Volume xx (2011), Number z, pp. 1–7

Graph-Based Reflectance Segmentation

Elena Garces[1] and Diego Gutierrez[1] and Jorge Lopez-Moreno[1]

[1]University of Zaragoza, Spain

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, 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 reflectance properties of the materials. 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, shift-median, bilateral), analyzing their effects in the segmentation process. Our results can be transferred to any image segmentation algorithm.

1. Introduction

Over the years, the problem of image segmentation has been widely addressed under different perspectives and for different purposes. Also, 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 have similar pixels, while adjacent regions should be, among them, dissimilar with respect to a selected characteristic. 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 [CM97], 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, another method which take into account the luminance variations due to shading, would obtain the correct segmentation in one region (see Figure 1).

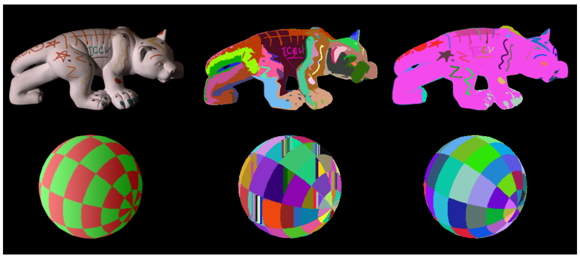


Figure 1: Segmentation example. (a) Original image. (b) Color-based segmentation. (c) Reflectance-based segmentation

In this paper, we propose a novel criterion for image segmentation which avoids erroneous segmentations caused by the presence of shading and yields regions of constant reflectance. Based on the use of a perceptually uniform color space [SPK98, FDB92], we introduce a new criterion in the segmentation algorithm developed by Felzenszwalb and Huttenlocher [FH04] which, in the last years, has been widely used for over-segmenting images [HEH05, SSN07, HEH07, MK10].

The high degree of configurability of this method and the lack of an analysis of the influence of its many parameters in the original paper, motivates our evaluation (Section 4), which we hope will serve as a base for future research in the field. We analyze the parameters of the algorithm showing the output at different scenarios, for different initial values and two graph implementations: grid and K-nearest neighbors graphs.

Additionally, we explore the use of two processing steps applicable to any segmentation algorithm. A pre-processing step using Mean Shift [CM02] and Bilateral Filter [TM98], and an iterative refinement of the the resulting clusters in order to increase its inner coherence.

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 clusters of pixels, introduced by Ren et al. [RM03], 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 [TSK01, RFE*06, ZK07]. There are three main algorithms commonly used for over-segmentation: N-Cut [SM00], Efficient Graph-Based [FH04] and watershed algorithm [VS91].

The first two algorithms are based on graph theory. The first one, Normalized Cuts [SM00], according to a cut criterion, makes minimum cuts in a graph which represent the image, in order to minimize the similarity between pixels that are being split. The second one, Efficient Graph-Based Segmentation algorithm [FH04], is the faster and most widely adopted. It maps the 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 [VS91]. It places selectively a set of seeds in the image which follow the typical region-growing scheme for obtaining the different clusters.

Recent work of Levinshtein et al. [LSK*09] propose a fast method for obtaining quasi-uniform superpixels, which they called *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 [FH04]. In a similar way, Moore et al. [MPW*08] devised an algorithm which build regular lattices of superpixels.

One of the main existing techniques which search clusters within a feature space is the Mean-Shift [CM02] algorithm. This method smooths initially the image and group similar pixels by its significant color for a posterior refinement and clusterization. Its performance is similar to the method by Felzenszwalb and Huttenlocher [FH04], although as pointed out in [UPH07] is very sensitive to its parameters.

The use of perceptual color spaces was firstly studied by Shafarenko et al. [SPK98] to obtain histogram-based segmentations. Later, Chong et al. [CGZ08] developed a new perceptual feature space for the segmentation. The approach of Mignotte [Mig08], combines information of several color spaces for perform its segmentation algorithm.

