Automatic Quick-Shift Segmentation for Color Images

Abdelhameed Ibrahim, Muhammed Salem, and Hesham Arafat Ali

Computers Engineering and Systems Department, Mansoura University, Egypt

Abstract

This paper proposes an automatic quick-shift segmentation method using illumination invariant representation of natural color images. In practice, the quick-shift segmentation method is sensitive to the choice of parameters, thus a quick tuning by hand is not sufficient. The proposed method segments images into homogeneous regions by applying the quick-shift method with initial values, then changes the quick-shift parameters values automatically to get the final segmented image. Changing parameters values make the proposed method flexible and robust against different image characteristics. We eliminate the factors that may affect natural image acquisition such as shadow and highlight by applying an invariant method. This method is valid for large size images. The effectiveness of the proposed method for a variety of images including different objects of metals and dielectrics is examined in experiments.

Keywords—*Quick-shift; color image segmentation; invariant representation*

1. Introduction

Image segmentation refers to the process of partitioning an image into homogeneous and connected regions or segments (sets of pixels, also known as superpixels). It is also typically used to locate objects and boundaries such as lines, curves, etc [1-2]. Superpixel [3-7] is commonly known as a perceptually uniform region in the image, thus removing shadow and highlight make image features more adequate. A superpixel representation greatly reduces the number of image primitives compared to the pixel representation. Moreover, superpixel segmentation provides the spatial support for computing region based features and change the representation of an image into another that is more meaningful and easier to analyze.

There are several image segmentation algorithms, some of them were designed for producing superpixels and may lack the ability to control the size, number, and compactness of the segments. Quick-shift [8] is a common method for image segmentation. The superpixels produced by quick-shift are not fixed in approximate size or number, unlike superpixelization schemes based on normalized cuts (e.g. [9]). A complex image with many fine scale image structures may have many more superpixels than a simple one, and there is no parameter, which puts a penalty on the boundary, leading to superpixels that are quite varied in size and shape. This produces segmentations consisting of many small regions that preserve most of the boundaries in the original image.

The superpixels extracted via quick-shift are controlled by three main parameters of ratio, KernelSize, and MaxDist. The ratio is a tradeoff between spatial consistency and color consistency. The KernelSize is the standard deviation of the parzen window density estimator, while MaxDist is the maximum distance between nodes in the quick-shift tree, which used to cut links in the tree to form the segmentation. Fulkerson et al. [10] use the same parameters for all of their experiments where these values were determined by segmenting a few training images by hand until they found a set that preserved nearly all of the object boundaries and had the largest possible average segment size. In practice, the algorithm is sensitive to the choice of parameters, thus a quick tuning by hand is not sufficient.

Color Image appearance and analysis is affected seriously by illumination factors such as shading, shadow, and specular highlight, observed from object surfaces in a natural scene. Therefore, image representation invariant to these factors proposed for color images in several ways [11-17]. These invariant representations play an important role in many applications such as segmentation, feature detection, edge and corner detection, object recognition, and image retrieval. Authors in [12] present an invariant method that is independent of the geometric parameters and is invariant to highlights and shading. This representation is available for all material surfaces including dielectric and metal, observed under a general illumination environment including colored light source.

This paper proposes an automatic quick-shift segmentation method using illumination invariant representation of natural color images. The quick-shift parameters values are changed automatically based on the invariant representation in real-time. The invariant method reduces shading, shadow, and specular highlight, which affect seriously the appearance and analysis of the natural color images but exceed the processing time, thus we divide image according to its size to reduce invariant time. Quantization process [18] is applied to get manual segmented image for similarity measure [19]. Quantization involves reducing the number of colors in an image by dividing the RGB color cube into a number of smaller boxes, and then mapping all colors that fall within each box to the color value at the center of that box. The performance of the proposed method for a variety of color images including different objects of metals and dielectrics is examined in experiments.

2. Superpixel Segmentation Methods

Algorithms for generating superpixels can be broadly categorized as either graph-based or gradient-ascent-based methods. Quick-shift is a gradient-ascent-based method. This will be explained in the following subsections.

2.1 Graph-based methods

NC05 – The Normalized cuts algorithm [9] recursively partitions a graph of all pixels in the image using contour and texture cues, globally minimizing a cost function defined on the edges at the partition boundaries. It produces very regular, visually pleasing superpixels. However, the boundary adherence of NC05 is relatively poor and it is the slowest among the methods (particularly for large images), although attempts to speed up the algorithm exist [20]. NC05 has a complexity of $O\left(N^{(3/2)}\right)$ [21], where N is the number of pixels.

