FAST COLOR IMAGE SEGMENTATION BASED ON LEVELLINGS IN FEATURE SPACE

Thierry Geraud, Giovanni Palma, Niels Van Vliet
EPITA Research and Development Laboratory
14-16 rue Voltaire, F-94276 Le Kremlin-Bicetre
France
thierry.geraud@lrde.epita.fr

Abstract This paper presents a morphological classifier with application to color image segmentation. The basic idea of a morphological classifier is to consider a color histogram as a 3-D gray-level image, so that morphological operators can be applied to it. The final objective is to extract clusters in color space, that is, identify regions in the 3-D image. In this paper, we particularly focus on a powerful class of morphology-based filters called levellings to transform the 3-D histogram-image to identify clusters. We also show that our method gives better results than other state-of-the-art methods.

Keywords: Classification; color image segmentation; color spaces; mathematical morphology; levellings.

1. Introduction

A classical approach to segmentation is to perform data classification in a judiciously chosen feature space. In the case of color images, trivial spaces are color ones such as red-green-blue (RGB) space or others more relevant with respect to human color perception. However, when dealing with natural images, image segmentation is a rather difficult task. In such images, objects are often textured, specular, and subject to color gradation and to noise. Consequently, color modes or classes usually do *not* have “simple” shapes in feature space, that is, they cannot be described easily by parametric models such as those described in [2]. In this context, mathematical morphology appears to be a suitable tool for studying data and extracting classes.

Every morphological classifier considers histograms as 3-D digital images in order to process them with common image operators. Segmentation of the color space is then used to classify the pixels in the original image. The main problem for natural images is to avoid over-segmented results.

In [9] a very simple morphological classifier based on *binary* mathematical morphology is proposed. The 3-D histogram is first thresholded to get a binary image in which only cluster cores appear. A morphological closing is then applied for regularization purpose and a connected component labelling process identifies the clusters. Unfortunately, this method does not take full advantage of the “level-shape” of histograms. In [14] an evolution of the former method is described. Before thresholding, the 3-D histogram is pre-processed by a morphological filter which digs the valleys, in order to increase the separability of clusters. A major problem of this method is that the initial relief between two clusters must be contrasted enough for them to be separated.

In [8] a difference of Gaussians from the histogram is computed and then thresholded. The resulting binary image of cluster cores is processed by a morphological closing and a connected component labeling is performed. Each component, i.e. each cluster, is then dilated to enlarge its volume in the feature space. At this stage, one cannot assign a label to every color: some colors of the original image do not belong to any cluster of the color space. Park et al. propose to assign such colors to their respective nearest clusters.

Last, a method is proposed in [1] which relies on morphological operators to model the clusters of training sets before to determinate class boundaries in feature space. However, this is not an automatic classifier.

This paper describes a very simple, efficient and effective clustering method based on a morphology study of data in color space. In section 2 we present a general scheme for histogram filtering and classification and we recall the definitions and some properties about openings, connected filters and levellings. In section 3 we explain how to modify the histogram in feature space to obtain relevant classification and we compare our method with others. Last we conclude in section 4.

2. About Histogram Filtering and Mathematical Morphology Operators

For the sake of clarity, this section deals with 1-D functions: $\mathbb{R} \rightarrow \mathbb{R}$. However, every notion given here is naturally expendable to greater dimensions. In the case of color images, these functions are $\mathbb{R}^3 \rightarrow \mathbb{R}$.

A General Scheme for Histogram Filtering and Classification

Basically, clustering in feature space aims at finding relevant peaks in this space. Consider the histogram of a gray image given in Figure 1. Without prior knowledge about the underlying intensity distributions of the object appearing in the original image, we can assume that proper locations for peak separations are close to minimum values for the function. These values are pointed out by the red bullets on the left diagram in Figure 1.

