Scale-Space Approach for Color Image Segmentation

Rémi Mégret

Scale-Space Approach for Color Image Segmentation

R. Megret

ESI/LAPS (UMR CNRS 5131), ENSEIRB, Talence, France

Abstract

This paper presents a new method for color image segmentation based on a scale-space clustering of the image pixels. Unlike standard image scale-spaces, which smooth the images, this approach consider an augmented space which combines spatial and color dimensions. The clustering relies on the mean-shift algorithm to find the modes of the position and color distribution of the pixels. In order to produce a non-hierarchical segmentation, we also present a method to prune the computed hierarchy by suppressing nodes that are not stable enough.

1. Introduction

Image segmentation is a central task for image and video analysis, especially in the context of multimedia indexing and retrieval. Indeed an image is made of several distinct visual structures, that are combined to produce a global perception of that image. Color is one one the most important low-level feature that can be used to extract some information about the visual structure of an image, such as homogeneous regions, that are most of the time related to objects or parts of objects.

There has been a lot of work done in color image segmentation, that Lucchese and Mitra [7] classify into three broad categories. Feature-based techniques segment the image by finding some clusters in a feature space (color or texture space) that do not take spatial relationships into account. Those methods are based on clustering pixel features, or histogram thresholding. Image-based techniques on the other hand try to satisfy both feature homogeneity and spatial compactness by explicitly manipulating the pixels. This category contains split-and-merge, region growing and edge-based approaches. Physics-based methods are more application specific, as they model the causes that may produce color variations.

There is an intermediate approach between feature- and spatial-based approaches, where color and spatial constituencies are enforced simultaneously, by working in a space that combines pixel positions and color features. This approach was in particular proposed by Comaniciu and Meer [3], with a mean-shift based color segmentation method. This method was optimized by Bailer et al. [1] to segment consecutive images of a video sequence. In this paper, we propose a new method based on the same idea on combining color and spatial features in order to produce a multi-scale segmentation of a color image.

In section 2 we present the mathematical and algorithmic foundations of clustering by mode seeking, which is the technique we apply to color image segmentation. In section 3 we discuss the method of Comaniciu and Meer [3] which is based on this framework, but operates at fixed scale. In section 4 we introduce the scale-space formalization for color image segmentation. Finally, in section 5 a new method to simplify the hierarchical segmentation is proposed.

2. Clustering by mode seeking

In this section, we recall some mathematics about clustering by mode seeking, which we will use for color and position clustering.

2.1. Problem

The general problem of density based clustering can be stated as follows:

> Given a set of n samples $\mathbf{x}_i$ in $\mathbb{R}^N$, group together samples that form dense clusters.

The mode seeking approach is based on the density estimate $f_K(\mathbf{x})$ of those samples using a kernel estimator:

$$f_K(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x}-\mathbf{x}_i), \tag{1}$$

where K is a kernel that defines the neighborhood over which the contribution of samples is integrated in order to estimate a local density. Each mode of this density function corresponds to a point that is denser, compared to its neighbors, and thus can be considered the center of a cluster. The watershed of $-f_K$, associates to each mode a natural influence zone, which is bounded by the influence of their modes.

The clustering problem can therefore be formalized by associating a cluster to each mode of f_K. A point belongs to a cluster if the trajectory computed by gradient ascent from this point converges to the mode of this cluster. The problem hence boils down to finding for each sample the mode it belongs to using gradient ascent.

2.2. Gradient ascent and fixed scale clustering

The mean-shift algorithm provides us with a way to accomplish gradient ascent of the density function f_K defined as in equation 1. For the sake of clarity, we will restrict ourselves to isotropic kernels, which are defined by their profile k and a scale parameter h:

$$K_h(\Delta\mathbf{x}) = \frac{c_{k,N}}{h^N}\; k(\|\Delta\mathbf{x}\|^2/h^2), \qquad (2)$$

where $c_{k,N}$ is a normalization constant depending on the profile k and the dimensionality N of $\mathbf{x}$.

