Adaptive Color Spaces based on Multivariate Gaussian Distributions for Color Image Segmentation

Alexander Ihlow and Udo Seiffert

Leibniz Institute of Plant Genetics and
Crop Plant Research (IPK) Gatersleben
Corrensstr. 3, 06466 Gatersleben, Germany
Pattern Recognition Group
{ihlow, seiffert}@ipk-gatersleben.de
http://mue.bic-gh.de

Abstract

We formulate an adaptive color space for segmenting an image into the two classes *"object of interest"* and *"background"* by using well-established methods from statistical pattern recognition. Both classes are modeled by a multivariate Gaussian distribution whose actual parameters are estimated via the Expectation Maximization (EM) algorithm. The output grayscale feature image is derived as the distance of each pixel's color to the decision boundary which is shaped bewteen the two class models. Based on this feature image, which provides a maximum discriminatory power with respect to the underlying model assumptions, the actual segmentation can be performed with appropriate methods from grayscale image processing. This adaptive color space is a practical tool for homogeneously colored scenes, as they appear, e.g., in microscopic images of biotechnical fundamental research.

1 Introduction

Color image segmentation is a fundamental problem in image processing. Recent developments like SIOX [1] and GrabCut [2] demonstrate considerable advances towards universal segmentation tools even for complex scenes. Notwithstanding, an all-purpose solution seems to be an unreachable ideal.

An essential preprocessing step for color image segmentation is to transform the vector-valued color information into another representation, for instance into a scalar-valued image which is intended to offer preferably a maximum discriminatory power between the objects of interest and the background. We want to call this the feature image, since it represents the intended features of the underlying color image. Generally, there exists no "standard transform" to perform this task. However, by using techniques from statistical pattern recognition it is possible to gain a color model which automatically adapts itself to the actual color distribution of the particular image and to derive a feature image out of this color model which ultimately meets the desired maximum disriminatory power. As a matter of course, when dealing with statistical models, the data needs to match the underlying model assumptions. The model which is used in the presented approach is the multivariate Gaussian distribution — a very popular and easy to handle one. Mapped onto the image segmentation problem, this model is valid for homogenously colored objects both in case of the *"object of interest"* and the *"background"*. This is true in a number of practically relevant cases. Especially in the context of biological image processing tasks the problem of identifying dyed objects arises in various facets. We demonstrate the worth of this parametric approach on some examples from biotechnical fundamental research.

This color space transformation procedure works as follows: First, the actual class parameters of the *"background"* and *"object of interest"* are estimated out of a rough initialization by using the EM algorithm (Section 2). Clearly, based on this two-class model it was easy to calculate the maximum likelihood class membership of each pixel which would result in a binary image.[1] But our intention is not decision-making, i.e., assigning a *binary* label to each pixel, but transforming the vector-valued color data into a *grayscale* image which actually reflects the *distance* of the pixel-color to the decision boundary between the classes. Therefore, we use appropriate distance measures to a linear or quadratic surface (Section 3). Since this transformation decouples the color information from the subsequent postprocessing (segmentation) algorithms, the whole pool of grayscale image segmentation algorithms like thresholding, region growing, edge detection, active contour models, level sets, morphological operations, etc. can be applied.

2 Data Model: Mixture of Gaussians

All subsequent considerations are based on the assumption that the entire image color distribution can be decomposed into $K = 2$ classes — namely the *"object of interest"* and the *"background"*. Hence, the distribution of the color samples $\mathbf{x}$ is modeled by the mixture

$$p(\mathbf{x}) = \sum_{k=1}^{K} P_k \, p(\mathbf{x} \,|\, \Theta_k) \qquad \text{with} \quad \sum_{k=1}^{K} P_k = 1 \quad \text{and} \quad \int_{\mathbf{x}} p(\mathbf{x} \,|\, \Theta_k) \, \mathrm{d}\mathbf{x} = 1 \,, \tag{1}$$

where P_k is the a priori probability of class k and Θ_k denotes the set of parameters which describes the distribution of class k.
Choosing an appropriate distribution for the mixture components is a crucial step. To keep the subsequent procedure mathematically manageable at all, we rely on the popular multivariate Gaussian distribution. Its equal-probability surfaces describe (hyper)ellipsoids in the d-dimensional space. Here, $d = 3$ corresponding to the RGB color space. The model parameters Θ_k consist of the mean vector $\boldsymbol{\mu}_k \in \mathbb{R}^{d \times 1}$ which describes the center of the ellipsoid, and the covariance matrix $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$ which determines its shape and orientation.