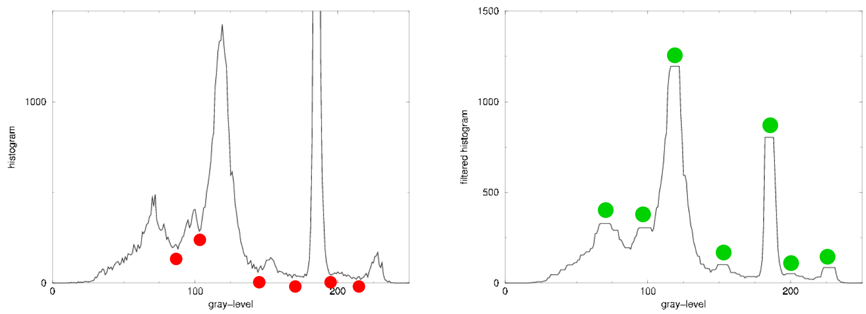

Figure 1. Histogram Filtering.

In a histogram most of its maxima are *not* representative of the presence of classes. Such maxima are just due to local variations of the function. They should be removed in order to keep significant peaks only, thus avoiding an over-classification of the feature space. A simple approach is to apply a filter that keeps (respectively removes), the proper (resp. invalid) maxima.

The result of such a filter is depicted on the right diagram in Figure 1. Every function maximum corresponds exactly to one relevant histogram peak. This is depicted by the green bullets on the right diagram in Figure 1. Furthermore, the expected locations of class separators (depicted by the bullets on the left diagram in Figure 1) are the only minima appearing in the filtered function.

Partitioning the feature space into classes is thus equivalent to put a frontier between every maxima. Such frontiers should be located on function minima. If we consider the negative of the filtered function (that is, $max_x(f(x)) - f(x)$), the classification problem can be re-phrased as follows: partitioning the feature space into classes is equivalent to separate every function minima *and* separations between minima should be located on function crest values. This operation is performed using a morphological filter, the geodesic watershed transform, as described in [13].

Finally, we end up with the following classification scheme, which is a much more simplified version of the one that has already been proposed in [6].

1 Compute image histogram,

2 if needed, regularize this histogram to get a better description of data in feature space,

3 apply a filter on this function to suppress inconsistent maxima,

4 invert the result,

5 run the watershed transform to get a partition of the feature space.

Obviously, the quality of such a classifier is highly dependent on the properties of the filter used in step 3. Thus, choosing and designing the appropriate filter is a critical step.

About Openings, Connected Filters and Levellings

Let us consider a function f defined on points and whose values $f(x)$ are quantified with n bits (this assumption allows us to simplify notations): $f : X \rightarrow [0..2^{n-1}]$, where X is a set of points. If $n = 1$, f is a Boolean function; if $n > 1$, we will say that f is a *scalar* function. The *flat zone* of f containing x, denoted by $\Gamma_x(f)$, is the largest connected component that includes x and such that $\forall x' \in \Gamma_x(f), f(x') = f(x)$. We have $\Gamma_x(f) \subset X$.

In the following, when a set of points $Z \subset X$ is given, we will denote $Z^{(i)}$ a connected component of Z, so that $Z = \cup_i Z^{(i)})$. Given a function f, $f_t(x)$ is defined as: $f_t(x) = 1$ if $f(x) \geq t$, 0 otherwise, and the set F_t as: $F_t = \{\, x \mid f_t(x) = 1 \,\}$

Morphological filters. A filter Φ is a *morphological filter* if it verifies two properties: it should be increasing ($f \leq g \;\Rightarrow\; \Phi(f) \leq \Phi(g)$) and idempotent ($\Phi \circ \Phi = \Phi$).

A morphological opening is an anti-extensive morphological filter ($\gamma_B \leq id$, where B is a structuring element) that can be expressed as the composition of an erosion and a dilation using a structuring element as defined in [11].

Because morphological opening is an anti-extensive filter, it can be used to suppress local maxima while "globally" keeping the information that was contained in the original image. However, these basic filters shift contours. This drawback makes their use redhibitory when object contours should be perfectly preserved ([5]).

Our objective is now to move to morphology-based filters that satisfy the contour preservation property. This family of filters is known as connected operators ([10]).

Connected operators. A filter ψ is a *connected operator* if the flat zones of the input function are included into the flats zones of the output function: $y \in \mathcal{N}(x)$ and $f(x) = f(y) \;\;\Rightarrow\;\; \psi(f)(x) = \psi(f)(y)$, where $\mathcal{N}(x)$ denotes the neighborhood of x. An equivalent definition comes with the decomposition of f into flat zones: ψ is a connected operator if $\forall x,\; \Gamma_x(f) \subset \Gamma_x(\psi(f))$.