SL08 – Moore et al. propose a method to generate superpixels that conform to a grid by finding optimal paths, or seams, that split the image into smaller vertical or horizontal regions [22]. Optimal paths are found using a graph cuts method similar to Seam Carving [23]. While the complexity of SL08 is $O\left(N^{(3/2)}\log N\right)$ according to the authors, this does not account for the pre-computed boundary maps, which strongly influence the quality and speed of the output.

2.2 Gradient-ascent-based methods

TP09 – Turbopixel method progressively dilates a set of seed locations using level-set based geometric flow [21]. The geometric flow relies on local image gradients, aiming to regularly distribute superpixels on the image plane. TP09 superpixels are constrained to have uniform size, compactness, and boundary adherence. TP09 relies on algorithms of varying complexity, but in practice, as the authors claim, has approximately $O(N)$ behavior. However, it is among the slowest algorithms examined and exhibits relatively poor boundary adherence.

SLIC10 – Simple linear iterative clustering (SLIC) is an adaptation of k-means for superpixel generation. Authors in [6] generate superpixels by clustering pixels based on their color similarity and proximity in the image plane. SLIC is similar to the approach used as a reprocessing step for depth estimation described in [7], which was not fully explored in the context of superpixel generation.

gSLIC11 – gSLIC [24] is a parallel implementation of the SLIC superpixel segmentation by using GPU and the NVIDIA CUDA framework. Using a single graphic card, this implementation achieves speedups of 10x to 20x from the sequential implementation. This allows using the superpixel segmentation method in real-time performance. The software is now online and is open source.

QS08 – Quick-shift [8] also uses a mode-seeking segmentation scheme. It initializes the segmentation using a medoid shift procedure. It then moves each point in the feature space to the nearest neighbor that increases the Parson Density estimate. It has relatively good boundary adherence and the superpixels produced by QS08 are not fixed in approximate size or number. Previous works have used QS08 for object localization [10] and motion segmentation [25] where the parameters were manually determined by segmenting a few training images. This algorithm segments a color image (or any image with more than one component) by identifying clusters of pixels in the joint spatial and color dimensions. Segments are local (superpixels) and can be used as a basis for further processing. Generating superpixels by quick shift are controlled by three parameters of Ratio-λ, Kernel Size–σ, and Distance–τ.

Ratio-λ: trade-off between spatial importance (X,Y,R,G,B) and color importance (R,G,B), it balances color-space proximity and image-space proximity and have a value from 0 to 1; Small ratio gives more importance to the spatial component and higher values give more weight to color-space.

Kernel Size–σ: Scale at which the density is estimated, it represents the standard deviation of the Parzen window density estimator (width of Gaussian kernel used in smoothing the sample density), it is common to refer to the Parzen window as "kernel" when the window is Gaussian, Higher means fewer clusters. For each pixel (x, y), quick shift regards $(x, y, I(x, y))$ as a sample from a $d + 2$ dimensional vector space. It then calculates the Parzen density estimate (with a Gaussian kernel of standard deviation σ)

$$E(x, y) = P\big(x, y, I(x, y)\big) = \sum_{x'y'} \frac{1}{(2\pi\sigma)^{d+2}} \exp\left(-\frac{1}{2\sigma^2}\begin{bmatrix} x - x' \\ y - y' \\ I(x, y) - I(x', y') \end{bmatrix}\right) \tag{1}$$

Then quick-shift constructs a tree connecting each image pixel to its nearest neighbor which has greater density value. Formally, write $(x',y') > P(x,y)$ if, and only if $P(x',y',I(x',y')) > P(x,y,I(x,y))$.

Distance–τ: the maximum distance- in the feature space - between nodes in the quick-shift tree this used to cut links in the tree to form the segmentation, Higher means fewer clusters. Each pixel $(x,y)$is connected to the closest higher density pixel parent (x,y) that achieves the minimum distance in

$$\mathrm{dist}(x,y) = \min_{(x',y') > P(x,y)} (x-x')^2 + (y-y')^2 + \|I(x,y) - I(x',y')\|_2{}^2 \quad (2)$$

The algorithm calculates a forest of pixels whose branches are labeled with a distance value. This specifies a hierarchical segmentation of the image, with segments corresponding to subtrees. Useful superpixels can be identified by cutting the branches whose distance label is above a given threshold τ (the threshold can be either fixed by hand, or determined by cross validation).