3. A Color-Based segmentation algorithm

Having evaluated the state-of-the-art in image segmentation methods, we have decided to incorporate our new segmentation criteria to the Efficient Graph-Based segmentation method proposed by Felzenszwalb and Huttenlocher [FH04]. There are two key reasons for this choice: First, as is pointed out in [LSK*09] is the more efficient segmentation algorithm until date, both in terms of computation time and accuracy (which allows the interactive use of this method), and second, the flexibility of their design for the addition of multidimensional criteria to its similarity function, which suits the purposes of this research.

In the original paper, the authors introduce the algorithm an a few of its results. Although they let its performance and its possibilities clear, 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 affecting the final result of the segmentation, is ambiguously addressed. For this reason, we evaluate of the method showed in Section 4. Before the study, we describe briefly in the following section how the algorithm [FH04] works.

3.1. Graph-Based Segmentation

The algorithm 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 [FH04] proposed two different graph structures: one based on a 8-neighbor grid (*GRID* graph) using the eight nearest screen-space positions, and the other based in the K nearest neighbors (*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,

$$w((v_i,v_j)) = \|C(v_i) - C(v_j)\| = \sqrt{\sum_{t=1}^{N} |C(v_i)_t - C(v_j)_t|} \quad (1)$$

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 a 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 considered for the weighting factor. The advantage of KNN over GRID is twofold: first, we can select a variable number of neighbors and second, the similitude function considers both the color and the spatial position per pixel, allowing the creation of 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 segmentation.

In the segmentation process, initially, each pixel correspond 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 some way the final size of the clusters and, hence, the final segmentation. As we show in the next section, the selection of its initial value it is not simple and depends in great manner on the image.

4. Optimal parameters and topology

For the study of the algortihm by Felzenszwalb and Huttenlochers [FH04], we start from the the code published in their web page † 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 [FH04]. Instead, it uses the Euclidean distance as pointed out in Equation 1. Nevertheless, our conclusions are not affected by the possible changes of the GRID graph segmentations results.

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 attachment 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 such 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 a segmentation based on the approximated reflectance of the material. Second, we propose a pre-processing step with two known image filters in order improve both the performance and the stability of the segmentation algorithms. Finally, we introduce a refinement iteration of the method which increases the internal coherence of the resulting clusters.

5.1. The influence of color space

The original work by Felzenszwalb and Huttenlocher [FH04] performs the image segmentation in RGB space as we have already shown in Section 4. Although their implementation produce compelling results

† http://people.cs.uchicago.edu/~pff/segment/

if we need an over-segmentation of the image into small constant color patches, they are not suitable if we require regions which represent 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.

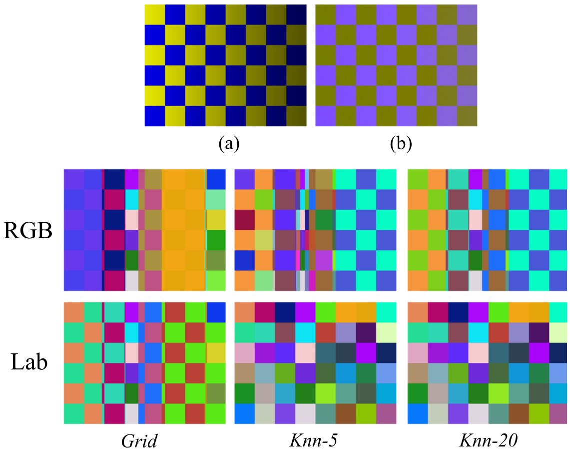


Figure 2: ***RGB Vs Lab comparison.*** *(a) Input image (b) Chrominance. For any type of graph (Grid with 8-neighbors and KNN with 5 and 20 neighbors are shown), the best segmentations of (a) are obtained in Lab space.*

Our method is designed to avoid a wrong interpretation of the scene caused by the use of 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 meaning. For this purpose, following previous approaches in the use of perceptually uniform color spaces [SPK98,CGZ08], we use *Lab* color space over a modified version of the commented algorithm Efficient Graph-Based [FH04]. We rely on the studies of Funt el al. [FDB92] which say that reflectance variations correspond to chromatical variations while luminance keeps constant, to define our new color vector $C(v)$ for Equation 1:

$$C(v) = \{0.5L, a, b\} \tag{2}$$

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 [Hor86], 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 $\{x, y, \{0.5L, a, b\}\}$ 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.