A criterion κ defined over a set is increasing if: ($Z \subseteq Z'$ and Z satisfies κ) $\;\Rightarrow Z'$ satisfies κ. The *trivial opening* of a connected set Z is defined by: $\gamma_\kappa(Z) = Z$ if Z satisfies κ, $\emptyset$ otherwise. This definition is trivially extended to a non-connected set $Z = \{Z^{(i)}\}$ following $\gamma_\kappa(Z) = \cup_i \gamma_\kappa(Z^{(i)})$.

An *attribute opening* γ_κ of a function f relies on an increasing criterion κ: $\gamma_\kappa(f) = \sum_t \gamma_\kappa(f_t)$ where $\gamma_\kappa(F_t) = \cup_i \gamma_\kappa(F_t^{(i)})$.

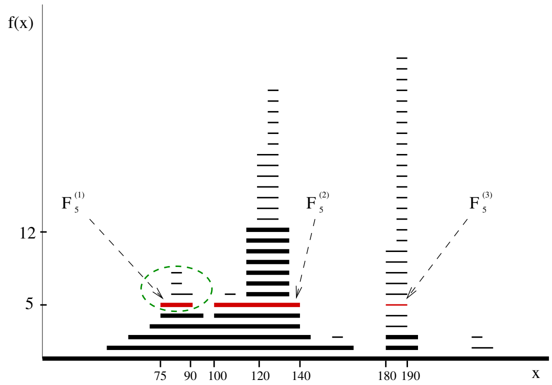

Figure 2. Area Opening and Volume Levelling.

A classical attribute opening is the area opening. The corresponding criterion is α_λ such as $\alpha_\lambda(Z)$ is verified iff $|Z| \geq \lambda$, where $|Z|$ denotes the number of points of Z and where λ is a given threshold. In Figure 2, a scalar function f has been decomposed into the sets $F_t^{(i)}$. The set F_5 is depicted in red. It has three connected components, and $F_5^{(1)} = [75, 90]$ (so $|F_5^{(1)}| = 16$). Filtering f by an area opening with $\lambda = 15$ is the function $\gamma_{\alpha_{15}}(f)$ that appears in Figure 2 when stacking bold lines. For instance, we have $\gamma_{\alpha_{15}}(f)(120) = 12$. Put differently, bold lines represent the sets $F_t^{(i)}$ which verify the criterion α_{15}. For instance, the criterion is not verified for $F_5^{(3)} = \{180, .., 190\}$ since $|F_5^{(3)}| = 11$ so we have $\gamma_{\alpha_{15}}(f)(185) < 5$. Please note that the flat zone of f_5 containing $x = 185$ is $\Gamma_{185}(f_5) = F_5^{(3)}$.

Levellings. Being connected is not such a strong property for an operator. Sometimes, we also want to preserve the local spatial ordering of function values. This leads to the definition of a sub-class of connected operators. A filter is a *levelling* if: $y \in \mathcal{N}(x)$ and $f(x) < f(y) \;\Rightarrow\; \psi(f)(x) \leq \psi(f)(y)$.

An interesting levelling is the volume levelling, ([12]). A volume can be computed from every $F_{t'}^{(i)}$ following:

$$\nu(F_t^{(i)}) = \sum_{t' \geq t,\, i' \text{ such as } F_{t'}^{(i')} \subseteq F_t^{(i)}} |F_{t'}^{(i')}|$$

For instance, the volume of $F_5^{(1)}$ is computed from the flat zones included in the ellipse drawn in Figure 2. We have $\nu(F_5^{(1)}) = |F_5^{(1)}| + |F_6^{(1)}| + |F_8^{(1)}|$. Last, the criterion used in filtering is based upon a volume threshold λ: $\nu_\lambda(Z)$ is verified iff $\nu(Z) \geq \lambda$. The filter is finally defined just like attribute openings: $\gamma_{\nu_\lambda}(f) = \sum_t \gamma_{\nu_\lambda}(f_t)$ where $\gamma_{\nu_\lambda}(F_t) = \cup_i \gamma_{\nu_\lambda}(F_t^{(i)})$. However, it is *not* an attribute opening since the criterion computation does not rely on $F_t^{(i)}$ only, but takes into account the components of $F_{t'}$ with $t' \neq t$.

