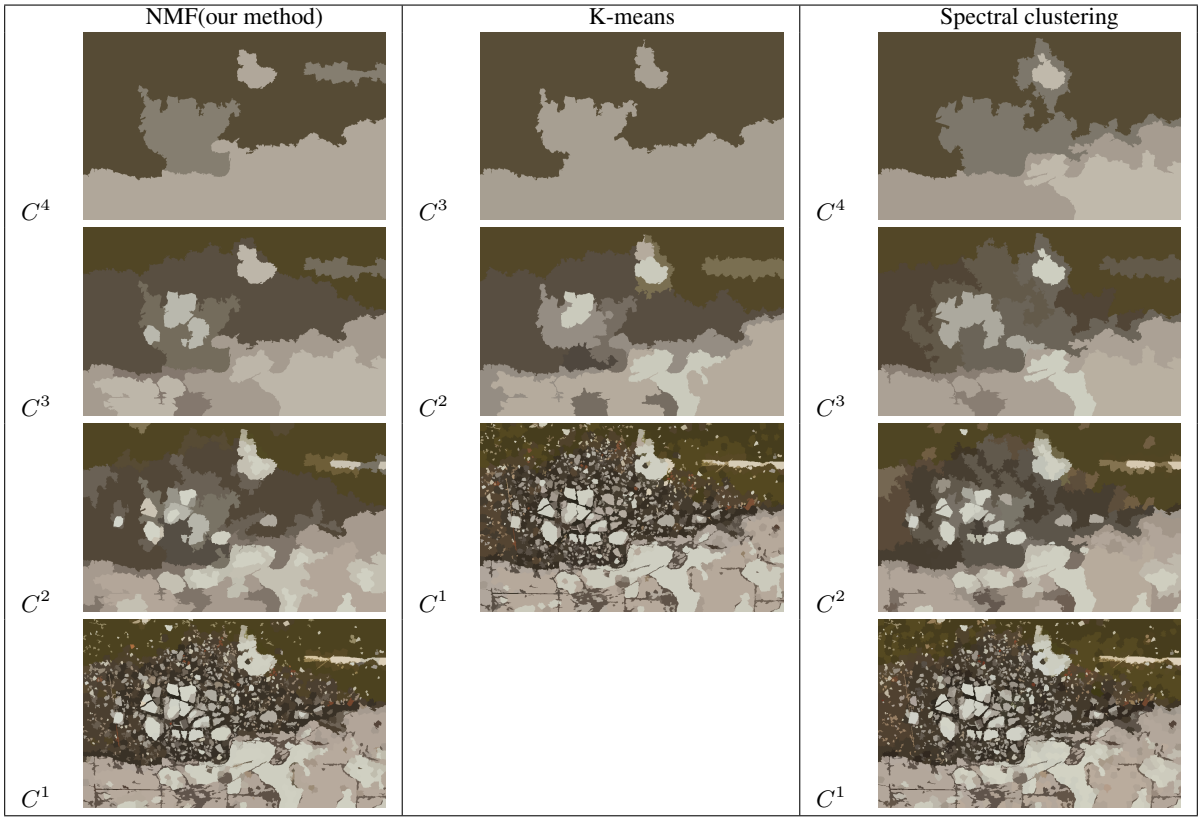


Figure 29: *Results with different clustering methods for image Concrete_floor_damaged.*

