Reflectance-based Image Segmentation

Elena Garcés[1], Diego Gutierrez[1], Jorge Lopez-Moreno[1]

[1] *Affiliation: Grupo de Informática Grafica Avanzada*
Aragon Institute of Engineering Research (I3A).
University of Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, Fax +34-976762043, e-mail: egarces@unizar.es

Abstract

Most of the unsupervised image segmentation algorithms use just RGB color information in order to establish the similarity criteria between pixels in the image. This leads in many cases to a wrong interpretation of the scene since these criteria do not consider the physical interactions which give raise to of those RGB values (illumination, geometry, and albedo) nor our perception of the scene. In this paper, we propose a novel criterion for unsupervised image segmentation which not only relies on color features, but also takes into account an approximation of the materials reflectance. By using a perceptually uniform color space, we apply our criterion to one of the most relevant state of the art segmentation techniques, showing its suitability for segmenting images into small and coherent clusters of constant reflectance. Furthermore, due to the wide adoption of such algorithm, we provide for the first time in the literature an evaluation of this technique under several scenarios and different configurations of its parameters. Finally, in order to enhance both the accuracy of the segmentation and the inner coherence of the clusters, we apply a series of image processing filters to the input image (median, mean-shift, bilateral), analyzing their effects in the segmentation process. Our results can be transferred to any image segmentation algorithm.

Keywords: image segmentation, graph topology, image editing

1 Introduction

Over the years, the problem of image segmentation has been widely addressed under different perspectives and for different purposes. Additionally, the goal of the segmentation is an important factor to consider as in many cases we need a trade-off between speed and accuracy. Although different algorithms have been proposed, all of them share the same idea: internally, the resulting regions should contain similar pixels, while adjacent regions should be dissimilar with respect to a selected feature. Therefore, the choice of the similarity criteria is an important decision as it conditions the final result of the segmentation.

Color and texture are usually the selected criteria for the segmentations and, although good enough for many applications, there are others for which they fall short. A region with constant reflectance but with a shading variation may be mistakenly segmented in two or more regions if we use directly color information. Instead, a method which take into account the luminance variations due to shading, would obtain the correct segmentation in one region (see Figure1).

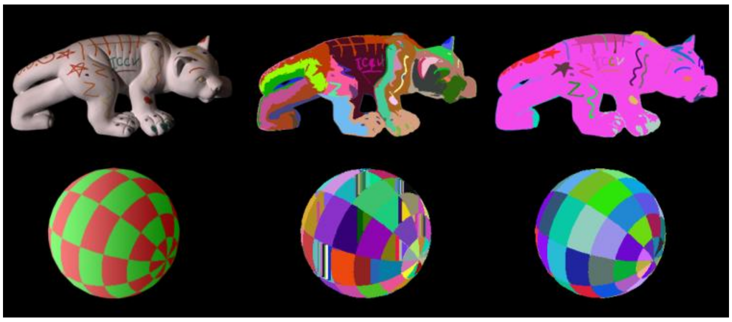


Figure 1: Segmentation example. (a) Original image. (b) Color-based segmentation. (c) Our reflectance-based segmentation

In this paper, we propose a novel criterion for image segmentation which avoids erroneous segmentations caused by the presence of shading and results regions of constant reflectance. Based on the use of a perceptually uniform color space (Shafarenko et al., 1998) we introduce our new criterion in the segmentation algorithm developed by (Felzenszwalb and Huttenlocher, 2004) which, in the last years, has been widely used for over-segmenting images (Hoiem et al., 2007; Micusik and Kosecka, 2010).

2 Related Work

The design of segmentation and clustering methods is highly dependant on the nature of both the input scenarios and expected behaviors, making almost impossible to cover the vast literature on this topic. Hence, in this section, we focus on the most relevant methods related to our approach: region-growing, graph-based and feature-based techniques. Inside this classification, we pay special attention to a subset of these methods which, over the last few years, are been widely used for over segmenting images into *superpixels*.

