Color segmentation of skin lesions with the generalizable Gaussian mixture model

Sigurdur Sigurdsson[1], Lars Kai Hansen[2] and Krzysztof T. Drzewiecki[1]

[1] Department of Reconstructive Surgery, National Hospital of Denmark, DK-2100 Copenhagen, Denmark.

[2] Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Denmark.

Introduction

In the last four years the Righospital and the Technical University of Denmark have been working on a image classification for skin lesions based on feature extraction and artificial neural network. The goal is to be able to classify skin lesions into three groups; malignant melanoma which is a very dangerous type of cancer, benign lesion which is a harmless lesion, and atypical lesion which has a risk of evolving to melanoma. Extraction of important features is vital for an accurate classification of lesions. Until now the features that have been extracted are: edge abruptness, asymmetry and color. The attention has not been focused on pigmented networks. Before extraction of features from the pigmented network is made, it is important to segment the pigmented network from other areas like structure less pigmented areas and depigmented areas. These three regions have to be segmented from each other. Each region can then be processed more to extract features that can increase the classification rate of lesions. Visual characteristics are as follows; both depigmented areas and unevenly distribution of structure less pigmented areas indicates melanoma. Regular networks usually indicates benign lesion, but thick and dark lines indicate melanoma.

The segmentation scheme suggested here is based on the assumption that the probability density of the data points (pixels) can be estimated as a Gaussian mixture model. First the lesion itself is extracted from the background of the image. The edge of the lesion is found with optimal thresholding [1], which has been developed with the extraction of the edge features [2]. Various color spaces can be used, the *RGB* color space is an obvious choice as it needs no transformation. Also the *Lab* is used, which is a nonlinear transformation of the *RGB* color space. The image is then segmented using a clustering algorithm which estimates each clusters as a multidimensional Gaussian probability density function. The optimal number of clusters is found by clustering with various number of clusters and selecting the one that minimizes a negative log-likelihood error function.

Segmentation method

The segmentation method can be divided in two, preprocessing and modeling. The preprocessing includes image median filtering, extraction of the lesion from the background with optimal thresholding, and at last the transformation from *RGB* to *Lab* color space. The modeling part includes estimating probability density with a Gaussian mixture model, the clustering algorithm and the maximum likelihood error function.

Image filtering

Before the segmentation the image is spatially filtered. This is done to prevent over segmentation, i.e. ensure smooth segmentation [3]. The filter method used here is the median filter [1]. With median filters a pixel value is replaced with the median value of the neighborhood pixels. The median filters is therefore a nonlinear filter. It is useful for removing isolated lines or pixels, while preserving the spatial resolution. But it performs badly when the number of noise pixels is more then half the number of pixels in the window. Generally the size of the filter is chosen such that it is odd and rectangular. When filtering a color image, each of the three color components in the *RGB* color space are median filtered separately.

Optimal thresholding

It is important to discard the data point outside the lesion. Otherwise the probability density will be dominated by healthy skin surrounding the lesion, which makes density estimation inside the lesion difficult and sensitive [3]. The method used here is optimal thresholding [4].

Optimal thresholding is useful when the image contains principal brightness regions. In the case of the lesion the image includes two principal brightness regions, the dark lesion and the light skin. The histogram of such a picture may be considered as an estimate of the brightness probability density function. This overall density function would be a mixture of two densities, one for the light and one for the dark region. The *RGB* image is transformed to a gray level image before thresholding. Further description of this algorithm can be found in [2].

Color transformation

One reason for using other color spaces than RGB is that experiments have shown that no single color space is best and it seems to depend somewhat on the image content [5]. The advantage of using the *Lab* color space is that it yields perceptually uniform spacing of colors, as the *Lab* is linear with visual perception, while the *RGB* is none linear. This is reasonable, as the rules for classifying lesion are made from visual perception. Another advantage is that the luminance factor L of the *Lab* color space could be discarded, as the luminance should be nearly constant for all pixels in the image. This could reduce the dimension of the data from 3 to 2, reducing data size and computation time. The a and b color components hold the color differences.

The color transformation from RGB to Lab is made under the assumption that the RGB color space is the CIE[1] standard. This is an simplification as the RGB color space is device dependent. First each pixel is transformed linearly from RGB color space to the XYZ color space with

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \tag{1}$$

where

$$R' = \frac{R}{R+G+B} \qquad B' = \frac{B}{R+G+B} \qquad G' = \frac{G}{R+G+B}$$

where the values of the RGB color components are normalized to lie between 0 and 1 [1]. Then the XYZ color space is transformed into the Lab color space using

$$L = 25\left(\frac{100Y}{Y_0}\right)^{1/3} - 16 \tag{2}$$

$$a = 500\left[\left(\frac{X}{X_0}\right)^{1/3} - \left(\frac{Y}{Y_0}\right)^{1/3}\right] \tag{3}$$

$$b = 200\left[\left(\frac{Y}{Y_0}\right)^{1/3} - \left(\frac{Z}{Z_0}\right)^{1/3}\right] \tag{4}$$

where X_0, Y_0, Z_0 is the reference white light, in this case they are all set to 1 [1].