3. Automatic Quick-Shift Segmentation method

The proposed method block diagram is shown in Figure 1. The method aims to extract superpixels by automatically modifying the quick-shift parameters based on invariant images to be used for object detection and recognition. First, we eliminate the factors that may affect image acquisition such as shadow and highlight by applying an invariant method. For $C_s = [C_s{}^{(R)} \quad C_s{}^{(G)} \quad C_s{}^{(B)}]^T$ be a 3D column vector representing the red, green, and blue sensor responses of a color imaging system observing the object, the invariant equation is given as [12]:

$$C_s{}'^{(i)} = \frac{C_s{}^{(i)} - \min\{C_s{}^{(R)}, C_s{}^{(G)}, C_s{}^{(B)}\}}{\sqrt{\sum_{j\in\{R,G,B\}} \left(C_s{}^{(i)} - \min\{C_s{}^{(R)}, C_s{}^{(G)}, C_s{}^{(B)}\}\right)^2}},$$

$$\text{For } i\in\{R,G,B\} \quad (3)$$

This representation is available for all material surfaces including dielectric and metal, observed under a general illumination environment including colored light source. However, the invariant takes more time if image size exceeds 250x250, so we have to divide image according to its size to reduce invariant time.

To reduce the number of colors of the invariant image for manual segmentation, we use a quantization process [18]. Manually segmented image is used in similarity measure [19] as a reference image. Quantization involves reducing the number of colors in an image by dividing the RGB color cube into a number of smaller boxes, and then mapping all colors that fall within each box to the color value at the center of that box. Figure 2 shows an example of the importance of the quantization.

We apply the quick-shift [8] method to extract superpixels from the invariant image to be compared with the manual segmented image. In practice, the algorithm is sensitive to the choice of parameters, so a quick tuning by hand is not sufficient. Figure 3 shows tuning quick-shift parameters manually versus automatic parameters tuning. Selecting wrong value for quick-shift may cause losing some image information or splitting it up into too few segments as shown in Fig. 3(b). It may also exceed the number of segments unnecessarily or splitting it up into too many regions as shown in Fig. 3(c). Thus, selecting quick-shift parameters values make segmented image more meaningful and easier to analyze as presented in Fig. 3(d).

The superpixels are controlled in this method by three parameters of ratio, KernelSize, and MaxDist. The initial parameters values are set, and then the similarity between segmented image and quantized invariant image is calculated. The accuracy of the segmentation results between the segmented image and the quantized invariant images is numerically demonstrated by the similarity measure with a suitable window size for labeled images [19]. The algorithm is repeated several times automatically with changing the parameters values until reaching the highest similarity (Max S). The overall proposed method steps for producing the final segmented image are as follow:

```
1  Input IMAGE                              // Natural color image
2  Check IMAGE size,apply invariant to IMAGE
3  Quantize invariant-IMAGE                 // Produce reference image
4  Initialize: Ratio,KernelSize, MaxDist    // Parameters Initialization
   Determine Ratio range: Min_Ratio to Max_Ratio
   Determine kernel size range: Min_KernelSize to Max_KernelSize
   Determine MaxDist range: Min_MaxDist to Max_MaxDist
5  Repeat
     Quick-shift Segmentation for invariant-IMAGE
     Calculate similarity (S)       // Set first S as reference S
       If New-S > Old-S
       TempRatio= Ratio             //Get values of best similarity
       TempKernelSize= KernelSize
       TempmaxDist= MaxDist
       Update: Ratio,KernelSize, MaxDist
     Until Ratio=Max_Ratio
         KernelSize =Max_KernelSize
         MaxDist =Max_MaxDist
6 Quick-shift Segmentation for invariant-IMAGE using  //Final segments
      Ratio= TempRatio,
      KernelSize = TempKernelSize
      MaxDist =TempmaxDist.
```