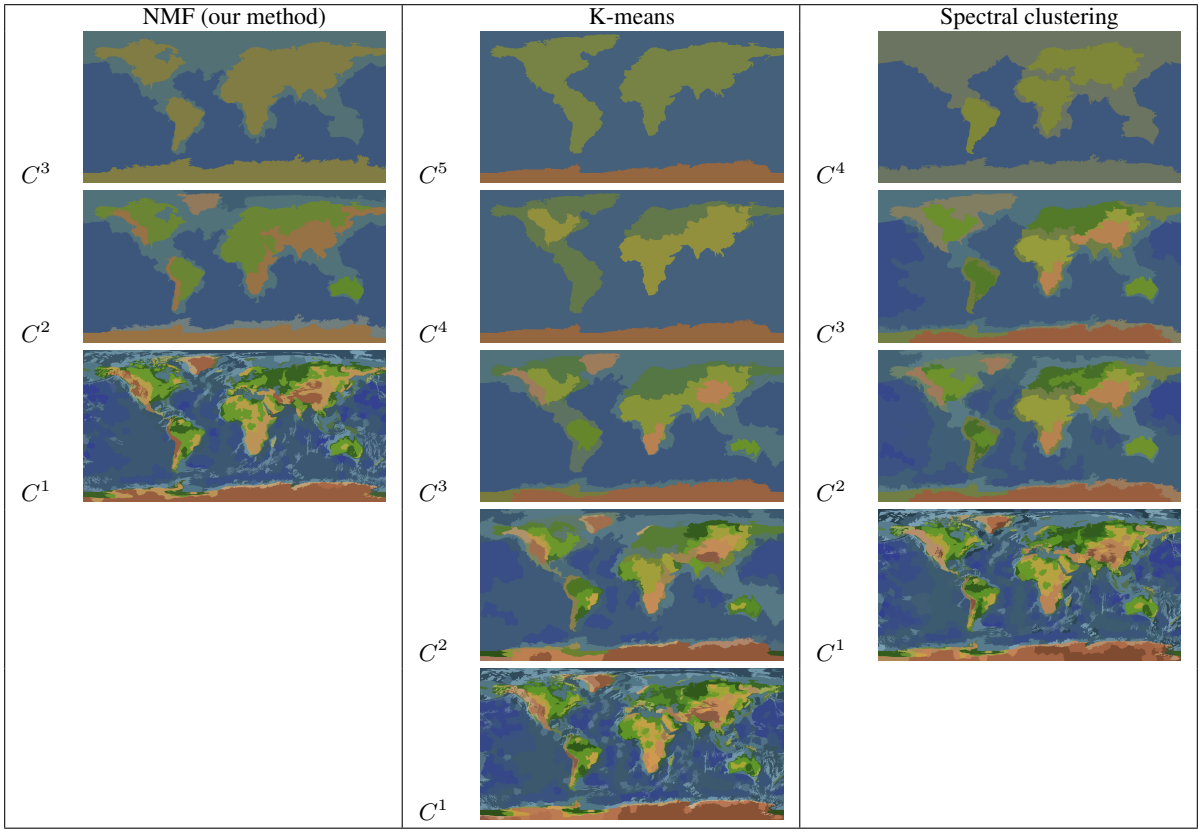


Figure 30: *Results with different clustering methods for image Earth.*

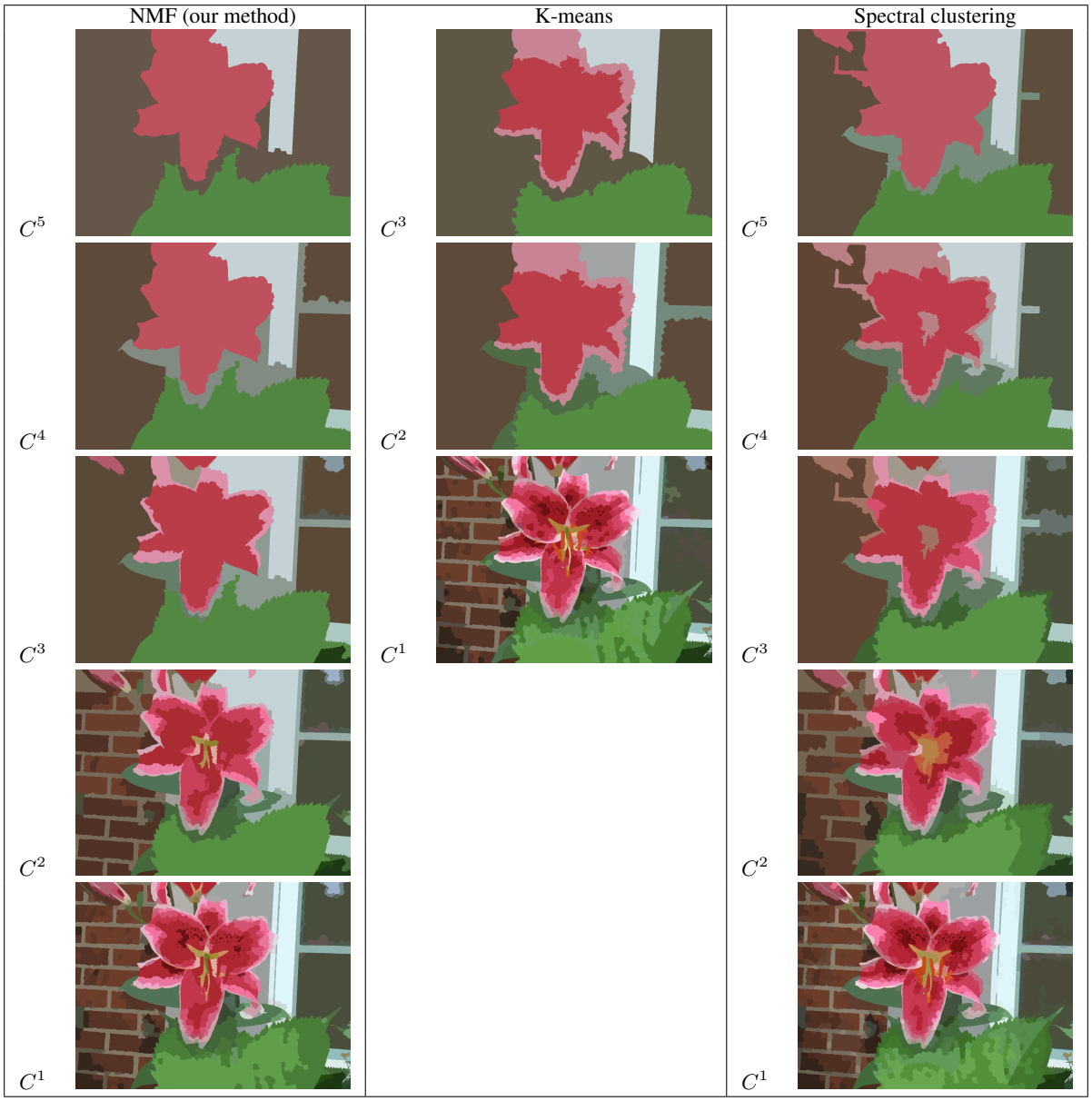