Gaussian mixture models

A Gaussian mixture model is a model where the density function is formed as a linear combination of K functions where the number K is much less then the number N data points. The model for the density is therefore written as a mixture distribution in the form

$$p(\mathbf{x}) = \sum_{j=1}^{K} p(\mathbf{x}|j)P(j) \tag{5}$$

where $\mathbf{x}$ is the input with N data points and D dimensions, $p(\mathbf{x}|j)$ is the jth component density and $P(j)$ is the prior probability of the data point having been generated from component j of the mixture. The component density functions are limited to be Gaussian distribution functions, given as

$$p(\mathbf{x}|j) = \frac{1}{(2\pi)^{D/2}\sqrt{\det \boldsymbol{\Sigma}_j}} \exp\left[-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)\right] \tag{6}$$

[1] Commission Internationale l'Eclairage

where $\boldsymbol{\mu}_j$ is the mean and $\boldsymbol{\Sigma}_j$ is the covariance matrix of the jth component. The Gaussian distribution function satisfy the constraints

$$\int p(\mathbf{x}|j)d\mathbf{x} = 1 \tag{7}$$

The prior probabilities are chosen to satisfy the constraints

$$\sum_{j=1}^{M} P(j) = 1 \tag{8}$$

$$0 \leq P(j) \leq 1. \tag{9}$$

Clustering algorithm

A clustering algorithm is used to find the parameter of each basis function, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$, and the probability for each of each basis function, $P(j)$. The algorithm involves a simple re-estimation procedure. The algorithm seeks to partition the data points into K disjoint subsets $\mathcal{S}_j$ containing N_j data points, in such a way that each data point is appointed the cluster which gives the highest probability $p(\mathbf{x}|j)$.

Some variables need to be initialized in the start. The $P(j)$ is set equal for all functions to K^{-1}. The covariance matrices $\boldsymbol{\Sigma}_j$ are also the same for all clusters, calculated using all the data points i.e.,

$$\boldsymbol{\Sigma}_{j,init} = (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T \tag{10}$$

where $\boldsymbol{\mu}$ is the mean of the data set given by

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^n \tag{11}$$

and $j = 1, 2, \ldots, K$. The mean of each Gaussian basis function $\boldsymbol{\mu}_j$ is initialized by selecting at random K data points in the data set.

The algorithm is then iterated as follows. Each data point is assigned a basis function which gives the the highest probability $p(\mathbf{x}|j)$, using equation (6). The mean $\boldsymbol{\mu}_j$ is given by

$$\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_{n\in\mathcal{S}_j} \mathbf{x}^n. \tag{12}$$

The covariance for each function $\boldsymbol{\Sigma}_j$ is given by

$$\boldsymbol{\Sigma}_j = \left(\mathbf{x}_{\mathcal{S}_j} - \boldsymbol{\mu}_j\right)\left(\mathbf{x}_{\mathcal{S}_j} - \boldsymbol{\mu}_j\right)^T \tag{13}$$

where $\mathbf{x}_{\mathcal{S}_j}$ are data point belonging to cluster j, and the prior probability $P(j)$ is found using

$$P(j) = \frac{N_j}{N}. \tag{14}$$

The data points are assigned again to a basis function which the probability is highest and the parameters are recomputed. This continues until there is no further change in grouping of the data points.

The data set is divided in two where one part is used to compute the mean $\boldsymbol{\mu}_j$ and the other part is used to compute the covariance $\boldsymbol{\Sigma}_j$. The prior probability is computed using all the data points.

Maximum likelihood

The problem of the clustering algorithm is that the number of clusters K has to be decided in advance. This can be solved using a error function which calculates the error for different number of basis functions and then selects the K which gives the lowest error. The maximum likelihood is used to measure the performance of the clustering. The likelihood function is given by

$$L = \prod_{n=1}^{N} p(\mathbf{x}^n). \tag{15}$$

The error function derived from the likelihood function is called the negative log-likelihood error function given by

$$E = -\ln L = -\sum_{n=1}^{N} \ln \left[\sum_{j=1}^{K} p(\mathbf{x}^n|j)P(j) \right]. \tag{16}$$

in [6]

As we have a huge number of data points for each lesion, it is reasonable to use a independent dataset, a test set, to measure the generalization performance with respect to different number of clusters.