$$p(\mathbf{x} \,|\, \Theta_k = \{\boldsymbol{\mu}_k, \, \boldsymbol{\Sigma}_k\}) = \frac{1}{\sqrt{(2\pi)^d \, |\boldsymbol{\Sigma}_k|}} \, \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)} \tag{2}$$

Having defined the parametric data model, as a next step we need to estimate the model parameters out of the given data set, i.e., from the particular image which we want to transform into the adaptive color space. This is done with the Expectation Maximization (EM) algorithm.

Estimation of Model Parameters using the EM Algorithm

The Expectation Maximization (EM) algorithm [4, 5] is an iterative technique for finding the maximum likelihood parameter estimates when matching a distribution on a given data set. During the iterations, first the probability (at iteration step t) of all N data samples $\mathbf{x}_n$ to belong to class k is calculated by Bayes' theorem, which is known as the *expectation step.*

$$P^t(k \,|\, \mathbf{x}_n) = \frac{P_k^t \; p(\mathbf{x}_n \,|\, \Theta_k^t)}{\sum_{j=1}^{K} P_j^t \; p(\mathbf{x}_n \,|\, \Theta_j^t)} \tag{3}$$

[1] Building more than two classes like, e.g., in [3] would result in a multi-label image.

In the subsequent *maximization step*, an updated parameter set for the iteration step $t+1$ containing the prior probabilities, mean vectors, and covariance matrices for each class is calculated.

$$P_k^{t+1} = \frac{1}{N} \sum_{n=1}^{N} P^t(k \,|\, \mathbf{x}_n) \tag{4}$$

$$\boldsymbol{\mu}_k^{t+1} = \frac{1}{N P_k^{t+1}} \sum_{n=1}^{N} P^t(k \,|\, \mathbf{x}_n)\, \mathbf{x}_n \tag{5}$$

$$\boldsymbol{\Sigma}_k^{t+1} = \frac{1}{N P_k^{t+1}} \sum_{n=1}^{N} P^t(k \,|\, \mathbf{x}_n)\, (\mathbf{x}_n - \boldsymbol{\mu}_k^{t+1})\, (\mathbf{x}_n - \boldsymbol{\mu}_k^{t+1})^T \tag{6}$$

As initialization, a rough guess of the class parameters is adequate. The algorithm is terminated at some stopping criterion, e.g., when the resulting class labelling of the data vectors does not change anymore.
For a deeper understanding of the EM algorithm as well as the derivation of the parameter estimation equations the interested reader may refer to [6].

3 Adaptive Color Spaces

Having obtained the actual model parameters for the two classes *"object of interest"* and *"background"*, we are ready to transform the color image into the adaptive color space, i.e., to gain the desired feature image. As already mentioned in the introduction, we are not interested in a binary image which would correspond to a maximum likelihood decision of each color pixel, but in a *grayscale* image which actually reflects the *distance* of the pixel-color to the decision boundary between the classes. Subsequently, appropriate grayscale image segmentation algorithms can be applied onto this feature image.