If we introduce G a second kernel whose profile g is

$$g(x) = -k'(x), \qquad (3)$$

then the mean-shift vector defined as

$$\mathbf{m}_h(\mathbf{x}) = \frac{\sum_i \mathbf{x}_i g(\|\mathbf{x}-\mathbf{x}_i\|^2/h^2)}{\sum_i g(\|\mathbf{x}-\mathbf{x}_i\|^2/h^2)} - \mathbf{x} \qquad (4)$$

is collinear with the gradient ∇f_{K_h} of the density of interest:

$$\mathbf{m}_h(\mathbf{x}) = h^2 \frac{c_{k,N}}{2c_{g,N}} \frac{\nabla f_{K_h}(\mathbf{x})}{f_{G_h}(\mathbf{x})}. \qquad (5)$$

The gradient ascent algorithm is therefore:

1) Initialize moving points $\mathbf{y}_i^0$ to the original point $\mathbf{x}_i$
2) Repeat steps 3-5 until convergence of $\mathbf{y}_i$
3) Compute the mean-shift vector $\mathbf{m}_h(\mathbf{y}_i^k)$
4) Apply the shift vector to the current position: $\mathbf{y}_i^{k+1} \leftarrow \mathbf{y}_i^k + \mathbf{m}_h(\mathbf{y}_i^k)$
5) If the density did not increase ($f(\mathbf{y}_i^{k+1}) < f(\mathbf{y}_i^k)$), then divide the shift vector by 2 and repeat step 3a.

This algorithm produce for each original point $\mathbf{x}_i$ a convergence point $\mathbf{y}_i^\infty$. This point should be a mode of the distribution. The convergence point may not be a mode in some degenerate cases (local minimum, saddle point). These situations are not met in practical situations and could be resolved by adding a small perturbation.

According to the definitions we drew at the beginning of this section, clustering the points amounts to grouping points associated to identical modes $\mathbf{y}_i^\infty$. From a practical point of view, points are declared to converge at the same point if the distance between two $\mathbf{y}_i^\infty$ is less than a given threshold ε.

2.3. Scale-space clustering

In the previous developments the scale parameter h was fixed to an arbitrary value. By allowing this continuous parameter to change, we can build a scale-space $(\mathbf{x},h) \mapsto f_{K_h}(\mathbf{x})$, which considers in a unified manner densities computed for all possible values of h.

We are interested in the position of modes of f_{K_h} for h fixed, but want to consider how these modes evolve when h changes. Leung et al. [5] showed that the position of modes form simple curves when h changes continuously. In order to build the multi-scale structure of the modes, modes can therefore be linked across scale by considering that a mode at a given scale is close to the corresponding mode when the scale changed slightly.

The general events that can occur when the scale increases can be classified [6] as creation (a mode appears at a scale, but no corresponding mode can be found on a lower scale), splitting (a node at a given scale can be linked to several modes at a higher scale), destruction (no corresponding mode can be found on a higher scale) and merging (several modes at a given scale can be associated to a mode at a higher scale).

The algorithm to build a hierarchical structure is described now.

1) Associate a mode to each original point, for h close to 0.
2) Repeat until h is above a fixed threshold:
   2.1) Compute the new position of each mode by mean-shift iterations until convergence
   2.2) Merge modes closer than the threshold ε
   2.3) Increase the scale: $h \leftarrow h\, dh$

This algorithm leads to a tree structure, where leaves correspond to the original points, and each node corresponds to a piece of curve in the scale-space associated to the displacement of a given mode across scale. Each node possesses an existence interval, comprised between an apparition scale and a disappearance scale. The apparition occurs either at initialization at scale $h = 0$, or when two existing modes merge. The disappearance occurs when the mode associated with the node merges with another mode.

Each node of the tree corresponds to exactly one mode of f_{K_h} for any scale in its existence interval. By cutting the tree for any given scale h we can therefore get a subset of the set of modes that would be computed by the fixed scale method. This difference is good for the robustness of the method. Indeed, as scale increases details should be smoothed out, so that created or split modes are not wanted, and we expect the number of modes to decrease. This property is true for one-dimensional scale-spaces with the gaussian kernel, but not for higher-dimensional spaces or other kernels.