3. Proposed Method and Comparative Results

Our method follows the classification scheme described in section 2. The key point is to use an appropriate filter to keep the relevant peaks in feature space only.

Histogram Filtering by Volume Levellings

Simply consider a histogram as a function f. The volume levelling filter has a simple interpretation: *it is a number of pixels in the original image*. The volume levelling process flattens a peak of the histogram only if the number of pixels (in the original image) which corresponds to the removed part of this peak is less than a given threshold. Another way to explain the meaning of this filter and the influence of the volume threshold is the following: no class will be created when the number of pixels from the original image would be less than λ.

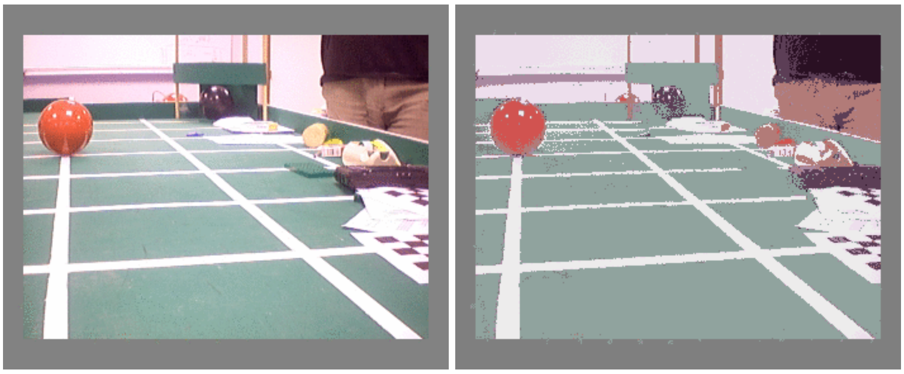

Figure 3. Classication using Volume Levelling Filtering: original image (left), our result (right, 6 classes).

The original robotic image contains 352×288 pixels encoded in 8 bits RGB; it is depicted in Figure 3 (left). First, the image histogram in the hue-saturation-lightness (HSL) space is computed (step 1). To speed-up the classification process, the histogram is down-sampled to 5 bits per color component and then regularized (step 2) with a Gaussian kernel ($sigma = 0.5$). For the filtering step (step 3), the volume threshold has been set to 0.05% of the number of pixels in the original image ($\lambda = 506$). Last, the filtered histogram is inverted (step 4) and the watershed transform (step 5) is applied to provide a color space partition into classes. This process leads to 6 color classes, and finally, the non-contextual labeling of the original image is depicted in Figure 3 (right). We can observe that the "green" class, corresponding to the table, is perfectly extracted in feature space, although the color of the table in the original image is not homogeneous. We have not yet performed a rigourous quantitative comparison of our results with other ones; nevertheless, extra results are accessible through the Internet from `www.lrde.epita.fr/dload/papers/iccvg04/` for a qualitative comparison of results over various images.

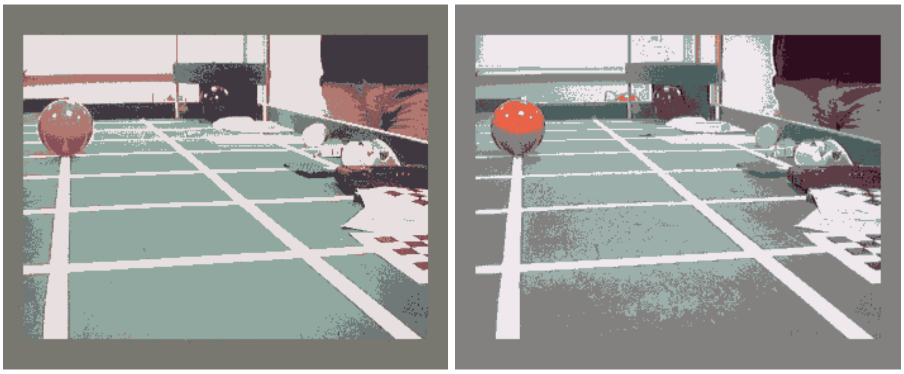

Figure 4. State-of-the-art Morphological Classifications: Zhang et al. (left., 8 classes) and Park et al. (right, 9 classes).