3.1 Quadratic and Linear Decision Boundary

Actually, this adaptive color space is obtained as follows: Due to the $K = 2$ ellipsoidal distributions (which reflect the color of the *"object of interest"* and the *"background"*) a quadratic decision boundary is shaped between these two classes i and j which is described by the implicit equation [7]

$$\mathbf{x}^T \mathbf{A} \mathbf{x} + 2\mathbf{a}^T \mathbf{x} + a_0 = 0\,. \tag{7}$$

Its parameters are given by (for the derivation of these equations please refer to Appendix B)

$$\mathbf{A} = \boldsymbol{\Sigma}_j^{-1} - \boldsymbol{\Sigma}_i^{-1} \tag{8}$$

$$\mathbf{a} = \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i - \boldsymbol{\Sigma}_j^{-1} \boldsymbol{\mu}_j \tag{9}$$

$$a_0 = -2\ln(P_j) + 2\ln(P_i) + \ln(|\boldsymbol{\Sigma}_j|) - \ln(|\boldsymbol{\Sigma}_i|) + \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}_j^{-1} \boldsymbol{\mu}_j - \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i\,. \tag{10}$$

If the covariance of both classes are equal, then $\mathbf{A} = \mathbf{0}$ and a linear decision boundary is obtained. A linear decision boundary can be forced by a linearization operation. In this case, the covariances $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\Sigma}_j$ in (9) and (10) are replaced by the pooled covariance matrix [7]

$$\boldsymbol{\Sigma}_{\text{pooled}} = \frac{1}{N-K} \sum_{k=1}^{K} (n_k - 1)\, \boldsymbol{\Sigma}_k\,, \tag{11}$$

where n_k denotes the number of samples in class k.

3.2 Distances to Linear and Quadratic Surfaces as Adaptive Color Space

Now, based on the quadratic or linearized surface, the adaptive color space can be defined by calculating the distance of each color sample to the decision boundary.

Linear Decision Boundary — Euclidean Distance

Since it is much easier to deal with the linear case, we start with the corresponding linear adaptive color space. The distance of a query point $\mathbf{p}$ to the linear decision boundary is just given by

$$d_l = 2\mathbf{a}^T\mathbf{p} + a_0\,. \tag{12}$$

Remember to use the pooled covariance matrix (11) for calculating the parameters $\mathbf{a}$ and a_0. Furthermore, Equation (12) should be normalized in such a way that the vector norm of $\mathbf{a}$ is $|\mathbf{a}| = \frac{1}{2}$.
In fact, this linear distance measure represents the well known FISHER linear discriminant [8].

Quadratic Decision Boundary — Perpendicular Distance

Generally, a nonlinear decision boundary provides a more accurate detachment of the class clusters and is therefore more suitable. As already mentioned, in case of cluster approximation by multivariate Gaussians, the decision boundary is inherently of a quadratic nature. There exist different definitions of distance measures to those curved surfaces. Thereby, the *perpendicular distance* d_p provides the most intuitive generalization of the distance measure to the linear decision boundary, since it represents the Euclidean distance between the query point $\mathbf{p}$ and the closest point on the surface. It is given by

$$d_p = \min_{\mathbf{x}} ||\mathbf{p} - \mathbf{x}|| \qquad \text{with} \qquad \mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{a}^T\mathbf{x} + a_0 = 0\,. \tag{13}$$

Unfortunately, the calculation of the perpendicular distance is not straightforward at all and involves finding the roots of a 6th-order polynomial. The step-by-step solution is given in Appendix A.

Quadratic Decision Boundary — Algebraic Distance

The *algebraic distance* of a query point $\mathbf{p}$ to the surface is defined by

$$d_a = \mathbf{p}^T\mathbf{A}\mathbf{p} + 2\mathbf{a}^T\mathbf{p} + a_0\,. \tag{14}$$

According to (22) and (23) (see Appendix B) this distance measure represents the difference of the generalized distances. It can be written also as $d_a = D_j^2 - D_i^2$. Again, as in case of the Euclidean distance to the linear decision boundary, a normalization should be applied. We suggest to normalize Equation (14) such that $|\mathbf{a}| = \frac{1}{2}$.

Other Distance Measures

In case of an ellipsoidal surface, a *radial distance* as well as a *normalized radial distance* [9] can be defined. However, due to this dramatic limitation of an ellipsoidal decision boundary this case is not of practical relevance.