Exceptions to this rule appear nevertheless only is specific configurations, as pointed out by Lindeberg [6] where the density function is rather elongated, and the smoothing cuts it in two parts. Some modes may also be created because of numerical instabilities: if the convergence tests stop the modes before they reach their theoretical location and if the threshold on the distance for modes fusion is too small, we may find several modes when there is theoretically only one. By imposing a hierarchical structure, we restrict ourselves to modes that can be linked to modes existing at the lowest scale, thus decreasing such phenomenons.

The previous algorithms and methods are very general, we will now see how they can be used in the context of color image segmentation.

3. Color Image Segmentation using Mean-shift

3.1. Position and color clustering

Comaniciu and Meer [3] introduced a color segmentation method based on the clustering of pixels in a position and color space. Instead of modeling the distribution of colors and positions with a parametric model, they use the non-parametric approach we detailed in the previous section: each mode of the colors and positions density corresponds to a cluster of pixels that are both close in position and color. Instead of applying parametric models to the distribution of colors and positions as was done in the Pfinder project [10], they use the non-parametric approach we detailed in the previous section: each mode of the colors and positions density corresponds to a cluster of pixels that are both close in position and color.

This density is expressed as:

$$H_{\beta,r}(\mathbf{p},\mathbf{c}) = \sum_i k\left(\left\|\frac{\mathbf{p}_i-\mathbf{p}}{r}\right\|^2 + \left\|\frac{\mathbf{c}_i-\mathbf{c}}{\beta}\right\|^2\right), \qquad (6)$$

where $(\mathbf{x}_i, \mathbf{c}_i)$ is the position and color of pixel number i in the image. We can indeed verify that this function corresponds to the general form of equation 1 required to apply the mode-seeking clustering approach. The considered space is a position and color space, with $\mathbf{x} \in \mathbb{R}^5$ being the concatenation of $\mathbf{p} \in \mathbb{R}^2$ and $\mathbf{c} \in \mathbb{R}^3$:

$$\mathbf{x} = \left(\frac{\mathbf{p}}{r}, \frac{\mathbf{c}}{\beta}\right) \qquad (7)$$

Using the algorithms we presented previously, the authors can cluster the pixels by associating each pixel with a mode of density $H_{\beta,r}$ in the position and color space. The segmentation is done for a fixed scale, where parameters β and r are tuned manually.

3.2. Post-processing

The previous clustering does not suffice for practical cases, which is why Comaniciu and Meer added an additional post-processing step. Once the modes have been computed, they are grouped based on a threshold on their respective distance in position and color space. Modes whose position difference is less than r and color difference less than β are merged. This grouping is transitive. Modes which are close enough are linked, and the connected components of the resulting graph define how the modes are merged.

This step is necessary because regions of an image may have uniform colors, but they generally have different sizes. Therefore their distribution in the position and color space looks like a set of elongated blobs, that extend along the position dimensions according to the region sizes, and along the color dimensions according to the color variance. The spatial scale r used to smooth the distribution being fixed globally, it can not match both large and small regions (see Figure 1). For small regions, r risks to be too large, and make contributions from one region to mix with another region, resulting in their fusion. For large regions, r may be too small: although the density can be high over the spatial support of the region, several modes can be present, which results in over-segmentation.

The modes grouping post-processing addresses this issue, by grouping the modes that are supposed to correspond to the same large region. The spatial scale r should then be chosen close to the size of the smallest regions to segment.