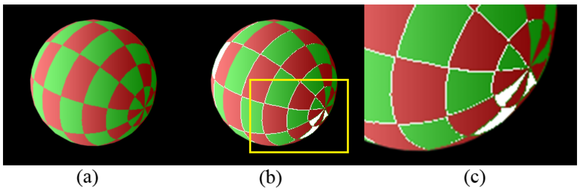


Figure 4: ***Segmentation examples.*** *White pixels represent an area classified as unique cluster. (a) Original input image. In (b),(c) we can observe how boundary pixels are wrongly selected as a large cluster of pixels due to the mix of colors between adjacent regions.*

5.2. Image Processing Filters and Iterative Processing

In order to improve the segmentation results, we propose a pre-processing step using Mean Shift filter [CM02] or Bilateral Filter [TM98]. These filters remove high-frequency texture and make the boundaries between clusters sharper, therefore, improving the final segmentation. We can see some results of applying this filters to an image in section 6

The use of Mean Shift filter before a segmentation algorithm was already proposed by Unnikrishnan et al. [UPH07], which, in order to obtain more stable segmentations which are less sensitive to parameter changes, applied such method before the Efficient Graph-Based segmentation algorithm [FH04]. The cited work by Unnikrishnan et al. [UPH07], suggested that this combination performs better than either two of the segmentation algorithms 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 with a standard deviation that exceeds the ranges of the image, get re-segmented. Also, after each iteration, we execute a filtering process which consisting 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 4 of pixels wrongly segmented due to this effect.