The idea of superpixels which are small and uniform sets of pixels, introduced by Ren et al. (Ren and Malik, 2003), allow a significant improvement of the computational efficiency of the algorithms, and also provide a low-level structure for algorithms which try to infer high-level information of the scene (Tao et al., 2001; Russell et al., 2006; Zitnick et al., 2007). There are three main algorithms commonly used for over-segmentation: N-Cut (Shi and Malik, 2000), Efficient Graph-Based (Felzenszwalb and Huttenlocher, 2004) and watershed algorithm (Vincent and Soille, 1991).

The first two algorithms are based on graph theory. The first one, Normalized Cuts (Shi and Malik, 2000), according to a cut criterion, makes minimum cuts in a graph which represents the image, in order to minimize the similarity between pixels that are being split. The second one, Efficient Graph-Based Segmentation algorithm (Felzenszwalb and Huttenlocher, 2004), is the faster and most widely adopted until date. It maps pixels in a feature space and uses a variable threshold for the segmentation (more details in Section 3}).

The last method widely used for over-segmentation is the watershed algorithm (Vincent and Soille, 1991). It places selectively a set of seeds in the image and by following the typical region-growing scheme, it obtains the different clusters.

Recent works of (Levinshtein et al., 2009) propose a fast method for obtaining quasi-uniform superpixels, which they call *turbopixels*, in regular graphs. Although its solution is the best providing over-segmentation in regular clusters, it is ten times slower than aforementioned Efficient Graph-Based Segmentation algorithm (Felzenszwalb and Huttenlocher, 2004). In a similar way, (Moore et al., 2008) devised an algorithm which builds regular lattices of superpixels.

One of the main existing techniques which search clusters within a feature space is the Mean-Shift (Comaniciu and Meer, 2002). This method smooths initially the image and groups similar pixels by its significant color for a posterior refinement and clusterization. Its performance is similar to the method by (Felzenszwalb and Huttenlocher, 2004), although as pointed out in (Unnikrishnan et al., 2007) is very sensitive to its parameters.

The use of perceptual color spaces was firstly studied by (Shafarenko et al., 1998) to obtain histogram-based segmentations. Later, (Chong et a., 2008) developed a new perceptual feature space for the segmentation.

3 Evaluation of a Color-Based Segmentation

Having evaluated the state-of-the-art in image segmentation methods, we decided to incorporate our new segmentation criteria to the Efficient Graph-Based segmentation method proposed by (Felzenszwalb and Huttenlocher, 2004). The main reasons for this choice are: first, as pointed out in (Levinshtein et al., 2009) it is the more efficient segmentation algorithm until date, both in terms of computational time and accuracy (which allows the interactive use of this method), and second, the flexibility of its design allow us to easily incorporate our segmentation criteria.

In the original paper, the authors introduce the algorithm an a few of its results. Although its performance and applicability are clearly exposed, they do not show empirically and with accuracy how the input parameters may affect the segmentation results. In particular, the selection of an initial *threshold*, which is a key part of the method since affects the final result of the segmentation, is ambiguously addressed. For this reason, we did the evaluation of the method showed in Section 4. Before the study, we describe briefly in the following section how the algorithm (Felzenszwalb and Huttenlocher, 2004) works.

3.1 Graph-Based Segmentation

The algorithm starts starts with an undirected graph G = (V, E) composed by a set of vertices v_i in V, corresponding to the pixels of the image to be segmented, and a set of edges (v_i,v_j) in E connecting pairs of neighboring pixels. Each edge has a weight $w((v_i,v_j))$ which represents the degree of similarity between the two connecting pixels. (Felzenszwalb and Huttenlocher, 2004) proposed two different graph structures: one based on a 8-neighborhood grid (*GRID* graph) using the eight nearest screen-space positions, and the other based in a *K*-Nearest Neighbor Graph (*KNN* graph), mapping each pixel in a N-dimensional space of features. Both the number *K* of connections per pixel and the N features can be freely defined.