Results

The segmentation method is first tested on a synthetic image. The image is shown in figure 1(a) and consists of three equally sized areas with different colors; red, green and blue. The image size is 300 × 400 pixels or 120.000 pixels which is a typical number of pixels, for a skin lesion in a dermatoscopic image. The top of the image is green, the middle is blue and the bottom is red. All the three color regions have Gaussian distributed pixel values and have the following mean

$$\boldsymbol{\mu}_{red} = [0.6 \;\; 0.4 \;\; 0.5] \qquad \boldsymbol{\mu}_{green} = [0.5 \;\; 0.6 \;\; 0.4] \qquad \boldsymbol{\mu}_{blue} = [0.4 \;\; 0.5 \;\; 0.6]$$

where the values are in the normalized RGB color space, i.e. all values are between 0 and 1. The mean of the colors is the same for each dimension in the color space. The covariance matrix for these three colors are the same and given as

$$\boldsymbol{\Sigma}_{red,green,blue} = \begin{bmatrix} (1/16)^2 & 0 & 0 \\ 0 & (1/16)^2 & 0 \\ 0 & 0 & (1/16)^2 \end{bmatrix}$$

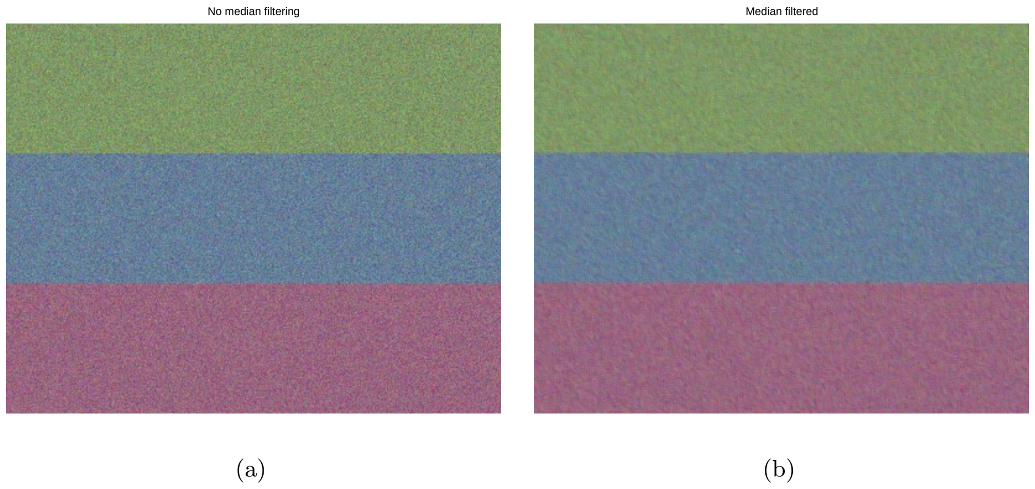


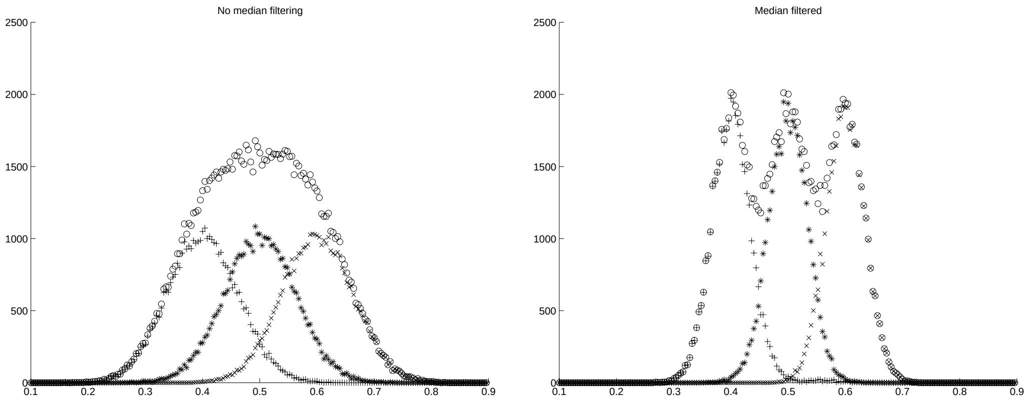
Figure 1 (a) the synthetic image and (b) the synthetic image median filtered with 3×3 filter.

where the color dimensions within each of the three colors are uncorrelated. The brightness of these colors are very similar, especially the color in the middle and bottom of the image are hard to tell apart. The image is filtered with a 3×3 median filter, and the filtered image is shown in figure 1(b). The filtering makes the image spatially smoother.

The histograms for one dimension for both the original image and the filtered image are shown in figure 2(a) and (b), respectively. Also the contribution of each color in the histogram is shown in figure 2. Only one dimension is shown as the independent histograms for each of the three dimensions in the color space look exactly the same. The histogram for the original image, figure 2 (a) the three Gaussian function are very close to each other with great overlap, making it impossible to see that there is more than one function. After median filtering the three colors have less variance which makes it possible to see that there are three functions. There is still overlap, but segmentation using all three dimensions will be less sensitive to overlap.