Figure 31: *Results with different clustering methods for image Flower.*

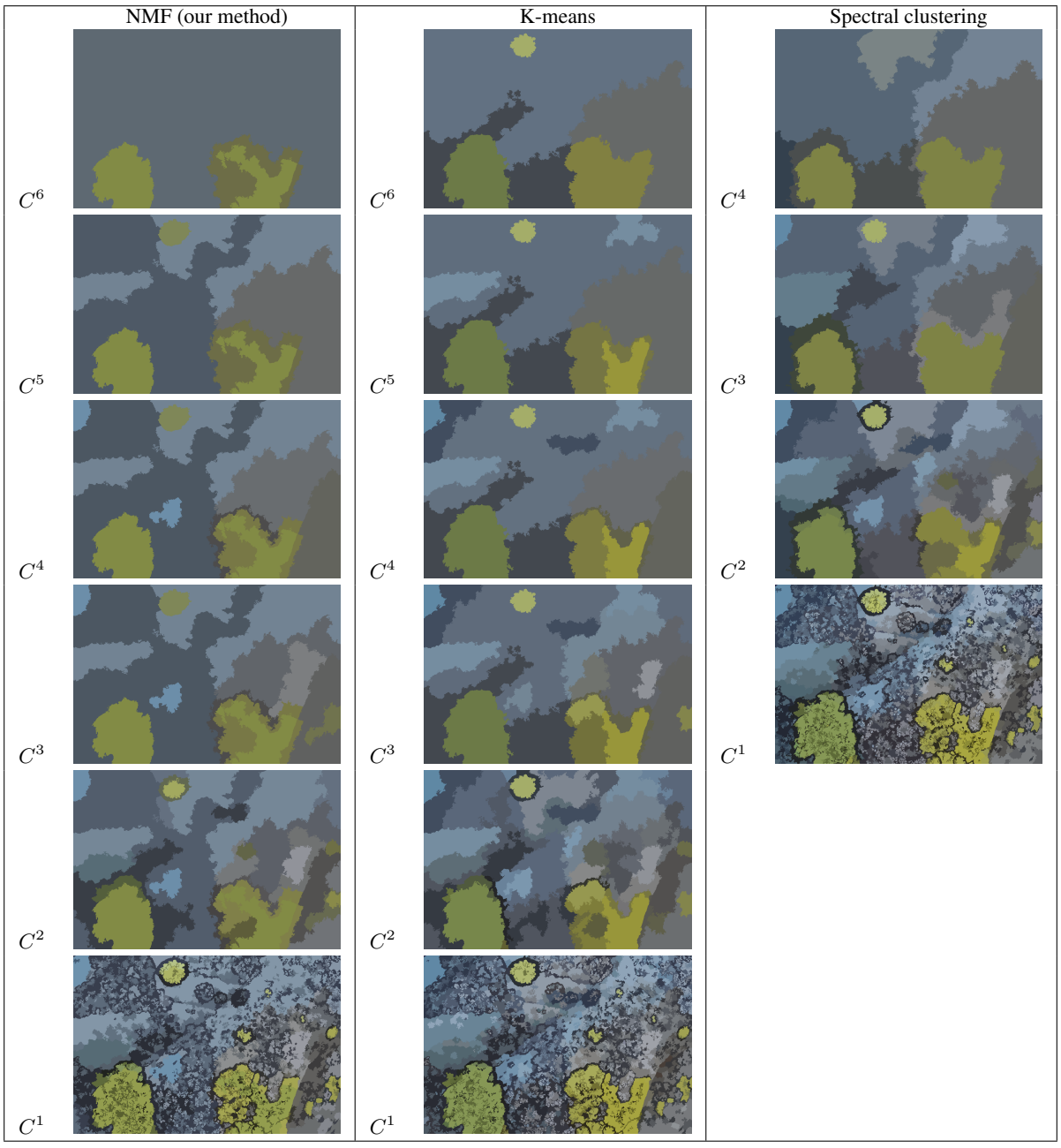


Figure 32: *Results with different