In the case of a GRID graph, the function defining the similitude between two pixels connected by an edge is given by their differences in color. As suggested by the authors, we use the Euclidean distance L_2,

Equation 1

$$w((v_i, v_j)) = \left\| C(v_i) - C(v_j) \right\| = \sqrt{\sum_{t=1}^{N} (C_t(v_i) - C_t(v_j))^2}$$

where C(v) is the color vector of the vertex v, being C(v)={r,g,b} in *RGB* space.

For *KNN* graphs, each vertex is mapped in the space {x,y,C(x,y)}, where (x,y) is the location of the vertex in the image and C(x,y) is the color of the corresponding point, which depends on the color model employed. In the same way as with *GRID* graphs, the authors suggest to use the Euclidean distance L_2 to set the weights of the edges. However, in this case, the position of the pixels in the image is also taken into account for the weighting factor. The advantage of KNN over GRID is twofold: first, we can select a variable number of neighbors, and second, since the similitude function considers both the color and the spatial position per pixel, it allows connections between separated regions of the image with similar color values, in opposition to the locality of the GRID approach. However, the faster performance of GRID graphs makes them to be considered for the segmentations.

In the segmentation process, initially, each pixel corresponds to one cluster, then, in a posterior refinement the regions are merged according to a merging criterion. The algorithm finds the boundaries between regions by comparing two quantities: the first based in the difference between neighboring regions and the second based in the inner difference of each region plus a *variable threshold*, whose initial value is defined by the user and also depends on the size of the clusters. Intuitively, the difference between two regions is relevant if it is greater than the inner variation of, at least, one of the regions.

The variable threshold devised by the authors controls in certain manner the final size of the clusters and, hence, the final segmentation. As we show in the next section, the selection of this initial value is not simple and depends in great manner on the image.

4 Optimal parameters and topology

To study the algortihm by (Felzenszwalb and Huttenlocher, 2004), we start from the code published in their web page[1] so, it is necessary to comment an issue about that version. The implementation provided by the authors does not segment the image in each color channel separately, although the authors claim in the paper to work better for GRID graphs. Instead, it uses the Euclidean distance as pointed out in Equation 1. Nevertheless, our conclusions are not affected by this variation.

In order to evaluate the algorithm, we performed a series of experiments with *GRID* and *KNN* graphs (in the latter, varying the number of neighbors from five to fifty) over a set of synthetic and real images. Also, due to the lack of a concrete explanation of how the initial threshold affects the segmentation, and for the sake of automatization, we analyzed the output varying this value in a large range of values.

By observing the segmentation results for RGB version in Figure 8, we can see that GRID graphs are less sensitive to changes in the initial threshold, while if we modify this value in KNN graphs, we observe more influence in the coarseness of the segmentation. Also, the ability to capture non-local properties of the image with KNN graphs provides better segmentation results since the local neighborhood adapts to the geometry of the objects.

Attending to the initial threshold (*th*), our experiments show that unless we wanted an over-segmentation of the image at any case (th = 200), the selection of this value can not be automatic and depends in great manner on the image. While a good value for Figure 8 is 800 or 1000, in other figures could be 2000 or 4000 (see additional results in the attached files). Which is more, to select manually the optimal value for the threshold do not guarantee a correct segmentation. Notice how the regions obtained in Figure 8 for RGB version do not contain areas of constant reflectance. Instead, clusters are divided into small patches which do not follow the shape of the object and neither have reflectance meaning in the image. To avoid these problems and in order to obtain correct reflectance-based segmentations, we propose the method described in the following section.

5 Graph-Based Reflectance Segmentation

In this section we present our graph-based segmentation approach. First, we introduce our novel segmentation criterion which provides segmentation based on the approximated reflectance of the material. Second, we propose a pre-processing step with two known image filters (Mean Shift (Comaniciu and Meer, 2002) and Bilateral Filter (Tomasi and Manduchi, 1998) in order improve both the performance and the stability of the segmentation. Finally, we introduce an iterative refinement which increases the internal coherence of the resulting clusters.