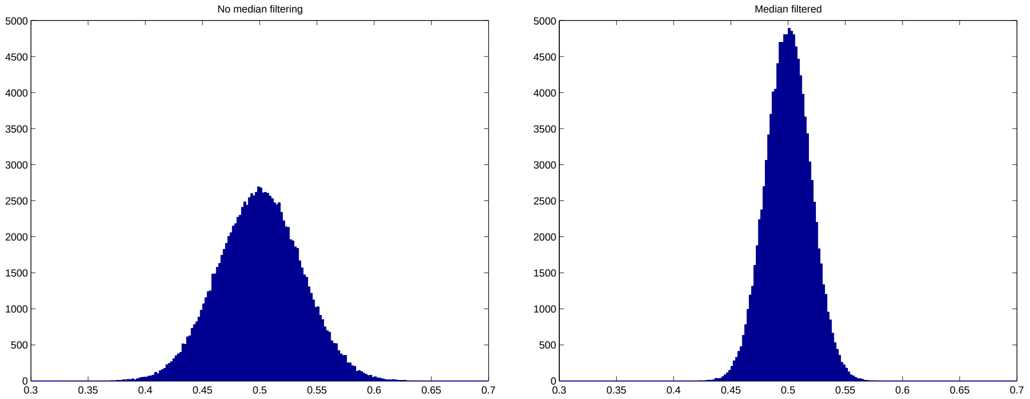
Color images are sometimes transformed to luminance images and then segmented with thresholding. Both the original image and the filtered image are transformed from RGB color space to luminance image with $L = (R + G + B)/3$ where the R, G and B are the RGB values of a single pixel, and the L is the corresponding luminance value of the pixel. The histogram of the luminance image for both the original image and the filtered image is shown in figure 3(a) and (b) respectively. These histograms show that the images are impossible to segment successfully with thresholding methods using the luminance. Thresholding methods require that the pixels have values that make at least two dominant groups. Here, only one group is visible.

The algorithm is now used to segment the original image and the filtered image. Some



(a) (b)

Figure 2 Histogram for one dimension in the RGB color space for (a) the synthetic image and (b) the filtered image.



(a) (b)

Figure 3 Histogram of the luminance image transformed from (a) the original image and (b) the filtered image.

practical things have to be taken into consideration. Iterating each time with over 100.000 pixels makes the algorithm computationally expensive and time consuming. To reduce computation time the data is randomized and divided into blocks of data with 10.000 pixels in each block. In each iteration a new block of data is used, thus making use of all the data available. The algorithm is iterated 100 times and 5.000 pixels are reserved as a test set to estimate the generalization of the model.

The average negative log-likelihood error of 10 runs for the original and the filtered as a function of 1 to 3 Gaussian functions are computed and shown in figure 4. The interesting part is that it was impossible to calculate the error for more than 3 Gaussian functions. When there were more than three the algorithm simply set the prior probability of the extra functions to zero so there were only three clusters left. This means that the algorithm can to some extent regulate the number of basis function and that it cannot estimate more functions than it can detect. In this case the image is very much adapted to the algorithm so the results of using real images could be different. Figure 4 shows that the error has a minimum at 3 Gaussian functions, which is the number of colored regions in the image.

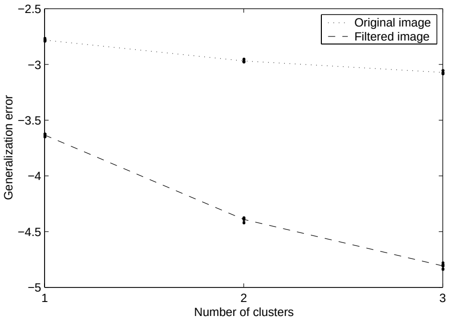


Figure 4 The negative log-likelihood error as a function of the number of Gaussian functions. Note that it was not possible to calculate the error for more than 3 clusters as the algorithm set the prior probability to zero for all clusters more that 3.

The results of the segmentation of the original image and the filtered image is shown in figure 5(a) and (b), respectively. The original image is diffused as the color of the pixels overlap somewhat as no spatial correction is applied before segmentation. The ratio of misclassified pixels is around 5 percent. The filtered image is visually much better segmented and misclassified pixels are only about 0.7 percent. These misclassified pixels could be eliminated with help of image morphology on the segmented image.

The other experiment is segmentation of a skin lesion. In figure 6 the image of the lesion is shown. The edge of the lesion is shown in the figure and is marked with a white line. The number of pixel representing the lesion are over 120.000. This lesion has two of the three types of regions, depigmentation and pigmented network. The depigmented areas are marked in figure 6. The rest of the lesion has pigmented network which varies in intensity, from very dark network to light network. Examples of these areas are also marked in figure 6. This difference indicates that the pigmented network will be divided into smaller regions.