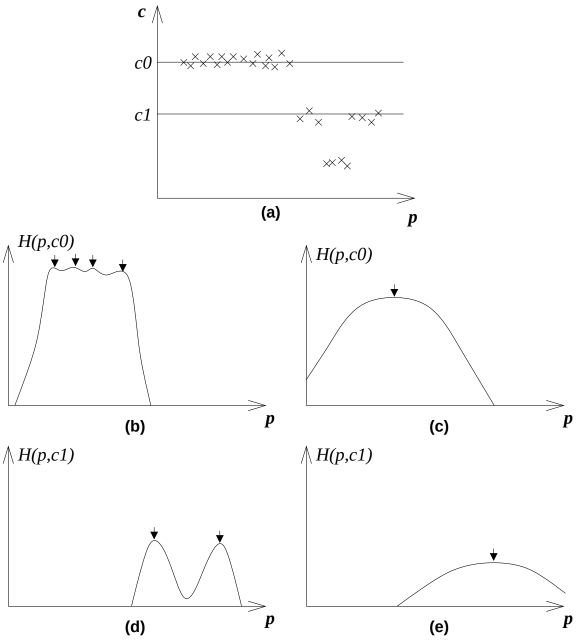


Figure 1: *The same spatial bandwidth r can not apply simultaneously to small and large regions. (a): Samples $\mathbf{x}_i$ used to compute the position and color density $H_{\beta,r}(\mathbf{x},\mathbf{c})$, for a small r (left column), and for a larger r (right column). (b) and (c): Density $H_{\beta,r}(\mathbf{x},\mathbf{c})$ for $\mathbf{c}=\mathbf{c}_0$, which is the color of the large region; multiple modes are found when r is small. (d) and (e): Density $H_{\beta,r}(\mathbf{x},\mathbf{c})$ for $\mathbf{c}=\mathbf{c}_1$, which is the color of two small regions; only one mode is found when r is large.*

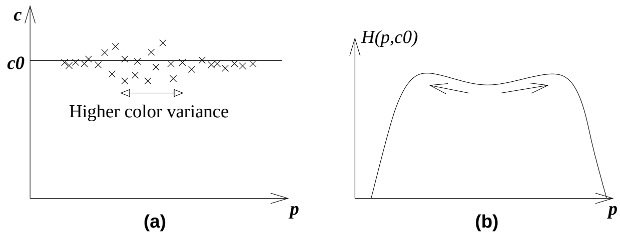


Figure 2: *Case where the modes associated to one homogeneous region are split apart by the gradient ascent.*

Small regions will be associated one mode directly. Larger regions will be associated several modes, that will be merged in the post-processing step. This is based on the heuristic assumption that the kernel smoothing shifts the modes toward the center of the homogeneous regions. While this is true close to the boundaries on the region, because of the contrast with the exterior, we can find cases where the interior of the region does not verify this property. Figure 2 represents a region whose color variance is not uniform, and which makes the modes to get far away instead of coming close to each other. This problem occurs because the modes fusion does not take into account the density function, but only the modes locations.

4. Scale-space Color Image Segmentation

Defining an appropriate scale for segmentation is a difficult task, as we have seen that mode estimation alone cannot handle all region sizes correctly. The post-processing grouping of modes is one possibility to alleviate this problem, although is still requires to fix the scaling parameters beforehand. It this section, we propose a new method based on the same position and color density approach, but embedded in a scale-space that allows us to handle all region sizes, without the need of the post-processing step.

4.1. Local Color Distribution

Before we detail the algorithm itself, we now replace the approach of clustering in the position and color space in a more general context, to get some insight on what kind of visual structures the methods we present depend on.

A local color distribution can be associated to each position $\mathbf{p}$ in the image. This distribution represents the color distribution in a neighborhood of $\mathbf{p}$, and is sometimes referred as *local color histogram*. A general formulation was proposed by Kœnderink and van Doorn [4]:

$$H_{\sigma,\beta,r}(\mathbf{p},\mathbf{c}) = \sum_i K_r\left(\mathbf{p}_i - \mathbf{p}\right) K_\beta\left(L_\sigma(\mathbf{p}_i) - \mathbf{c}\right), \qquad (8)$$

where

- $\mathbf{p}$ is the position around which the local color distribution $\mathbf{c} \mapsto H(\mathbf{p},\mathbf{c})$ is computed. $\mathbf{p}_i$ is the pixel number i in the image.
- K_r and K_β are smoothing kernels (K_β is a normal kernel in [4]).
- σ (internal scale) is the bandwidth of a gaussian filter applied to the original color image I to remove unnecessary spatial details. L_σ is the result of this smoothing. σ is characteristic of the level of resolution that is used to analyze the image.
- r (external scale) represents the size of the spatial neighborhood around $\mathbf{p}$.
- β (tonal scale) represents the smoothing bandwidth along the color dimension. This parameter defines the characteristic distance between two colors to be considered similar.