[1] http://people.cs.uchicago.edu/~pff/segment/

5.1 The influence of color space

The original work by (Felzenszwalb and Huttenlocher, 2004) performs the image segmentation in RGB space as we have already shown in Section 4. Although their implementation produce compelling results if we need an over-segmentation of the image into small constant color patches, they are not suitable if we require regions representing the reflectance of the materials. In Figure 2 we can see an example of a situation in which a surface with constant albedo regions and shading produced by a horizontal light source is mistakenly segmented using the RGB color space. Notice how the erroneous clusters follow vertical areas of constant luminance.

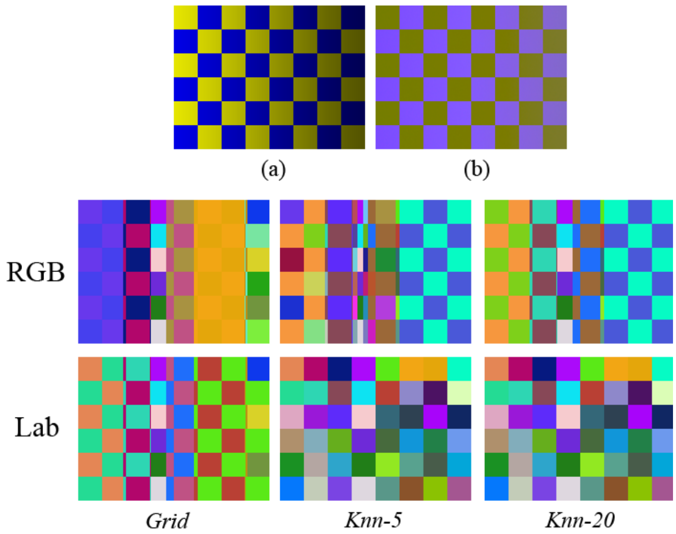


Figure 2. RGB Vs Lab comparison. (a) Input image (b) Chrominance. For any type of graph (Grid with 8-neighbors and KNN with 5 and 20 neighbors are shown), the best segmentations of (a) are obtained in Lab space.

Our method is designed to avoid the wrong interpretation of the scene caused by using RGB color space. Its goal is to go further and to look for clusters of approximately constant reflectance, rather than just obtaining constant color patches without significance. For this purpose, following previous approaches in the use of perceptually uniform color spaces (Shafarenko et al., 1998; Chong et al., 2008), we use *Lab* color space (CIE L*a*b*) over a modified version of the commented algorithm Efficient Graph-Based (Felzenszwalb and Huttenlocher, 2004). We rely on the studies of (Funt et al., 1992) which say that reflectance variations correspond to chromatical variations while luminance keeps constant to define our new color vector C(v) for Equation 1:

$$C(v) = \{0.5L\,, a, b\}$$

where C(v) is the color vector for vertex v and L, a, b are the values of such vertex in *Lab* color space.

This vector is a key part of the algorithm as it determines the similarity between pixels in the image. With our new definition, we associate changes in reflectance with changes in chromaticity. Experimentally, we have seen that to weight the luminance channel by 0.5 yields to plausible results for the segmentation because it helps to distinguish adjacent objects with similar chromaticity but different luminance.

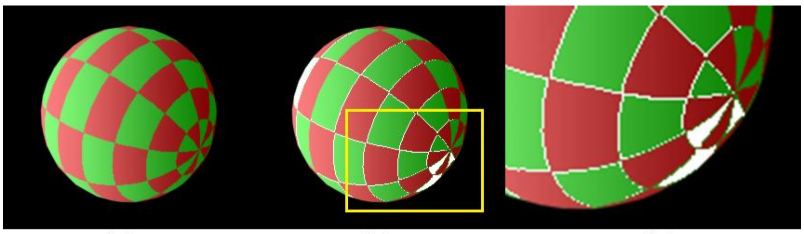
Following the assumptions of (Horn, 1986), who pointed out that at local level shading produces smooth variations of luminance while reflectance keeps constant, we benefit from the KNN graph implementation due to the fact that the feature space {x, y, 0.5 L, a, b} contains both the pixel position and the chromatic channels. Therefore, in the construction of the graph, the local neighborhood of each pixel adapts to the geometry of the object providing better segmentations. See in Figure 2 that the segmentation using *Lab* color space with KNN graphs is now correct.