The size of median filter is chose to be 21 × 21. By using such a large filter the air bubbles are completely removed and the image will be smooth and easier to segment.

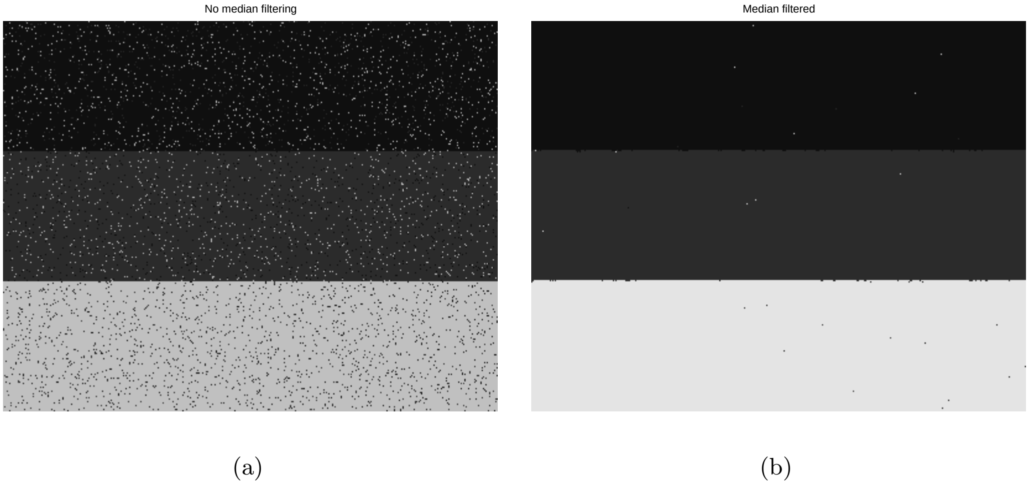


Figure 5 The segmented images using (a) the original image and (b) the filtered image.

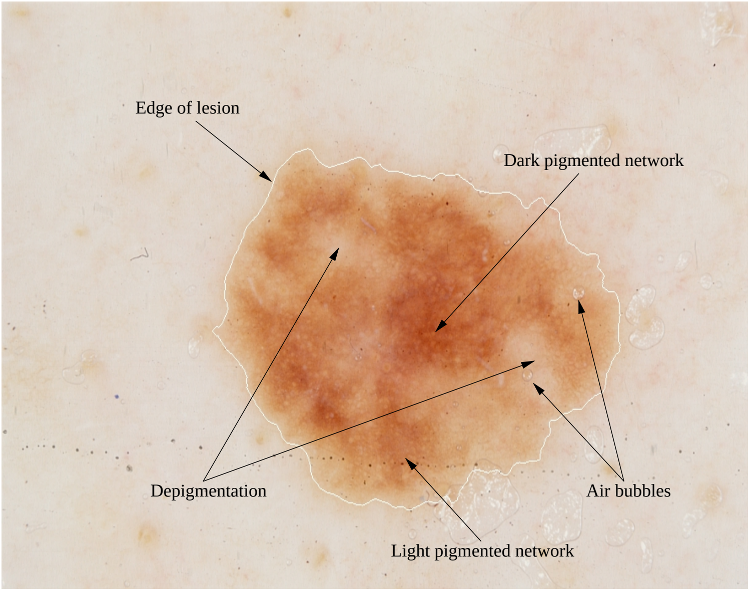


Figure 6 The original image of the lesion, with depigmentation and pigmented network.

The *RGB* color space histograms, independent for each dimension of the filtered image, shown in figure 7. These histograms show that there are no clear groups of pixels, in each of the single dimensions, that could easily be segmented. It is possible that the grouping is three dimensional. The same thing can be seen in the histogram for the *Lab* color space, shown in figure 8. If we look at the *L* part or the luminance of the *Lab* color space, it varies very little compared to the range of *L* which is from 1 to 100. This supports a *ab* sub-color space, using only the color differences and discarding the luminance part. A two dimensional histogram of the *ab* color space is shown in figure 9. There are no distinctive groups, as the histogram is like a mountain ridge with many small peaks. The conclusion is, after looking at the histograms for all the color spaces, that segmentation of a skin lesion is not an easy task. The colors are very similar, and no clear boundaries.