This representation is denoted by Kœnderink and van Doorn as *locally orderless image*. This expresses the fact that H does not contain information on the spatial organization of colors on a local level, since local contributions have been mixed by smoothing. Non local organization is still kept, as different positions $\mathbf{p}$ can have different local color distributions. The authors developed a method to produce out of an image several other images having the same local color distribution, but different pixelwise organization. Such images look scrambled (see examples in [4]), as the local order is modified. However, large homogeneous regions can still be discerned, as well as contrasted regions, even small ones.

The density in the position and color space used by Comaniciu and Meer is a special case of the local color distribution of Equation 8 when K is a separable kernel such that $K(\mathbf{x}) = K_r(\mathbf{p})K_\beta(\mathbf{c})$ and $\sigma \to 0$ (no smoothing of the original image).

When $r \to \infty$, the position and color distribution becomes a purely color distribution. Clustering pixels by associating to a mode then amounts to clustering the color distribution of the image, and retro-projecting the classes found onto the image. This approach was used in particular in the BlobWorld system by Carson et al. [2] who represent a color image as a set of color and texture blobs. They cluster color and texture features, without taking spatial information into account. Once clusters have been found, the spatial regions are found by retro-projection : each pixel is labeled like the cluster its color and texture belong to.

Three scale-space structures are associated with H in the Locally Orderless Images : one is related to the smoothing of the original image with parameter σ, the second is related to spatial integration with parameter r, the third is related to color distribution smoothing with parameter β. Therefore several possibilities exist to define a multi-scale segmentation method based on the position and color distribution, depending on which scales are to be used.

4.2. Multi-scale segmentation

In order to produce a hierarchical segmentation out of equation 8, one has to define a unique scale parameter. In this paper, we chose to keep σ fixed, and bind both β and r linearly to a unique scale parameter h:

$$\sigma = \sigma_0, \quad \beta = \beta_0 h, \quad r = r_0 h. \qquad (9)$$

Smoothing of the original image can then be considered a pre-processing step to remove some additive random noise that may be present in the image. The density function whose modes are to be computed is then:

$$H_h(\mathbf{p},\mathbf{c}) = \sum_i k\left(\frac{\|\mathbf{p}_i - \mathbf{p}\|^2/{r_0}^2 + \|c_i - \mathbf{c}\|^2/{\beta_0}^2}{h}\right). \qquad (10)$$

Equation 10 corresponds to the scale-space clustering of the point set $\{(\mathbf{p}_i/r_0, \mathbf{c}_i/\beta_0)\}_i$, that we detailed in section 2.3. The whole scale-space is a 6 dimensional space (2 dimensions for the position, 3 for the color, 1 for the scale parameter h).

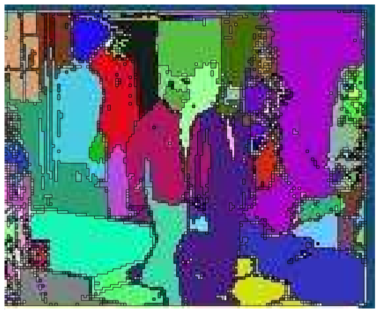
This algorithm leads to a tree structure, where leaves correspond to image pixels, and each node corresponds to a region, which correspond to one mode over its existence interval. When cutting this tree at a fixed scale we can produce a segmentation of the image.

In such a framework we do not have anymore the problem of choosing a unique scale (as we illustrated in figure 1), which could not handle simultaneously regions of different sizes. Indeed, each scale in considered in turn, and regions detected are organized in a unique multi-scale structure.