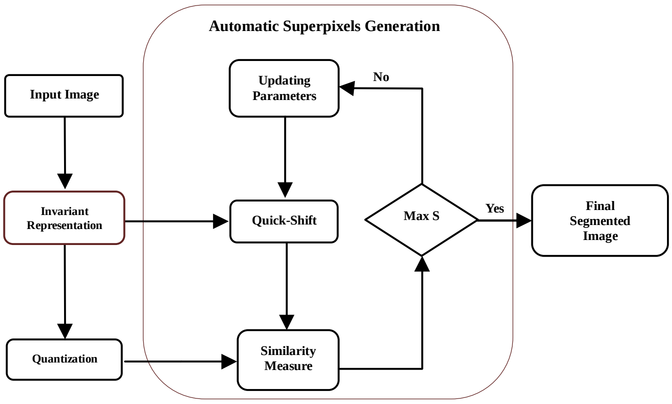
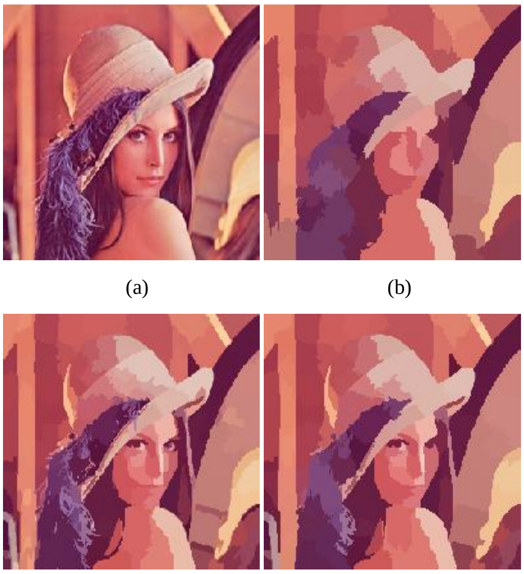


Fig. 1: Automatic quick-shift segmentation method block diagram

(a) Original Image (b) Quantized Image

Fig. 2: Image Quantization [18].



(a) (b)

(c) (d)

Fig. 3: Tuning quick-shift parameters manually versus automatic parameters tuning. (a) Original image, (b) Manual tuning with few segments, (c) Manual tuning with many segments, (d) Automatic tuning.

4. Experiments

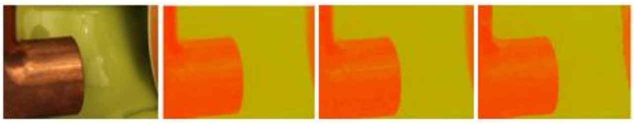
We use a digital still camera for objects observation. The imaging system consists of a Canon EOS-550D digital camera, incandescent lamps, and a personal computer. The normalized color values were calculated for RGB channels by eliminating illumination effect using the GretagMacbeth color checker. The camera system moves freely to capture color images for various objects. The experiments are run on CPU Intel Core2 Due 2GHz with 2G memory using Matlab.

We have examined the performance of the proposed method for different color images of natural scenes. Figure 4 shows a color image that including different objects. The image with size of (409x312). Figure 4(b) presents the invariant representation of the image to eliminate the factors that may affect image acquisition such as shadow and highlight. The quantized manual segmented image as a reference image for similarity measure is shown in Fig. 4(c). Figure 4(d) shows the final segmented image. We note that the final segmented image by the proposed method is better than the invariant image. The edges of the proposed segmented image are more sharpen; however, the invariant images are look noisy.

For the results in Fig. 4, we test different values of the quick-shift parameters of ratio, KernelSize, and MaxDist. Table 1 shows that the best similarity of 91.25% for this image is produced in CPU time of 21.47s and the best parameters values are Ratio=0.2, KernelSize=4, and MaxDist=30. In our experiments, we note that the best values for most of the images are KernelSize=2 and MaxDist =10, however the ratio is different from image to another according to the nature of the image and the objects within it.

To show the effectiveness of the proposed method, we have used a test scene including a metal object of copper and a dielectric object of ceramic (chicken).Figure 5 shows analysis of a part of a color image. Figure 5(a) shows the original color image. Figure 5(e) introduces the 3D view of the part of the original image, Fig. 5(f) shows the 3D view of the invariant image, and Fig. 5(g) shows the 3D view of the proposed method. Note that the shadow and highlights are disappeared from both samples, even though the segmentation of the proposed method has much less noise. It is clear and has sharp edges than the invariant representation [12].

The final experiment has been tested the proposed method stages compared to k-mean algorithm [26] as shown in Figure 6. Figure 6(a) shows three different color images. The manual segmented images are presented in Figure 6(b). Figure 6(c) shows the invariant representations of the tested images. Figure 6(d) illustrates the k-means segmentation based on the original image. The proposed method segmentations based on the invariant images are shown in Fig. 6(e). Note that the proposed method is better as its segmented images are clearer than the k-means results. The results show the performance and stability of the proposed method.



(a) (b) (c) (d)

Fig.4: Illustrates different stages of the proposed algorithm. (a) Original image, (b) Invariant image, (c) Manual segmented image by quantization, (d) Proposed quick-shift segmented image

Table1. Results for the image in Figure 4.