5.2 Image processing filters and iterative processing

In order to improve the segmentation results, we propose a pre-processing step using one of these filters: Mean Shift filter (Comaniciu and Meer, 2002) or Bilateral Filter (Tomasi and Manduchi, 1998). These filters, by removing high-frequency texture and making the boundaries between regions sharper, improve the final segmentation. We can see some examples of applying these filters in Section 6.

To use Mean Shift filter before a segmentation algorithm was already proposed by (Unnikrishnan et al., 2007) which, in order to obtain more stable segmentations and less sensitive to parameter changes, applied such method before the Efficient Graph-Based segmentation algorithm (Felzenszwalb and Huttenlocher, 2004). This work (Unnikrishnan et al., 2007) suggested that the combination of these two methods (Comaniciu and Meer, 2002) and (Felzenszwalb and Huttenlocher, 2004) performs better than either two of them separately.

The results of the segmentation can be further refined (increasing the inner coherence of the clusters) by performing, after the first segmentation, an iterative process in which those clusters whose standard deviation exceeds the ranges of the image are re-segmented. Also, after each iteration, we execute a filtering process which consists in a median 2x2 filtering which reduces the color mix produced by the discretization in pixels of the region boundaries. This minimizes the misclassification of those mixed pixels. We can observe an example in Figure 3 of pixels wrongly segmented due to this effect.



(a) (b) (c)

Figure 3. RGB Vs Lab. (a) Input image (b) Chrominance. For any type of graph (Grid with 8-neighbors and KNN with 5 and 20 neighbors are shown), the best segmentations of (a) are obtained in Lab space.

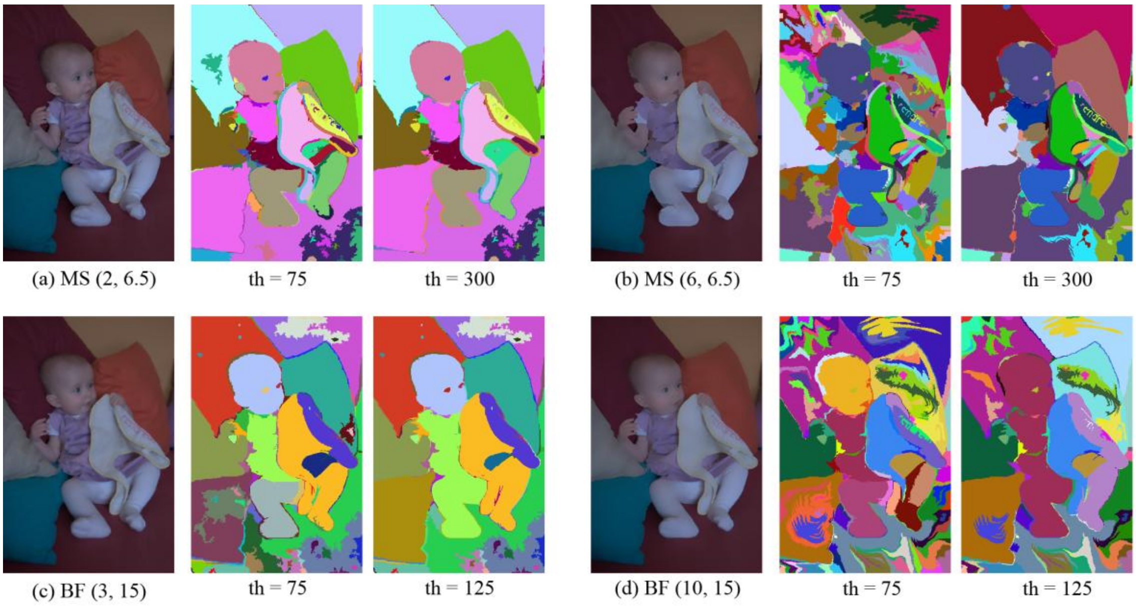


Figure 4. Pre-Processing step. Filtered image (a),(b),(c),(d) and two segmentation results varying threshold. First row with (MS) Mean Shift filter and (spacial bandwidth, color bandwidth). Second row with (BF) Bilateral Filter and (radius, luminance threshold).