3.3 Examples

In the end, the technique of parametrizing the color distribution by two multivariate Gaussians and obtaining a quadratic or a linearized decision boundary in between provides three applicable adaptive color spaces: They are expressed as the perpendicular distance d_p to the quadratic decision boundary, the algebraic distance d_a to the quadratic decision boundary, and the Euclidean distance d_l to the linearized decision boundary.
The performance of this approach is demonstrated on the figure pages. Let us exemplarily start with the first one: Out of the color image (Figure 1), first the actual class parameters are calculated via the EM algorithm. The resulting class labeling is shown in the clusters in Figure 2 (blue *"background"* and red *"object of interest"*). Figure 3 shows the ellipsoids of the parametric model and the resulting quadratic decision boundary. In this particular case, this quadric is shaped as a one-sheet hyperboloid. For the reason of clearness, the linearized decision boundary is not shown in the plot. Finally, Figure 4 shows the resulting grayscale feature images in pseudocolor and the corresponding probability density functions (PDF). Note that the pseudocolor scale is mapped on the abscissa of the PDF plots. Subfigure a) and b) show the feature images based on the measure d_p and d_l, respectively. Bottom, Subfigure c) and d) show the algebraic distance d_a: In Subfigure d), the algebraic distance was nonlinearly scaled by a factor and an exponential. Both parameters were set by numerically minimizing the mean squared error (MMSE) between the perpendicular distance and the algebraic distance. The intention of this nonlinear transformation is to approximate the expensive[2] calculation of the perpendicular distance by the algebraic distance. Clearly, appropriate values for the two transformation parameters have to be determined individually for each family of images. As a result, the nonlinearly transformed algebraic distance can be used as an inexpensive approximation of the perpendicular distance.
Figure 5 *et seqq.* and Figure 9 *et seqq.* show two additional examples. Sometimes it is sufficient to use the adaptive color space based on the linear decision boundary (Subfigure b). This strongly depends on the particular image material and the desired segmentation quality. In the example of Figure 9 *et seqq.*, definitely the quadratic decision boundary should be used since it provides a much better equalized background than the linear one.
Based on these feature images the actual segmentation can be performed using appropriate grayscale image segmentation methods like thresholding, region growing, edge detection, active contour models, level sets, morphological operations, etc.

4 Conclusions

We have demonstrated a technique to transform a color image into a grayscale feature image of maximum discriminatory power bewteen the two classes *"object of interest"* and *"background"* using well-established procedures from statistical pattern recognition. The proposed method requires the classes to be homogeneously colored to be describable as multivariate Gaussian distributions. Based on the quadratic (or linearized) decision boundary between these two distributions, the feature image is calculated as a distance measure of each pixel's color to the decision boundary. Due to the high computational burden of the perpendicular distance, we proposed to use a nonlinearly scaled version of the algebraic distance instead which provides a similar performance. Depending on the data and the desired quality of the segmentation result, the performance of the linearized solution (which in fact represents FISHER's linear discriminant) can be sufficient, too.

[2]Transforming a 1 megapixel image on a Pentium® 4 processor with 2.66 GHz, the calculation of the algebraic distance takes less than half a second. In contrast, calculating the perpendicular distance needs several minutes since each pixel involves finding the roots of a 6th-oder polynomial. This was done by calculating the eigenvalues of the 6×6 companion matrix [10].

Acknowledgements. We thank Patrick Schweizer from the Transcriptome Analysis Group at the Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben, and Katharina Goellner from the Molecular Plant Microbe Interarctions Group at the Max Planck Institute for Plant Breeding Research, Cologne, for providing the image material. This work was supported by the German Ministry of Education and Research (BMBF) under grant 0312706A.

A Perpendicular Distance to a Quadratic Surface

The distance computation problem regarding quadratic curves and surfaces is extensively treated by LENNERZ [11, 12]. To find the distance of a query point $\mathbf{p}$ to the quadratic functional $\mathbf{x}^T\mathbf{A}\mathbf{x}+2\mathbf{a}^T\mathbf{x}+a_0 = 0$, two main steps are required: First, the quadric the has to be transformed into the normal form (the query point has to be transformed as well) before actually the distance is calculated by solving a 6th-order polynomial.

