Sampled Mean Shift for Fast Image Segmentation

André Grilo, Alessio Del Bue, Alexandre Bernardino,
Instituto de Sistemas e Robótica, Instituto Superior Técnico
andremgrilo@gmail.com, adb@isr.ist.utl.pt, alex@isr.ist.utl.pt

Abstract

The mean shift algorithm has been used successfully in image segmentation. However this class of algorithms tends to consume considerable amount of computation. In this paper we present a sampled version of the mean-shift (M-S) algorithm. Instead of processing the whole image, our method iterates over a subset of the image points to initialize a line search process for region boundary delimitation. To finalize the segmentation, we merge regions with partial overlay. We illustrate the effectiveness of our method with both synthetic and real images.

1. Introduction

One of the most successful image segmentation to date is based on the mean-shift (M-S) algorithm [1] which is a general non-parametric technique for detecting the modes in the distribution of points in the joint space of color and pixel coordinates. However the classical approach considers all image points, which may render the algorithm too time consuming. With the proposed sampled mean-shift (SM-S) algorithm we develop a fast algorithm based on sparse approaches still open to significant research.

2. Sampled mean-shift segmentation

The proposed SM-S algorithm is divided in four stages. In the first stage we initialize (randomly) a set of candidate pixels which we call particles. Then, we apply standard M-S to converge to the closest distribution mode. Points are then grouped into clusters whose center (seed point) is used to start a line search process to identify region boundaries. The last step merges the detected regions with a consistent overlay. In the next sections we explain in detail each stage.

2.1 Random initialization and region seed points

We first initialize a set of pixels/particles uniformly sampled from the image. The number of particles is proportionally related to the image size. At each particle we run the standard M-S algorithm and let the particle converge to its corresponding color mode (called seed point). Dominant modes in the image will attract a given number of particles. We cluster together particles using a distance measure based on the image color and particle position after M-S convergence. The seed position is computed as the image centroid of such cluster. In this work we use the Manhattan distance to compute color dissimilarity:

$$\boldsymbol{c_1} = [r_1\ g_1\ b_1],\ \boldsymbol{c_2} = [r_2\ g_2\ b_2]$$
$$d = \sum |\boldsymbol{c_1} - \boldsymbol{c_2}|$$

In the case of cluster labeling, this test is done within a square area of 30 pixels from the test point.

2.2 Search lines

To calculate boundary points in an efficient way, a line search process is initiated from each seed in several directions. At each line, a boundary is detected if the color changes above a threshold with respect to the seed's color. To speed up the process, the line can be sub-sampled at a desired precision. At each sample point the color for testing is computed as the average of a neighborhood region, similar to [2]. The purpose of this last process is to increase robustness to image noise. The side effect is to blur the edges in the boundary contour.

A critical parameter in this algorithm is the angular separation between the lines generated in the search process. A too small value may result in an unnecessary level of resolution and slow down the algorithm. We have devised an adaptive method which considers a distance threshold between consecutive boundary points. If this distance is large, we add intermediate search lines, in a maximum of 20, to better cover the angular range. This is particularly important for non-convex regions.

At the end of this process, we obtain a convex polygon per seed composed by all ordered boundary points. Since each region may contain several seeds, a merge procedure must be applied to fuse regions with partial or full overlap.

2.3 Merge hulls

A region is defined by merging the polygons that have similar color. The procedure for merging must also take into account that polygons may overlap and regions can be non-convex. The merging method can be simplified by the following steps: (i) list the line segments of the original polygons; (ii) calculate intersection points between them, creating a new list with all the segments; (iii) test for inclusion of the mean point of each segment in all the other polygons, if the point is inside another polygon, that segment is not the most external and it is deleted; and (iv) reconstruct the final polygons following one external point until the path leads to the same point again.

3. Results

The results on artificial images define with good precision the different regions. The precision gets higher with more initial points. For 1000 initial points, Figure 1 a) shows the search lines beginning in the seed points. The color test is performed in a sparse way, avoiding long runtimes:

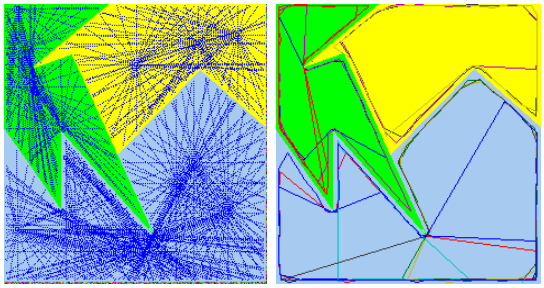

Figure 1 - a) Search lines in a variable number to guarantee good region coverage. b) Hulls from boundary points.

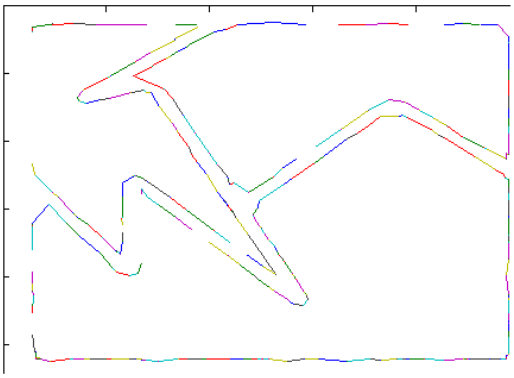

Figure 2 - Final region segmentation result.

Having the boundary points, it is trivial to construct the hulls for each cluster. These are the ones to be combined later in only one hull per region. Figure 1 b) shows the hulls before being connected and Figure 2 the connecting path after applying the merging algorithm.

With real images the results are less accurate due to the complexity and smoothness of the regions.

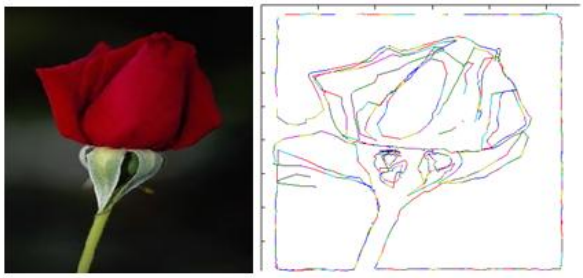

Figure 3 - Final Segmentation with real image.

The running time for a 128x128 image with 500 sample points in SM-S is of 10 seconds for the seed calculation plus 2 seconds for the line search and 1 second for merging the hulls. With the same conditions M-S runs for 7 minutes and 25 seconds. This represents a reduction of approximately 35 times in computation time.

4. Conclusion

We have presented a sampled version of the mean shift that allows a significant reduction on computation time with similar precision in simple cases. The algorithm begins with M-S iterations on a set of initial random points, in order to find the seeds for each region. From each seed a variable number of line searches are performed with more or less samples depending on the accuracy needed for the application. After a color mismatch between the seed and the line test point, a boundary point is calculated. Combining the final points of the search lines we obtain a convex polygon. To compute the final region's boundary we merge the polygons that overlay fully or partially if their color is similar.

Acknowledgements

Work partially funded by FCT (ISR/IST plurianual funding) through the PIDDAC Program funds.

References

[1] D. Comaniciu, P. Meer: Mean shift: A robust approach toward feature space analysis. IEEE Trans. PAMI, 24, 603-619, 2002.

[2] J. Sauvola and M. Pietaksinen. "Adaptive document image binarization", *Pattern Recognition*, 33, 2000