6 Results

We have applied our method to a variety of input images. In some cases, and for the sake of clarity, we have masked out the main objects of the scene using a binary mask which defines the background in black.

In a similar fashion as with the RGB version (Felzenszwalb and Huttenlocher, 2004) (see Section 4), we performed a series of experiments in order to evaluate our algorithm with different graph implementations and different threshold values (Figure 8, Lab). From our experience, we can automatically set the optimal threshold for each image to the seventy percent of the maximum weight of the image edges. In Lab color space, unlike RGB, this value changes for each image due to the variability of the range of values that takes each color channel depending on the image. Nevertheless, our experiments show that independently of the image, we obtain compelling segmentations with a threshold between 50 and 100.

By paying attention to the type of graph, we can observe that there are not remarkable differences if we increase the number of neighbors for KNN graphs, finding with 5 neighbors a good solution (see Figure 5). Even GRID graph works acceptable with our implementation, although the fixed locality of the graph connections may incur in slight errors. We see in Figure 8-Lab a bad performance of GRID graph in the over-segmentation of the wall.

Figure 5: Segmentation examples using Knn-5 graph. The threshold value (th) is different for each image. Top right image copyright: original image from Captain Chaos, flickr.com

Our analysis of the pre-filtering step (see Figure 4) shows that by applying, before the segmentation, a soft Mean Shift filter, we obtain in most cases more accurate and defined clusters. Nevertheless, a coarse Mean Shift filter produces too quantized images which yield to non admissible segmentations. Attending to the segmentation after applying the Bilateral Filter, we find that, although this filter facilitates the gathering of similar regions, it also removes some contrasts inducing the disappearance of certain clusters. In both cases, the application of these filters yields to a more stable algorithm that is less sensitive to changes on the threshold value, due to the increment of the inner coherence of the clusters. Although the use of these filters is not necessary, in some cases, its application improves the segmentation results.

Comparing our results with the ones obtained by the algorithm developed by (Felzenszwalb and Huttenlocher, 2004), we observe the following: our implementation obtains coherent clusters which represent constant reflectance patches of the surface, while the RGB version obtains irregular clusters which neither follow a certain distribution nor respect the homogeneity of the surface, splitting flat constant color regions. Also, the use of *Lab* color space in our method, allow computing automatically the threshold value, unlike in the RGB version, where such value is strongly dependent on the image and cannot be pre-computed.

Our method is suitable for both color and gray scale images (see Figure 6), and performs properly for segmenting objects which do not contain high frequency textures. In such a case, to obtain a segmentation which captures each detail, we would need very small thresholds. To use too small thresholds in our algorithm, forces constant reflectance clusters to be split, thus produces erroneous segmentations (see Figure 7 for an example). For circumvent the problem, we could segment the image into different levels of detail just varying its threshold parameter for a posterior combination.

Figure 6: Segmentation example of gray scale image. (a) Input image. (b) Segmentation result. Notice how the clusters group objects of similar luminance.regions, hence, is incorrect.

Figure 7: Segmentation examples of high frequency texture. (a) Input image. (b) Knn 5 and th = 10, (c) Knn 5 and th = 75. Notice how the clusters of the sleeve in (b) do not follow constant reflectance regions, hence, is incorrect.

6 Conclusions

We have presented a novel criterion for segmenting images which, relying on the use of a perceptually uniform color space, obtains a segmentation based on the reflectance property of the materials. We have implemented this criterion into one of the most relevant segmentation methods until date (Felzenszwalb and Huttenlocher, 2004), which is characterized by both its efficiency and accuracy for over-segmenting images into clusters of uniform RGB color. Our approach benefits from its efficiency and achieves a segmentation which adapts to the geometry of the objects by ignoring luminance variations due to shading.