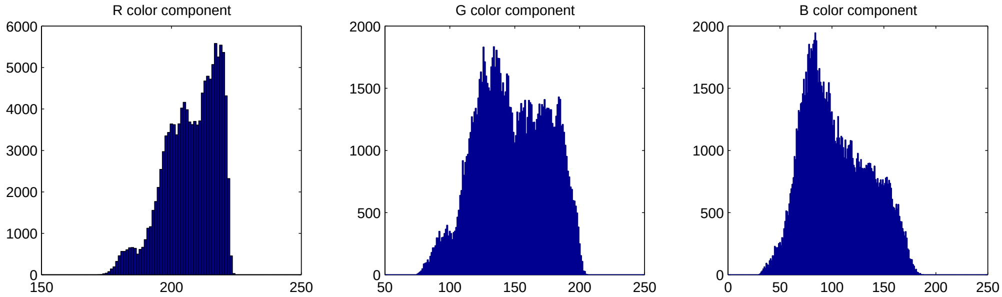


Figure 7 Histograms of each dimension in the *RGB* color space for the filtered lesion image

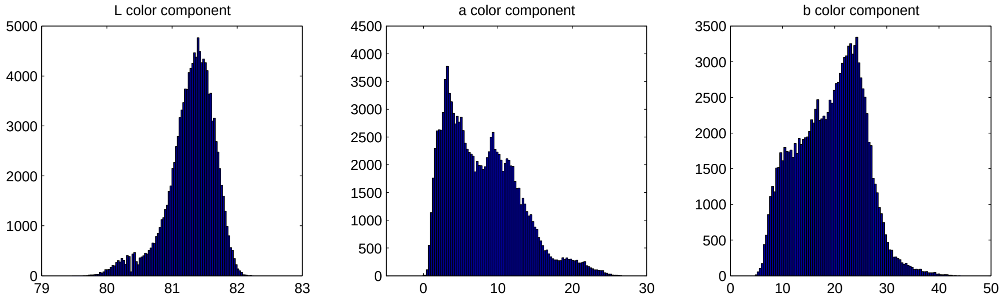


Figure 8 Histogram of each dimension in the *Lab* color space for the filtered lesion image.

The algorithm is applied to the three color spaces, *RGB* and *Lab*, and the sub-color space *ab*. The data is divided into blocks for each iteration as before. The algorithm is iterated 200 times, which is double the number used with the synthetic image. This increase in iterations is because the lesion image is not as well defined as the synthetic image, and the algorithm uses more iterations to converge.

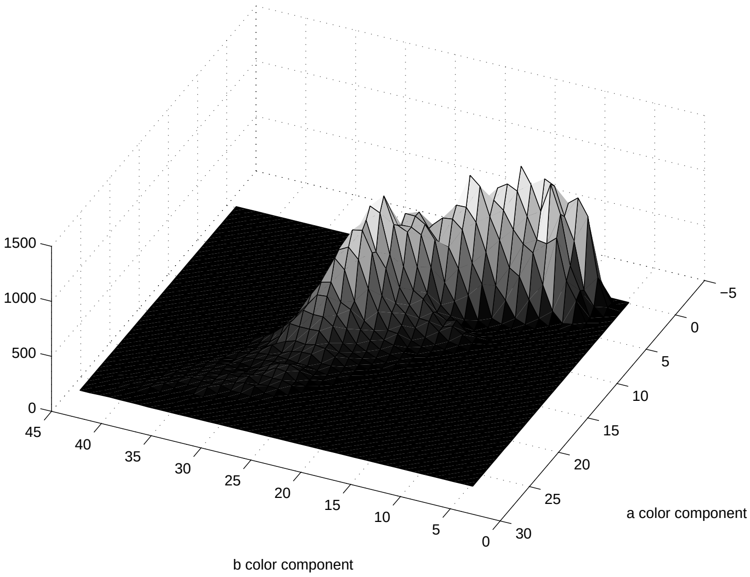


Figure 9 A two dimensional histogram for the *ab* sub-color space for the filtered lesion image.

In figure 10 the error of 13 run as a function of 1 to 10 clusters is shown for the *RGB* color space. The error has a minimum at 9 clusters. The problem is that the error at 9 clusters is not very different from the error at 7 to 10 clusters, which makes the assumption that a minimum of error is at 9 clusters statistically vague. It is interesting to see the

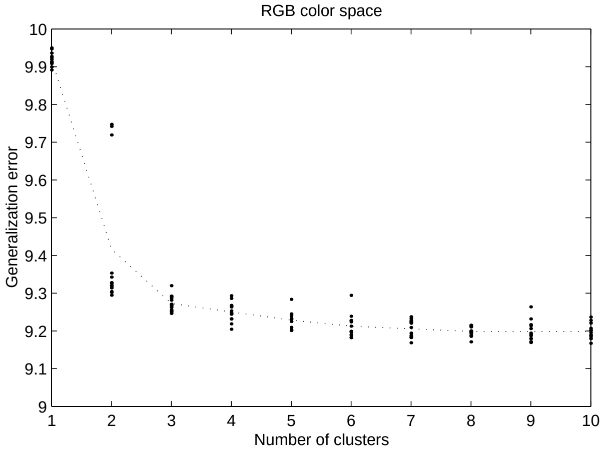


Figure 10 The dotted line shows the average error of 13 runs for 1 to 10 clusters, using the *RGB* color space. The dots represent the error for individual runs.

distribution of error at 2 clusters. Two models are estimated with different errors. This is in contrast with models using different number of clusters.