clustering methods for image Moss.*

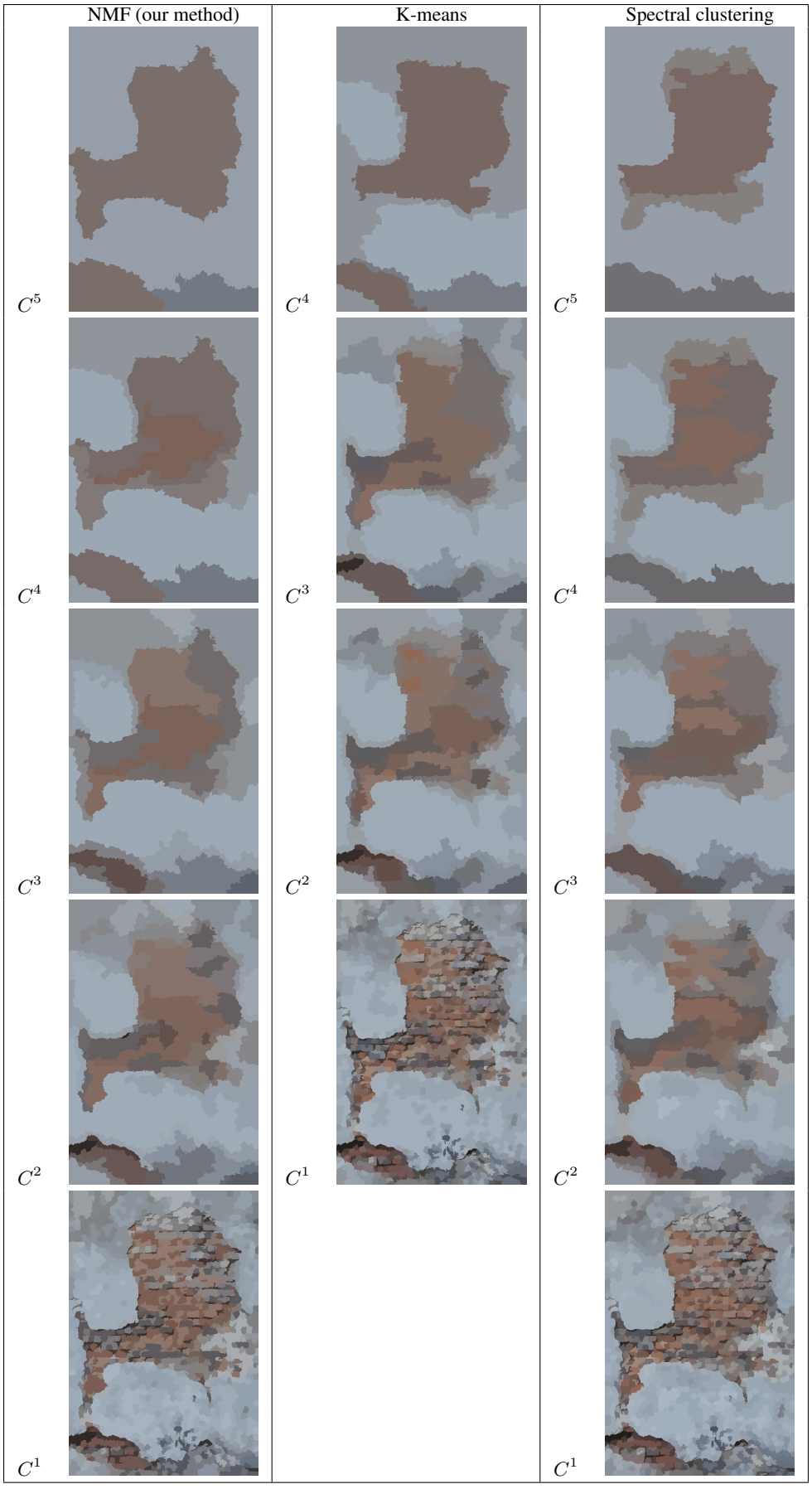


Figure 33: *Results with different clustering methods for image Plaster_damaged.*