Ratio	KernelSize	MaxDist	Similarity	CPU-Time (s)
0.2	2.0	10.0	0.8512	12.00
0.2	2.0	20.0	0.8865	13.66
0.2	2.0	30.0	0.9053	18.05
0.2	4.0	10.0	0.8649	15.37
0.2	4.0	20.0	0.8955	17.55
0.2	**4.0**	**30.0**	**0.9125**	**21.47**
0.9	2.0	10.0	0.8373	11.29
0.9	2.0	20.0	0.8811	14.27
0.9	2.0	30.0	0.9018	17.87
0.9	4.0	10.0	0.8554	15.18
0.9	4.0	20.0	0.8920	17.77
0.9	4.0	30.0	0.9096	21.64

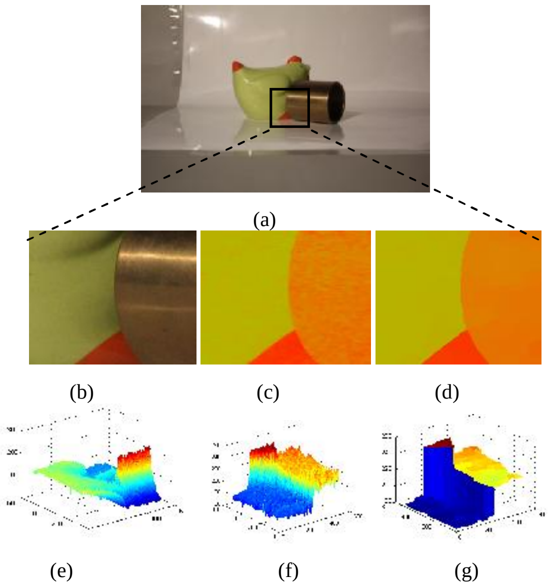


Fig. 5: Proposed automatic Quick-shift segmentation evaluation for natural color image including a metal object of copper and a dielectric object of ceramic. (a) Original image, (b) Part of Original image, (c) Invariant representation, (d) Proposed method, (e) 3D surface view for (b), (f) 3D surface view for (c), (g) 3D surface view for (d).

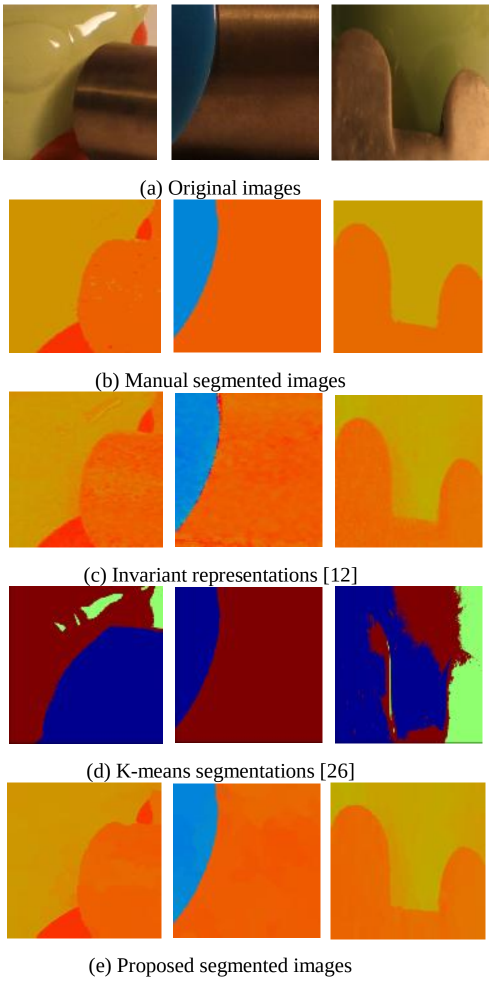


Fig. 6. Illustrates different images and stages of the proposed method

5. Conclusions

In this paper, we proposed automatic quick-shift segmentation method using illumination invariant representation of natural color images. The quick-shift parameters values are updated automatically based on the invariant representation in real-time. The invariant method reduces shading, shadow, and specular highlight, which affect seriously the appearance and analysis of the natural color images but exceed the processing time, thus we divide image according to its size to reduce invariant time. Quantization process is applied to get manual segmented image for similarity measure. We examined the performance of the proposed method for a variety of color images including different objects of metals and dielectrics in experiments. The segmentation of the proposed method has much less noise. It is clear and has sharp edges compared to the invariant representation. The proposed method is better as its segmented images are clearer than the k-means results. The results show the performance and stability of the proposed method.