In figure 11 the error of 13 runs as a function of 1 to 10 clusters for the *Lab* color space is shown. The error has no minimum within 10 clusters. This limits the use of the *Lab* color space for segmenting lesions with this method, as an optimal number of clusters cannot be found.

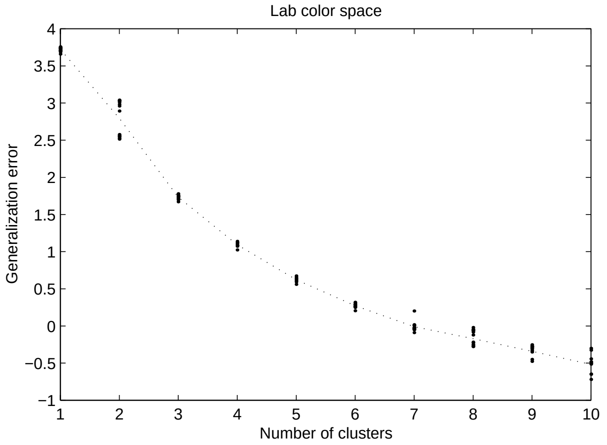


Figure 11 The dotted line shows the average error of 13 runs for 1 to 10 clusters, using the *Lab* color space. The dots represent the error for individual runs.

In figure 12 the error of 13 runs as a function of 1 to 10 clusters for the *ab* color space is shown. The error has a very small minimum at 6 clusters. The error value from 3 to 10

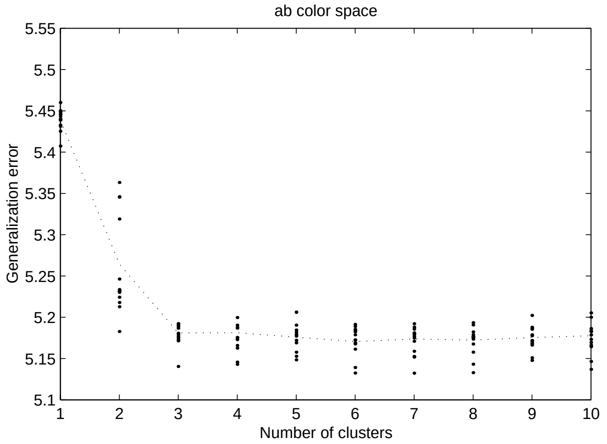


Figure 12 The dotted line shows the average error of 13 runs for 1 to 10 clusters, using the *ab* color space. The dots represent the error for individual runs.

clusters changes very little, indicating that adding more clusters after 3 clusters improves the estimate of the input density very little. This could be because the first 3 clusters make the shape of the input density while more clusters estimate single peaks in the density, as seen in figure 9.

After evaluating the generalization error for different color spaces and different number of clusters, the only realistic model for segmentation is using the *ab* color space with 3 clusters. For comparison the *RGB* color space with 9 clusters and *ab* color space with 6 clusters will also be segmented. The result of the segmentation for the *RGB* color space with 9 clusters, *ab* color space with 6 clusters and 3 clusters are shown in figure 13(a), (b) and (c) respectively.

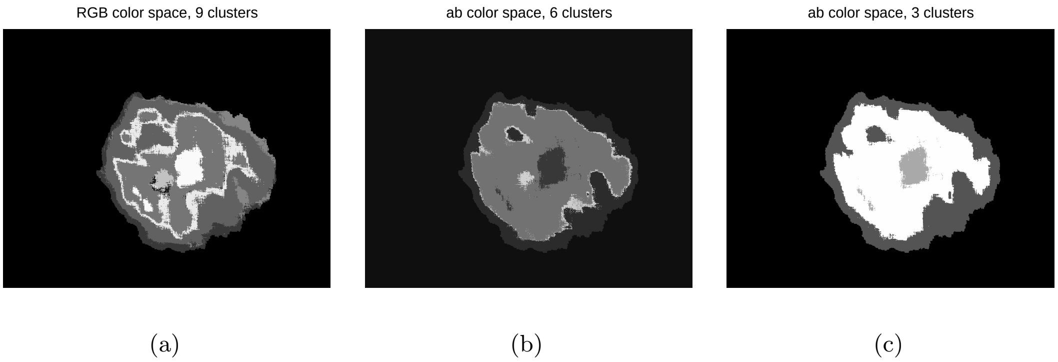
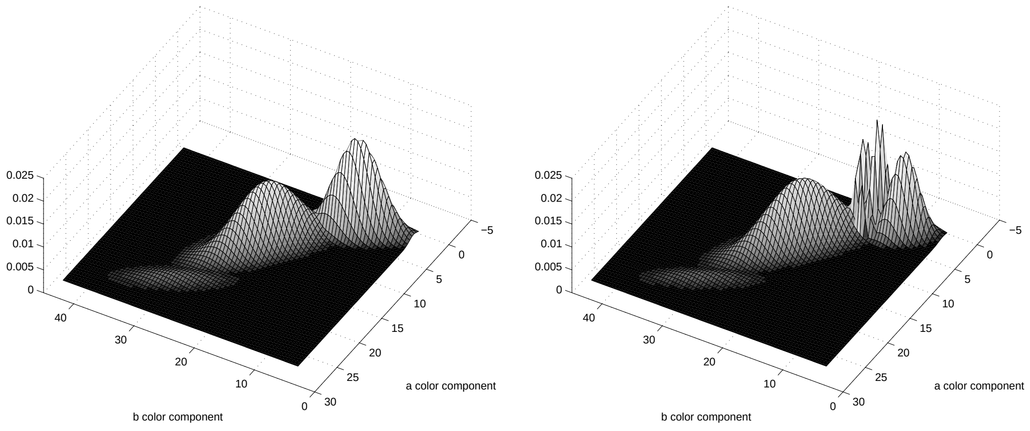