We have also provided an evaluation of the original algorithm by (Felzenszwalb and Huttenlocher, 2004) we have explored its input parameters and analyzed the output at different scenarios. Our experiments have shown that this algorithm is suitable for a fast over-segmentation into irregular clusters, but its application to high level segmentation is very unstable since the choice of the input parameters is not intuitive and cannot be automatically calculated.

Finally, in order to improve the segmentation results, we have contributed with the application of additional image processing filters (mean shift, bilateral filter, median), which may be used along with any segmentation algorithm. We have evaluated its performance with our segmentation algorithm, showing that its application yields to a more stable segmentations which are less sensitive to changes on its parameters. Moreover, we have devised that applying an iterative process over the segments of the image by re-segmenting those which not follow certain statistics; we obtain more accurate and coherent segmentations.

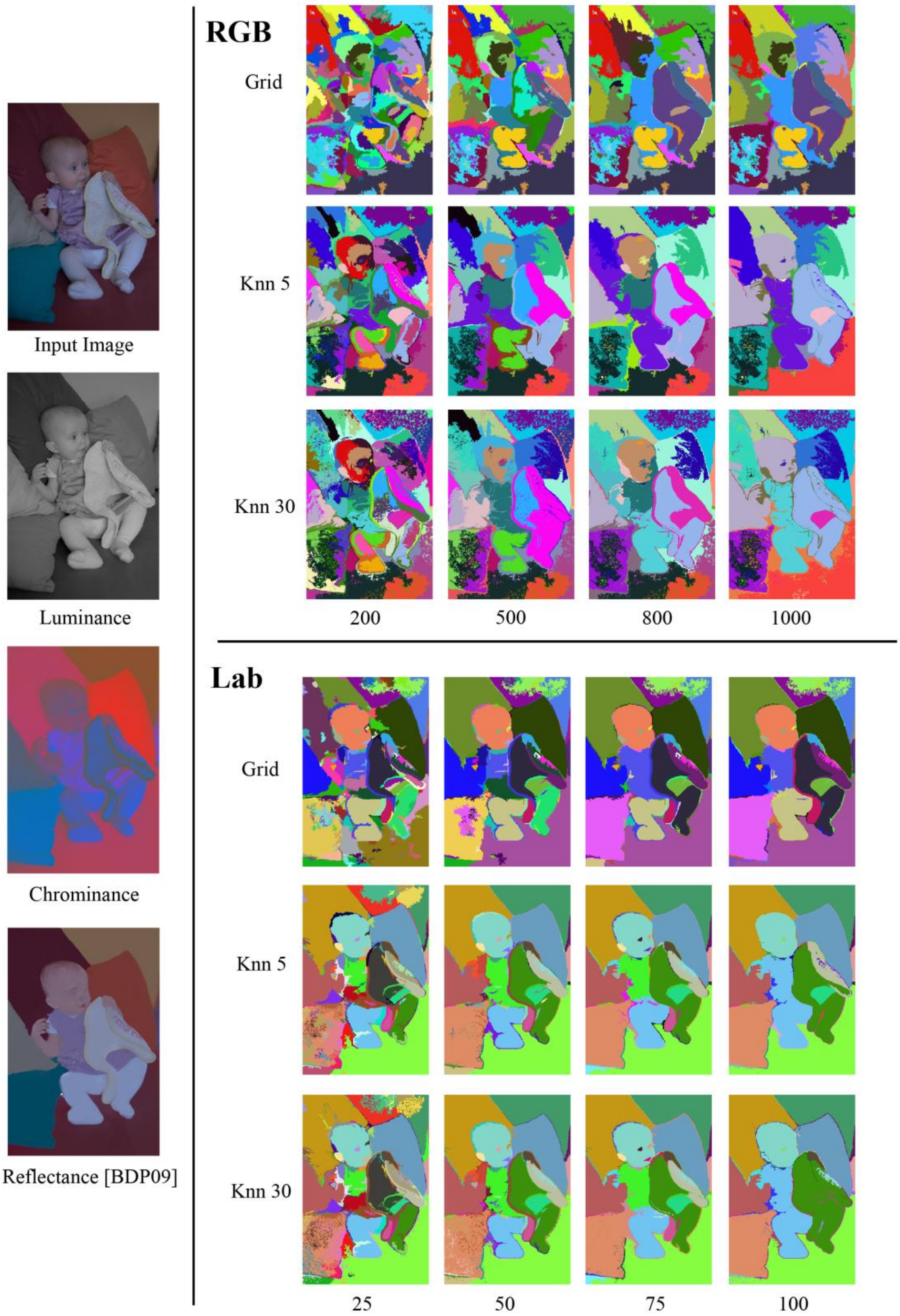


Figure 8: Parameters exploration. Segmentation results for the Input Image with RGB (top) and Lab color space (bottom). We explore Grid and Knn graphs with 5 and 30 neighbors. Also, we vary the threshold with the values showed. Notice how the correct segmentations (Lab, Knn graph and th ≥ 50) follow the reflectance image obtained by (Bousseau et al., 2009) in their intrinsic image decomposition. We observe how the best segmentations are obtained in Lab color space for all the cases.

7 Acknowledges

This research has been funded by a Marie Curie grant from the Seventh Framework Programme (grant agreement no.: 251415), the Spanish Ministry of Science and Technology (TIN2010-21543), the Gobierno de Aragon (projects OTRI 2009/0411 and CTPP05/09) and a generous gift from Adobe's Advanced Technology Lab at San Jose. Elena Garces was funded by a research grant from the I3A Fellowship Program.

References

Bousseau, A., Paris, S. and Durand, F. (2009). User Assisted Intrinsic Images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)*.

Chong, H. Y., Gortler, S. J. and Zickler, T. (2008). A perception-based color space for illumination-invariant image processing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 61:1-61:7.