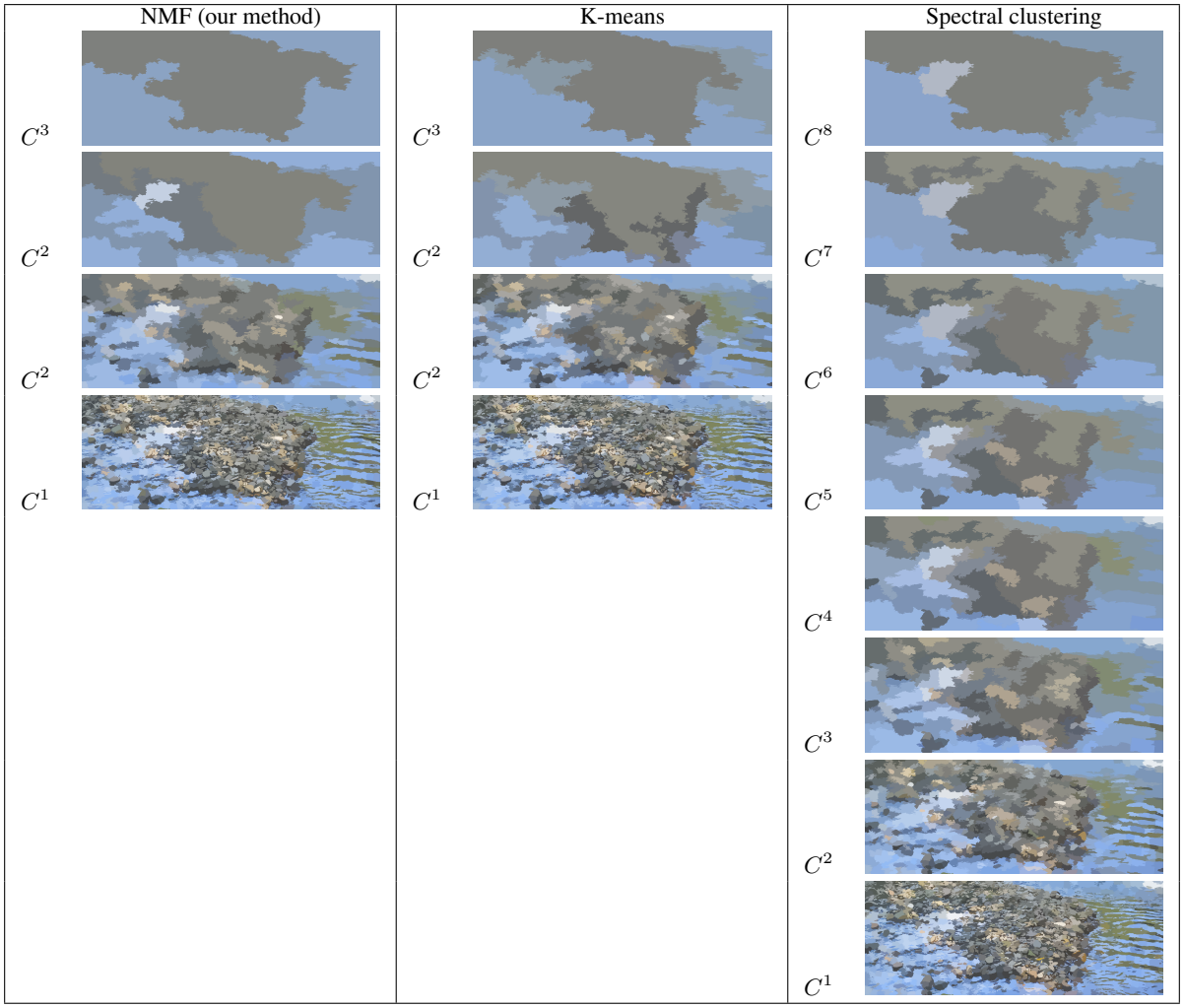


Figure 34: *Results with different clustering methods for image River.*

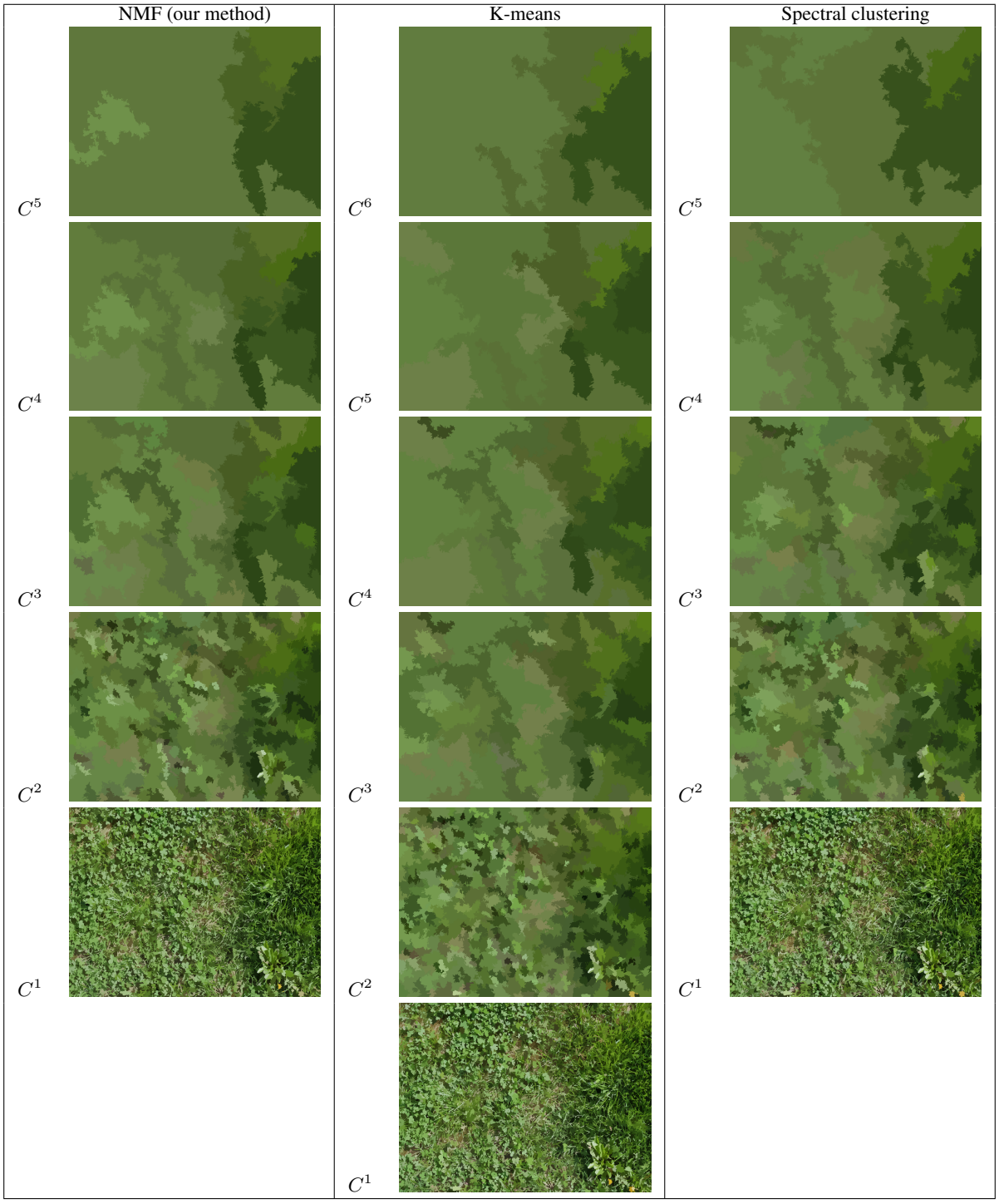


Figure 35: *Results with different clustering methods for image Grass.*

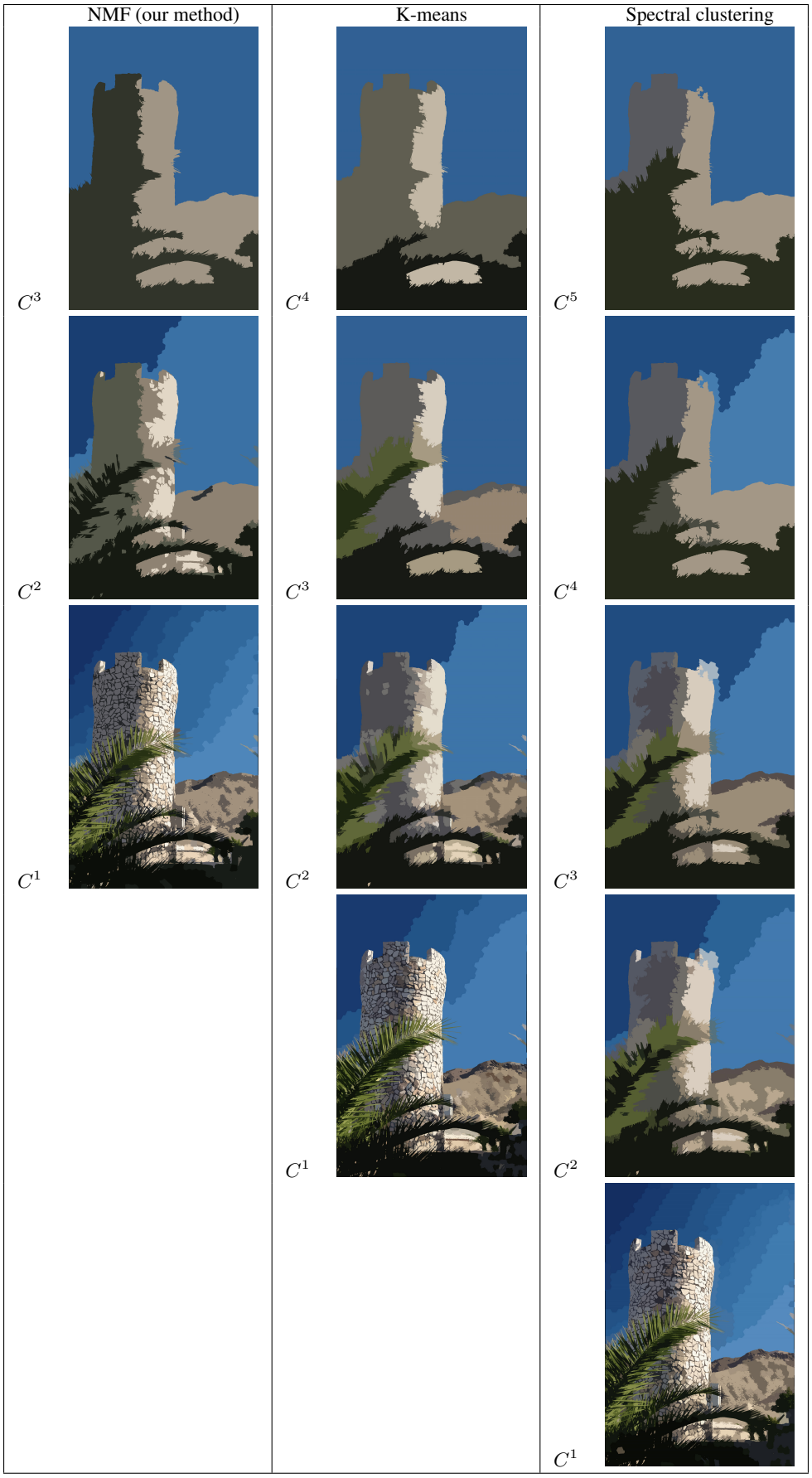


Figure 36: *Results with different clustering methods for image Tower.*

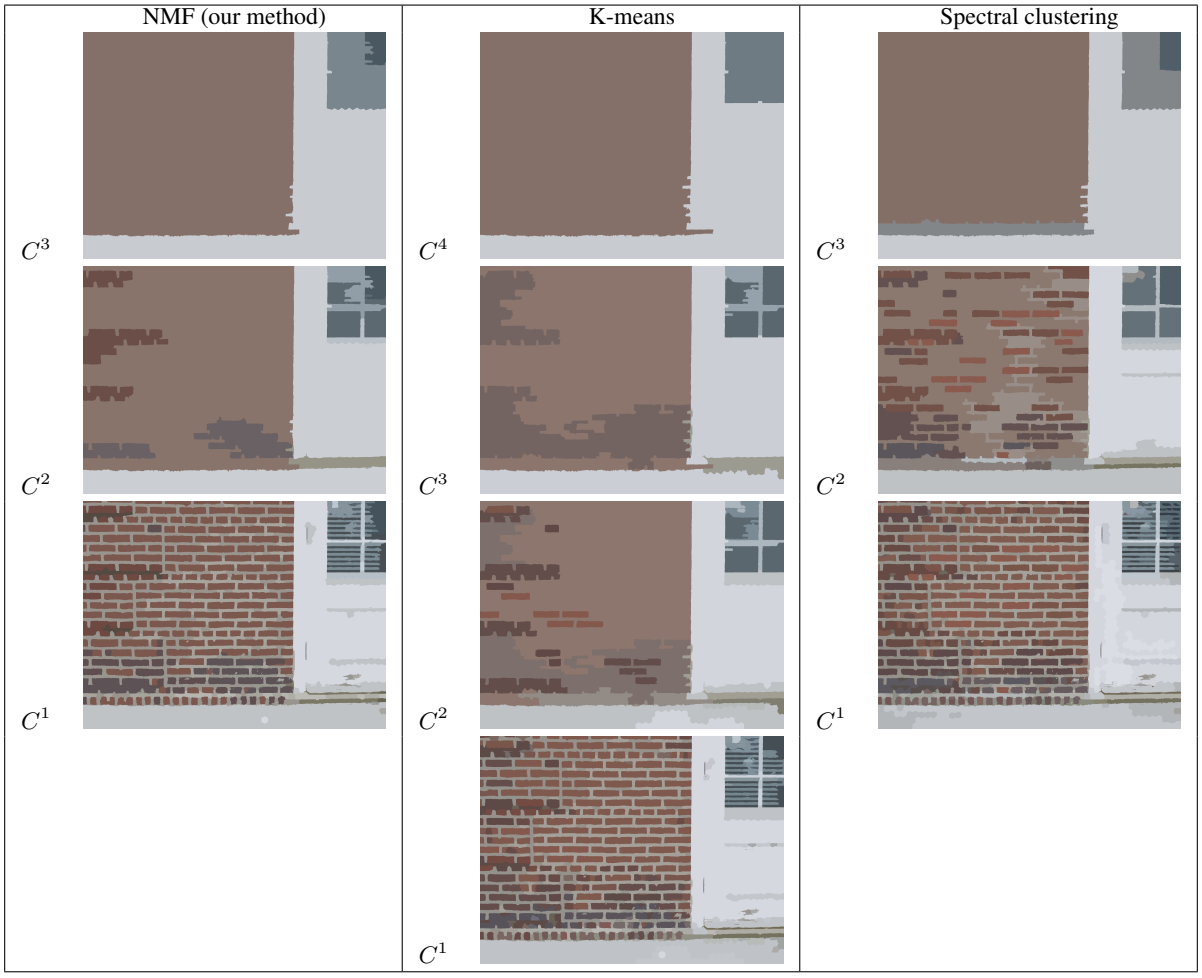