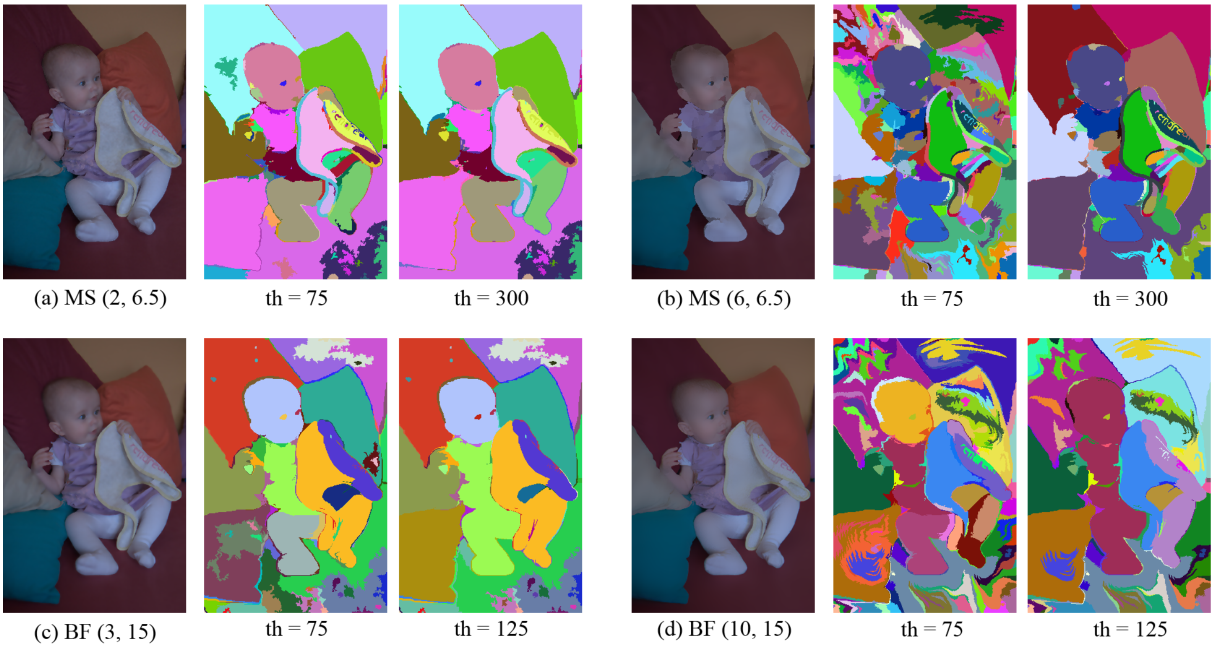


Figure 3: Pre-Processing step. First row (a) y (b), (MS) Mean Shift filter with (spacial bandwidth, color bandwidth). Second row (c) y (d), (BF) Bilateral Filter with (radius, luminance threshold)

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.

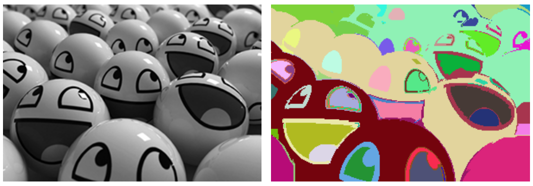
In a similar fashion as with the RGB version [FH04] (see Section 4), we performed a series of experiments in order to evaluate our algorithm with different graph implementations and different threshold values (Figure 8, Lab). From our experience, we can automatically set the optimal threshold for each image to the seventy percent of the maximum weight of all the edges of the image. 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 graphs show an acceptable performance with this implementation, although the spacial locality of the graph connections may incur in slight errors. We see in Figure 8-Lab an example of such problem with GRID graphs in the over-segmentation of the wall.

Our analysis of the pre-filtering step (see Figure 3) show that by applying a soft Mean Shift filter to the image before the segmentation, in most cases we obtain more accurate and defined clusters. Nevertheless, a coarse Mean Shift filter produces too quantized images, yielding 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 such 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 this filters is not necessary, in some cases its application improve the segmentation result.

If we compare our results with the ones obtained by the algorithm developed by Felzenszwalb and Huttenlocher [FH04], our implementation obtains coherent clusters which represent constant reflectance patches of the surface, while the RGB version [FH04] 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 to compute automatically the threshold value, unlike in the RGB version, where such value is strongly dependent on the image and cannot be precomputed.

Our method is suitable for both color and gray scale images (see Figure 6), and performs properly for segmenting objects which do not contains high frequency textures. In such cases, for obtaining a segmentation which captures



(a) (b)

Figure 6: ***Segmentation example of gray scale image.*** (a) Input image. (b) Segmentation Result. Notice how the clusters group objects of similar luminance.

(a) (b) (c)

Figure 7: ***Segmentation example of high frequency texture.*** (a) Input image. Output for Knn 5 and, (b) th = 10 (c) th = 75. Resulting segmentation with th = 10 yields to non correct segmentation. Notice how the clusterization of the sleeve not follow constant reflectance regions in such case (b).

each feature individually, we would need very small thresholds. The use of too small thresholds in our algorithm forces constant reflectance clusters to be split, thus producing 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.

7. Conclusions

We have presented a novel criterion for segmenting images, which rely on the use of a perceptually uniform color space to obtain 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, which it is characterized by both its efficiency and accuracy [FH04] for over-segmenting images into clusters of uniform RGB color. Our approach benefits from its efficiency performing a segmentation into clusters of constant reflectance which conforms 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 [FH04] exploring its input parameters and analyzing 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, we have provided additional cues to improve the segmentation results with the use of 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 coherence segmentations.

8. 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 Instituto de Investigación en Ingeniería de Aragón.