Figure 3 illustrate the properties of the fixed-scale approach and the scale-space approach. The segmentations shown have been computed with r_0 equal to the height of the image, and $\beta_0 = 100$ in L*u*v* color space. This ensures that both position and color have comparable dynamics. The scales-space algorithm was started at $h = 0.05$, and increased by a factor of $dh = 1.1$. We can observe that the modes of the fixed-scale method (b) lead to over-segmentation at $h = 0.1$,

(a)

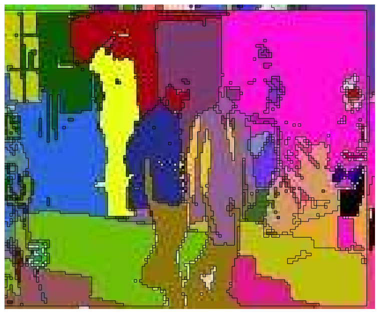
(b)



(c)

(d)

(e)



(f)

Figure 3: *Segmentations produced on the image (a), for* $r_0 = \text{image height}$ *and* $\beta_0 = 100$ *with L*u*v* color space. (b): Fixed scale segmentation without post-processing* $h = 0.1$. *(c): Fixed scale segmentation with post-processing* $h = 0.1$. *(d),(e) and (f): Segmentations produced by cutting the hierarchical segmentation at scales* $h = 0.1$, $h = 0.2$ *and* $h = 0.3$ *respectively.*

as we discussed, while the hierarchical approach produces less regions for the same scale (d).

5. Hierarchy simplification

The hierarchical segmentation produced previously contains information at several scales simultaneously. It is not viewable directly like a fixed scale segmentation. This limitation is not a problem when the segmentation is included in a system where several interpretations of the same image are useful, like for object tracking or image indexing, if both small and large visual elements are to be taken into account.

The great number of regions found at multiple scales can nevertheless be too large and need some special handling, specially in order to avoid the numerous small regions at low scales. First, the hierarchical structure helps reducing the number of regions tremendously, as modes detected for one scale get linked with modes detected for other scales. The redundancy across scales is therefore taken into account. Two regions are detected when they represent two different visual elements : either they are disjoint, or one corresponds to smaller details inside the other.

In order to further reduce the number of regions found, or if a non-hierarchical segmentation is needed, the clustering tree can be pruned, by removing some nodes which are not stable enough. The stability is expressed as the length of the existence interval of a node, that is the difference between the highest and the lowest scales for which a mode is associated to it. This difference is denoted by the lifetime of a node.

Witkin [8] proposed a *top-level description*, where each top-level node whose lifetime is smaller than the average lifetime of its immediate children is suppressed. Its children then become top-level nodes. The process is repeated until no top-level node can be suppressed. Wong [9] proposed a *longest lifetime first* algorithm, where a node is suppressed if any of its children has a longer lifetime.

We found those methods unadapted for our application. Indeed, the top-level description only considers removal of top-level nodes. If some nodes inside the hierarchy are very stable but are not the direct children of the top-level nodes, they may not be considered at all. On the opposite, the longest lifetime approach always select stable nodes, but at the same time select their siblings, which may be very unstable. Both effect can be seen in figure 4 (a) and (c).

Our method is related to the top-level description. Instead of considering only top-level nodes for suppression, we consider in turn each node, starting from the leaves and going up in the hierarchy. With this algorithm, very stable nodes can go up in the hierarchy by making their parents be suppressed. Unlike the longest lifetime first approach, a node is suppressed if its lifetime is less than the average of its children, therefore leading to a trade-off between making stable nodes go up and letting their unstable siblings do the same. Figure 4 (b) show the improvement that this technique has in our context.