Figure 4 depicts the result of the morphological classifiers proposed in [14], and in [8] respectively. As we can see, classification in color space is less relevant than with our method (many artifacts appear in the resulting image due to bad class identification in feature space). Moreover, we have tried to tune the parameters of both of these classifiers but we did not succeed in getting a correct result with 6 classes.

Contextual Segmentation

The method presented here does not take into account contextual information to assign end labels to points. Thus, noise-like effects might appear in the labeled image. In such a case a contextual labeling using Markov random fields can be applied as presented in [6].

Implementation Details

To regularize with a Gaussian kernel, we use the fast recursive implementation explained in [4]. For the volume filter, we use an implementation based on the union-find algorithm from Tarjan. Our implementation is an adaptation of the one proposed for attribute openings in [7]. In all our experiments, we use `olena`, a generic image processing library written in C++ that we have developed ([3]). This library is available under the GNU Public Licence (GPL) through the Internet from `http://olena.lrde.epita.fr`

4. Conclusion

We have proposed a morphology-based classifier in feature space that takes advantage of levellings. First, it gives relevant results due to the strong properties of levellings. Second, both parameters of our method are very intuitive: a variance for the regularization, if needed, and the minimal number of pixels of a class. Last, the method is fast: a color image segmentation with our method takes less than 0.5s on a common computer—we have a 1,7 GHz personal computer running GNU/Linux.

References

[1] T. Barata and P. Pina. Improving classification rates by modelling the clusters of training sets in features space using mathematical morphology operators. In *Proceedings of the 16th Intl. Conf. on pattern Recognition*, volume 4, pages 90–93, Quebec City, Canada, 2002. IEEE Computer Society.

[2] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 750–755, San Juan, Puerto Rico, June 1997.

[3] J. Darbon, T. Geraud, and A. Duret-Lutz. Generic implementation of morphological image operators. In *Mathematical Morphology, Proceedings of the 6th Intl. Symposium (ISMM)*, pages 175–184. Sciro Publishing, April 2002.

[4] R. Deriche. Recursively implementing the gaussian and its derivatives. Technical Report 1893, INRIA, 1993.

[5] T. Geraud. Fast road network extraction in satellite images using mathematical morphology and markov random fields. In *Proceedings of the EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP)*, Trieste,Italy, June 2003.

[6] T. Geraud, P.Y. Strub, and J. Darbon. Color image segmentation based on automatic morphological clustering. In *Proceedings of the IEEE Intl. Conf. on Image Processing*, volume 3, pages 70–73, 2001.

[7] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):484–494, 2002.

[8] S.H. Park, I.D. Yun, and S.U. Lee. Color image segmentation based on 3-D clustering: Morphological approach. *Pattern Recognition*, 31(8):1061–1076, 1998.

[9] J.-G. Postaire, R.D. Zhang, and C. Lecocq-Botte. Cluster analysis by binary morphology. *IEEE Trans. on PAMI*, 15(2):170–180, 1993.

[10] J. Serra and P. Salembier. Connected operators and pyramids. In *Proceedings of SPIE Image Algebra and Mathematical Morphology IV*, volume 2030, pages 65–76, San Diego, CA, USA, 1993.

[11] P. Soille. *Morphological Image Analysis – Principles and Applications*. Springer-Verlag, 1999.

[12] C. Vachier. Morphological scale-space analysis and feature extraction. In *Proceedings of IEEE Intl. Conf. on Image Processing*, volume 3, pages 676–679, Thessaloniki, Greece, October 2001.

[13] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans. on PAMI*, 13(6):583–598, 1991.

[14] R.D. Zhang and J.-G. Postaire. Convexity dependent morphological transformations for mode detection in cluster analysis. *Pattern Recognition*, 27(1):135–148, 1994.