RGB color space, 9 clusters
ab color space, 6 clusters
ab color space, 3 clusters
(a)
(b)
(c)

Figure 13 The lesion image segmented with (a) *RGB* color space and 9 clusters , (b) *ab* color space and 6 clusters and (c) *ab* color space and 3 clusters.

The segmented image in figure 13(a) using 9 clusters has far to many clusters and not useful for segmenting skin lesions from a dermatoscopic viewpoint.

The two other segmented images, figure13(b) and (c) are similar in two ways. Firstly, the pigmented network is divided into dark and light network. Even though the network is not grouped together, this is no catastrophe. The color of the lines in the network could indicate different types of lesion, and this could be computed for each part of the pigmented network. Secondly, the edge of the lesion is clustered with the depigmented areas. This is a problem as the skin lesion without any depigmentation will also have skin like edges. This has to be solved, either by using another method to extract the lesion from the skin, or to use some correction after segmentation.

The difference between figure 13(b) and (c) using 3 and 6 clusters is best by examining the probability density of the Gaussian functions, shown in figure 14(a) and (b), respectively. In figure 14(a) the Gaussian function representing the dark pigmented network is on the left, the light pigmented network is in the middle, and the the depigmentation is on the right. The difference between 14(a) and (b) is that the depigmentation represented by one Gaussian function in (a) is represented by 4 functions in (b). 3 of these Gaussian functions are very narrow, which means that the corresponding clusters have few pixels



(a)

(b)

Figure 14 The Gaussian probability density function for the *ab* sub-color space using (a) 3 cluster and (b) 6 clusters.

compared to the other Gaussian functions. Clusters with few pixels are not useful for segmentation.

The results shows that of the three color spaces, *RGB*, *Lab* and *ab*, the *ab* gives the best results. Using 3 clusters gives visually the best result, while being close to the value of the minimum generalization error. Minimizing an error function to optimize a model does not mean that the segmentation will be good, as the “correct” segmentation is based on human perception which is difficult to measure with an error function.

Conclusion

We have shown that segmentation of skin lesions is difficult. The density of the pixel values (histogram) show that there are no clear groups of pixels that can easily be isolated. This makes segmentation methods using threshold useless.

The proposed method estimates the pixel density using multidimensional Gaussian functions. The pixels are then clustered according to the probability of the functions. The number of Gaussian functions is selected so that a generalization error is minimized. This can be a problem as error tends to decrease with increasing number of functions. By using color space transformation to extract color components and removing image luminance a usable error measurement was obtained.

The result is reasonable considered the density of the pixels. Using 3 clusters the pixels were divided into dark and light pigmented network and depigmented areas. Pixels around the edges tend to be classified as depigmentation. This is difficult to avoid, regardless of segmentation method, the color of edge pixels is very similar to pixels with in depigmentation. A solution could be to use some a priory information that skin surrounding the lesion is similar to depigmentation.

References

[1] Jain, Anil K. *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[2] Hintz-Madsen, Mads *The Probabilistic Framework for Classification of Dermatoscopic Images.* Ph.d thesis, Section for Digital Signal Processing, Technical University of Denmark, Denmark, 1998.

[3] Schmid, Ph and Fischer, S. *Colour Segmentation for the Analysis of Pigmented Skin Lesions.* Signal Processing Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland.

[4] Gonzalez, Rafael C. and Wintz, Paul. *Image Processing.* Addison-Wesley Publishing Company, 1987.

[5] Gauch, John and Hsia, Chi-Wan. *A Comparison of Three Color Image Segmentation algorithms in four color spaces.* SPIE, vol 1818, Visual Communication and Image Processing, pp. 1168-1181, 1992.

[6] Bishop, Christopher M. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.