References

[1] Krishna Kant Singh and Akansha Singh. A Study Of Image Segmentation Algorithms For Different Types Of Images," International Journal of Computer Science Issues, Vol. 7, Issue 5, 2010.

[2] M.Seshashayee, K. Srinivas Rao, Ch. Satyanarayana, and P. Srinivasa Rao, "Image Segmentation Based on a Finite Generalized New Symmetric Mixture Model with K–Means,"International Journal of Computer Science Issues, V ol. 8, Issue 3, No. 2, 2011

[3] X. Ren and J. Malik, "Learning a classification model for segmentation," In Proc. IEEE International Conference on Computer Vision (ICCV), 2003.

[4] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2012.

[5] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2009.

[6] R.Achanta, A.Shaji, K.Smith, A.Lucchi, P.Fua, and S.Susstrunk, "SLIC Superpixels," EPFL Technical Report 149300, 2010.

[7] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," International Journal of Computer Vision (IJCV), 75:49–65, 2007.

[8] A. Vedaldi, and S. Soatto, "Quick shift and kernel methods for mode seeking", In Proc. ECCV, 2008.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), pages 888-905, 2000.

[10] B. Fulkerson, A. Vedaldi, and S. Soatto. "Class segmentation and object localization with superpixel neighborhoods", IEEE 12th International Conference on Computer Vision (ICCV), Pages 670 - 677, 2009.

[11] A. Ibrahim, S. Tominaga, and T. Horiuchi, "Invariant representation for spectral reflectance images and its application", EURASIP Journal on Image and Video Processing, Pages 1-12, 2011.

[12] A. Ibrahim, S. Tominaga, and T. Horiuchi, "Illumination-invariant representation for natural color images and its application", IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI2012), 2012.

[13] J.-M. Geusebroek, R. Boomgard, A. W. M. Smeulders, and H. Geerts, “Color invariance,” IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol.23, no.12, pp.1338–1350, 2001.

[14] T. Gevers and A. W. M. Smeulders, “Color based object recognition,” Pattern Recognition, vol.32, no.3, pp.453–464, 1999.

[15] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol.32, no.9, pp. 1582-1596, 2010.

[16] R. T. Tan and K. Ikeuchi, “Separating reflection components of textured surface using a single image,“ IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), vol.27, no.2, pp.178-193, 2005.

[17] S. P. Mallick, T. E. Zickler, D. J. Kriegman, and P. N. Belhumeur, “Beyond Lambert: reconstructing specular surfaces using color,“ IEEE Proc. Computer Vision and Pattern Recognition (CVPR'05), vol.2, pp.619-626, 2005.

[18] P. S. Periasamy, S. Athi Narayanan, and K. Duraiswamy, "A Hybrid Method for Image Quantization", International Journal of Mathematics & Computation, Vol. 5, No. D09, 2009.

[19] T. Horiuchi, "Similarity measure of labelled images," Proc. IAPR 17th International Conference on Pattern Recognition (ICPR'04), vol. 3, pp.602-605, 2004.

[20] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition", In IEEE Computer Vision and Pattern Recognition (CVPR), 2005.

[21] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2009.

[22] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel Lattices," IEEE Computer Vision and Pattern Recognition (CVPR), 2008.

[23] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," ACM Transactions on Graphics (SIGGRAPH), 26(3), 2007.

[24] C. Yuheng Ren and I. Reid, "gSLIC: a real-time implementation of SLIC superpixel segmentation", University of Oxford, Department of Engineering Science, 2011

[25] A. Ayvaci and S. Soatto, "Motion segmentation with occlusions on the superpixel graph," In Workshop on Dynamical Vision, 2009.

[26] Duda RO, Hart PE, Stork DG: Pattern Classification. 2nd edition. John Wiley & Sons, New York; 2001.

Abdelhameed Ibrahim was born in Mansoura city, Egypt, in 1979. He attended the Faculty of Engineering, Mansoura University, in Mansoura, where he received Bachelor and Master Degrees in Engineering from the electronics (Computers Engineering and Systems) department in 2001 and 2005, respectively. He was with the Faculty of Engineering, Mansoura University, from 2001 through 2007. In April 2007, he joined the Graduate School of Advanced Integration Science, Faculty of Engineering, Chiba University, Japan, as a doctor student. He received Ph.D. Degree in Engineering in 2011. His research interests are in the fields of computer vision and pattern recognition, with special interest in material classification based on reflectance information.