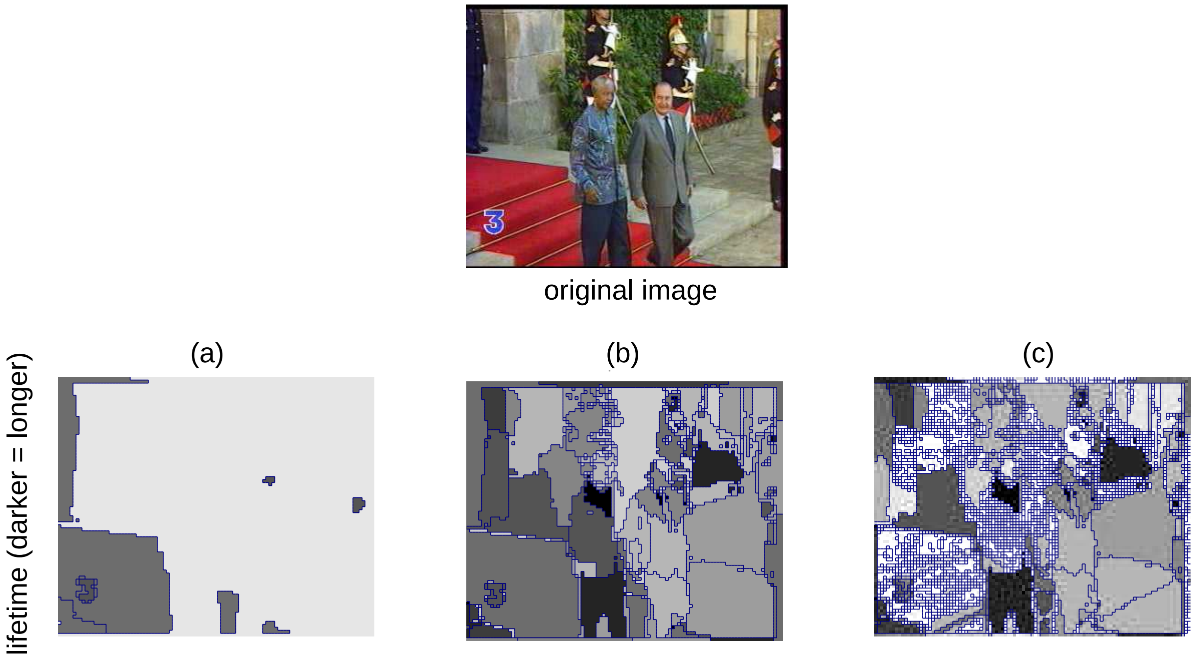


Figure 4: *Segmentations found by simplification of the multi-scale color image segmentation. The gray-level represents the lifetime of each selected region: darker means a more stable node, with a longer lifetime. (a): Top-level description. (b): Multi-level simplification. (c): Longest lifetime first.*

6. Conclusion

In this paper, we have presented a hierarchical color image segmentation. It is based on the hierarchical clustering of image pixels in a color and position space. We have shown how our method relate to the existing fixed-scale approach proposed by Comaniciu and Meer [3]. In particular, we showed that a post-processing step used in the original method what necessary in order to avoid over-segmentation, while our hierarchical approach produce stable regions, without requiring such a step.

We have also discussed the properties of the methods, and interpreted them within the framework of Locally Orderless Images [4]. From that formalism, we used two out of three scale parameters in order to build our scale-space color image segmentation method. Using a different set of parameters may lead to another vein of approaches.

References

[1] W. Bailer, P. Schallauer, H. B. Haraldsson, and H. Rehatschek. Optimized mean shift algorithm for color segmentation in image sequences. In *Conference on Image and Video Communications and Processing*, pages 593–600, San Jose, CA, USA, 2005.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Color- and texture-based image segmentation using EM and its application to image querying and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[3] D. Comaniciu and P. Meer. Mean-Shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[4] J. J. Koenderink and A. van Doorn. The structure of locally orderless images. *International Journal of Computer Vision*, 31(2/3):159–168, 1999.

[5] Y. Leung, J.-S. Zhang, and Xu Z.-B. Clustering by scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1396–1410, 2000.

[6] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Netherland, 1994.

[7] L. Lucchese and S.K. Mitra. Color image segmentation: A state-of-the-art survey. *Proc. of the Indian National Science Academy (INSA-A)*, 67, A(2):207–221, 2001.

[8] A. P. Witkin. *Image Understanding*, chapter Scale-Space Filtering: A New Approach to Multiscale Description, pages 79–95. Ablex, Norwood, NJ, 1984.

[9] Y.-F. Wong. Clustering by melting. *Neural Computation*, 5(1):89–104, 1993.

[10] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.