References

[BPD09] BOUSSEAU A., PARIS S., DURAND F.: User assisted intrinsic images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009) 28*, 5 (2009). 8

[CGZ08] CHONG H. Y., GORTLER S. J., ZICKLER T.: A perception-based color space for illumination-invariant image processing. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 61:1–61:7. 2, 4

[CM97] COMANICIU D., MEER P.: Robust analysis of feature spaces: color image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)* (Washington, DC, USA, 1997), CVPR '97, IEEE Computer Society, pp. 750–. 1

[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24* (May 2002), 603–619. 2, 4

[FDB92] FUNT B. V., DREW M. S., BROCKINGTON M.: Recovering shading from color images. In *Proceedings of the Second European Conference on Computer Vision* (London, UK, 1992), ECCV '92, Springer-Verlag, pp. 124–132. 1, 4

Knn 5, th = 50 *Knn 5, th = 150* *Knn 5, th = 75*

Knn 5, th = 50 *Knn 5, th = 50* *Knn 5, th = 75*

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

[FH04] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Efficient graph-based image segmentation. *Int. J. Comput. Vision 59* (September 2004), 167–181. 1, 2, 3, 4, 5, 6, 8

[HEH05] HOIEM D., EFROS A. A., HEBERT M.: Automatic photo pop-up. *ACM Trans. Graph. 24* (July 2005), 577–584. 2

[HEH07] HOIEM D., EFROS A. A., HEBERT M.: Recovering surface layout from an image. *International Journal of Computer Vision 75*, 1 (October 2007). 2

[Hor86] HORN B. K.: *Robot Vision*. MIT Press, 1986. 4

[LSK*09] LEVINSHTEIN A., STERE A., KUTULAKOS K., FLEET D., DICKINSON S., SIDDIQI K.: Turbopixels: Fast superpixels using geometric flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 31*, 12 (2009), 2290 –2297. 2

[Mig08] MIGNOTTE M.: Segmentation by fusion of histogram-based k -means clusters in different color spaces. *Image Processing, IEEE Transactions on 17*, 5 (May 2008), 780 –787. 2

[MK10] MIČUŠÍK B., KOŠECKÁ J.: Multi-view superpixel stereo in urban environments. *Int. J. Comput. Vision 89* (August 2010), 106–119. 2

[MPW*08] MOORE A., PRINCE S., WARRELL J., MOHAMMED U., JONES G.: Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), pp. 1 –8. 2

[RFE*06] RUSSELL B. C., FREEMAN W. T., EFROS A. A., SIVIC J., ZISSERMAN A.: Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (Washington, DC, USA, 2006), CVPR '06, IEEE Computer Society, pp. 1605–1614. 2

[RM03] REN X., MALIK J.: Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2* (Washington, DC, USA, 2003), ICCV '03, IEEE Computer Society, pp. 10–. 2

[SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22* (August 2000), 888–905. 2

[SPK98] SHAFARENKO L., PETROU H., KITTLER J.: Histogram-based segmentation in a perceptually uniform color space. *Image Processing, IEEE Transactions on 7*, 9 (Sept. 1998), 1354 –1358. 1, 2, 4

[SSN07] SAXENA A., SUN M., NG A.: 3-d reconstruction from sparse views using monocular vision. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (2007), pp. 1 –8. 2

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision* (Washington, DC, USA, 1998), ICCV '98, IEEE Computer Society, pp. 839–. 2, 4

[TSK01] TAO H., SAWHNEY H., KUMAR R.: A global matching framework for stereo computation. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (2001). 2

[UPH07] UNNIKRISHNAN R., PANTOFARU C., HEBERT M.: Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 1 (June 2007), 929–944. 2, 4

[VS91] VINCENT L., SOILLE P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell. 13* (June 1991), 583–598. 2

[ZK07] ZITNICK C. L., KANG S. B.: Stereo for image-based rendering using image over-segmentation. *Int. J. Comput. Vision 75* (October 2007), 49–65. 2