Comaniciu, D. and Meer, P. (1997). Robust analysis of feature spaces: color image segmentation. *Computer Vision and Pattern Recognition.*

Comaniciu, D. and Meer, P. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 603-619.

Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient Graph-Based Image Segmentation. *Internationa Journal of Computer Vision*, 167-181.

Funt, B. V., Drew, M. S. and Brockington, M. (1992). Recovering Shading from Color Images. *European Conference on Computer Vision*, (pp. 124-132).

Hoiem, D., Efros, A. A. and Hebert, M. (2007). Recovering Surface Layout from an Image. *International Journal of Computer Vision*.

Horn, B. K. (1986). *Robot Vision.* MIT Press.

Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J. and Siddiqi, K. (2009). TurboPixels: Fast Superpixels Using Geometric Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2290 -2297.

Micusik, B. and Kosecka, J. (2010). Multi-view Superpixel Stereo in Urban Environments. *International Journal of Computer Vision*, 106-119.

Moore, A.P., Prince, S., Warrell, J., Mohammed, U. and Jones, G. (2008). Superpixel lattices. *Computer Vision and Pattern Recognition*, 1-8.

Ren, X. and Malik, J. (2003). Learning a Classification Model for Segmentation. *IEEE International Conference on Computer Vision.*

Russell, B. C., Freeman, W. T., Efros, A.A., Sivic, J. and Zisserman, A. (2006). Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1605-161).

Shafarenko, L. and Petrou, H. and Kittler, J. (1998). Histogram-based segmentation in a perceptually uniform color space. *IEEE Transactions on Image Processing*, 1354 -1358.

Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tao, H., Sawhney, H.S. and Kumar, R. (2001). A global matching framework for stereo computation. *IEEE International Conference on Computer Vision*, (pp. 532-539 vol.1).

Tomasi, C. and Manduchi, R. (1998). Bilateral Filtering for Gray and Color Images. *IEEE International Conference on Computer Vision.*

Unnikrishnan, R., Pantofaru, C. and Hebert, M. (2007). Toward Objective Evaluation of Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 929-944.

Vincent, L. and Soille, P. (1991). Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zitnick, C. L. and Kang, S. B. (2007). Stereo for Image-Based Rendering using Image Over-Segmentation. *International Journal of Computer Vision*, 49-65.