Figure 37: *Results with different clustering methods for image Wall_with_door.*

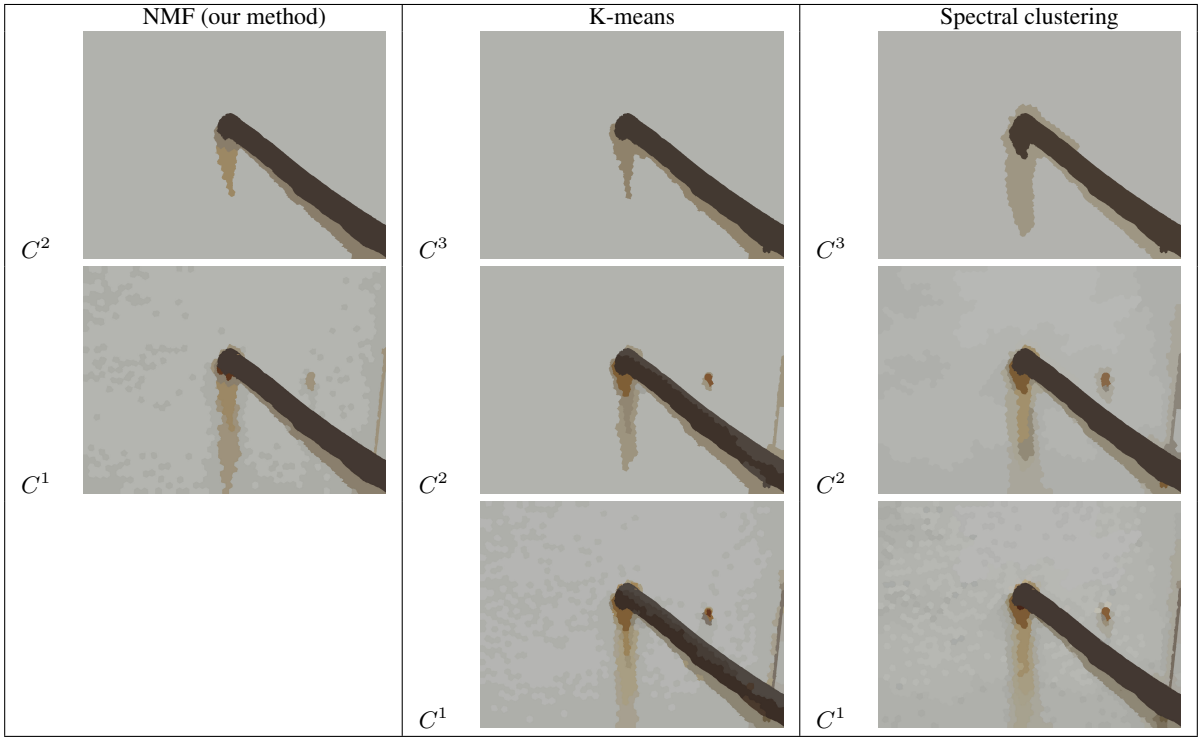


Figure 38: *Results with different clustering methods for image Wall_with_pipe.*

Acknowledgements

Images courtesy: Eli Lockerman (Tower), Mayang's Textures (Bubbling_rusty_metal), www.textures.com (Plaster_damaged, Brick_small_dirty, Concrete_floor_damaged, Moss), maps.jpl.nasa.gov (Earth), other images taken by authors.

References

ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SUSSTRUNK, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 11 (November), 2274–2282.

ARBELÁEZ, P., MAIRE, M., FOWLKES, C., AND MALIK, J. 2011. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 5 (May), 898–916.

LOCKERMAN, Y. D., XUE, S., DORSEY, J., AND RUSHMEIER, H. 2013. Creating texture exemplars from unconstrained images. In *Proceedings of the 2013 International Conference on Computer-Aided Design and Computer Graphics*, IEEE Computer Society, CADGRAPHICS '13, 397–398.

MOUNT, D. M. 2010. ANN programming manual. Tech. rep., University of Maryland, College Park, Maryland.

MOVELLAN, J. R. 2002. Tutorial on gabor filters. *Open Source Document*.