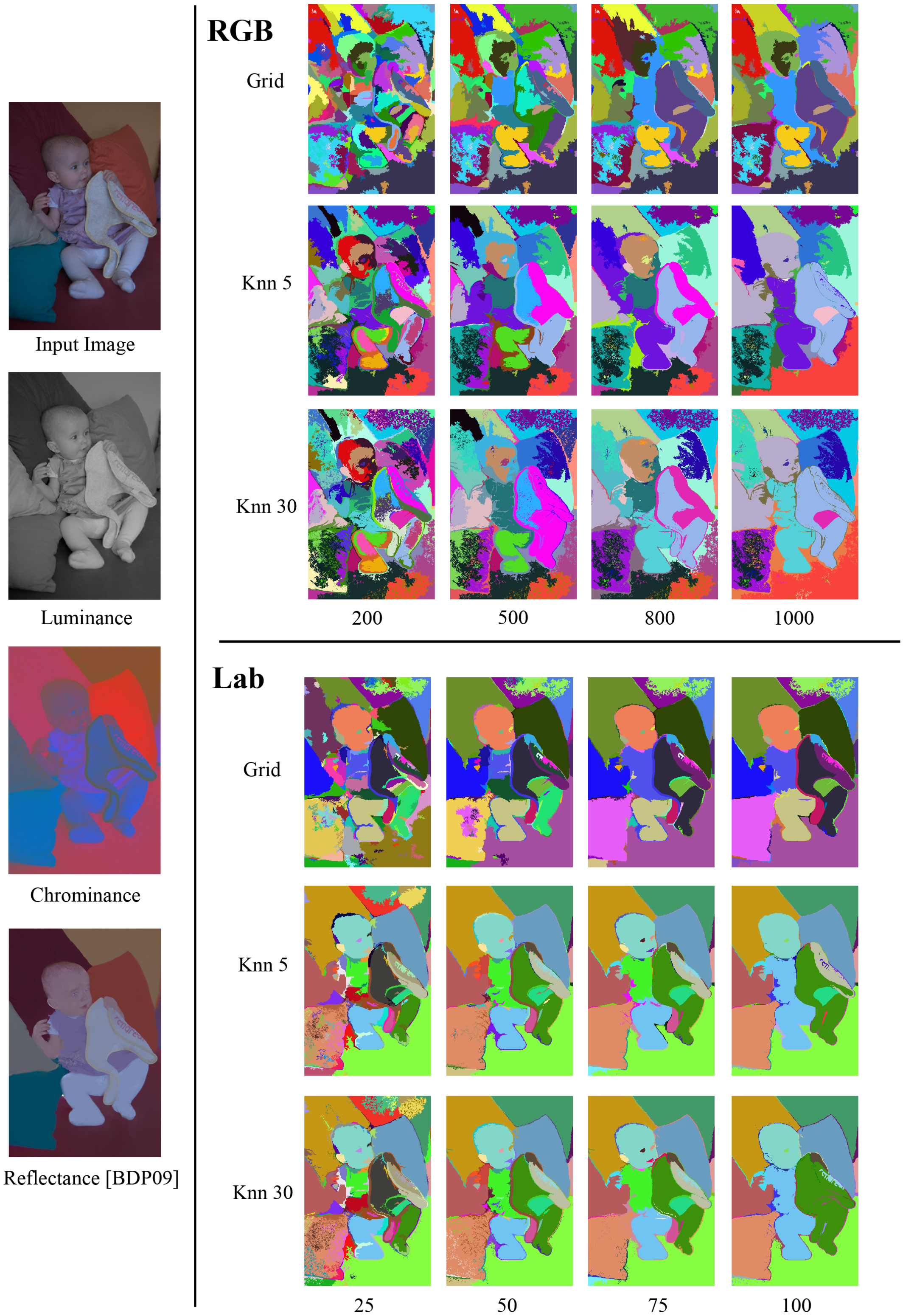


Figure 8: ***Parameters exploration.*** *We show the segmentation results for the Input image with RGB and Lab color space. We explore Grid and Knn graphs with 5 and 30 neighbors. Also we vary the threshold with the values showed in the image. For RGB version [FH04], we have chosen the range of the threshold values empirically. Notice how the corrects segmentation follow the reflectance image obtained by Bousseau et al. [BPD09] in their intrinsic image decomposition.*