A.1 Transforming the Quadratic Functional to the Normal Form

A quadric is in normal form if the matrix $\mathbf{A}$ is diagonal and $\mathbf{Aa} = \mathbf{0}$. To transform the arbitrary quadric $\mathbf{y}^T\mathbf{B}\mathbf{y} + 2\mathbf{b}^T\mathbf{y} + b_0 = 0$ into normal form [12], first $\mathbf{B}$ is diagonalized to

$$\mathbf{D} = \mathbf{R}^T\mathbf{B}\mathbf{R}\,, \tag{15}$$

where $\mathbf{R}$ is a rotation matrix, obtained by the eigenvalue decomposition of $\mathbf{B}$. This coordinate transformation is applied in terms of $\mathbf{x} = \mathbf{R}^T\mathbf{y}$ which gives the surface equation

$$\mathbf{x}^T\mathbf{D}\mathbf{x} + 2\mathbf{d}^T\mathbf{x} + d_0 = 0\,, \qquad \text{with} \quad \mathbf{D} = \operatorname{diag}\{\lambda_1, \lambda_2, \lambda_3\}, \quad \mathbf{d} = \mathbf{R}^T\mathbf{b}, \quad \text{and} \quad d_0 = b_0.$$

After this rotation, a translation of the coordinate system by the vector $\mathbf{t}$ is introduced. Therefore, $\mathbf{x}$ is substituted by $\mathbf{x} + \mathbf{t}$, yielding the surface equation

$$\mathbf{x}^T\mathbf{D}\mathbf{x} + 2\tilde{\mathbf{d}}^T\mathbf{x} + \tilde{d}_0 = 0\,, \qquad \text{with} \quad \tilde{\mathbf{d}} = \mathbf{D}\mathbf{t} + \mathbf{d} \quad \text{and} \quad \tilde{d}_0 = \mathbf{t}^T\mathbf{D}\mathbf{t} + 2\mathbf{d}^T\mathbf{t} + d_0.$$

Choosing the elements of the vector $\mathbf{t}$ according to

$$t_i = \begin{cases} -\frac{d_i}{\lambda_i} & \text{if } \lambda_i \neq 0, \\ 0 & \text{else} \end{cases} \tag{16}$$

makes the condition $\mathbf{D}\tilde{\mathbf{d}} = 0$ to hold.

A.2 Distance Calculation

Given the quadratic surface $\mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{a}^T\mathbf{x} + a_0 = 0$ in normal form, i.e., $\mathbf{A}$ is diagonal and $\mathbf{Aa} = \mathbf{0}$, then the distance of a query point $\mathbf{p}$ to the surface is calculated [11] by finding the roots of the 6th-order polynomial

$$\begin{aligned} f(\alpha) = \; & A_1 p_{\alpha 1}^2 d_2^2 d_3^2 + A_2 p_{\alpha 2}^2 d_1^2 d_3^2 + A_3 p_{\alpha 3}^2 d_1^2 d_2^2 + \\ & 2(a_1 p_{\alpha 1} d_1 d_2^2 d_3^2 + a_2 p_{\alpha 2} d_1^2 d_2 d_3^2 + a_3 p_{\alpha 3} d_1^2 d_2^2 d_3) + a_0 d_1^2 d_2^2 d_3^2 \end{aligned} \tag{17}$$

with $\mathbf{p}_\alpha = \mathbf{p} - \alpha\mathbf{a}$ and $d_i = 1 + \alpha A_i$ (A_i is the ith diagonal element of $\mathbf{A}$).
Then,

$$\mathbf{x}_\alpha = (\mathbf{I} + \alpha\mathbf{A})^{-1}(\mathbf{p} - \alpha\mathbf{a}) \tag{18}$$

gives the intersection points of the surface and the lines standing perpendicular on the surface and passing through $\mathbf{p}$. Finally, out of the maximum 6 real-valued solutions,

$$d_p = \min_\alpha ||\mathbf{x}_\alpha - \mathbf{p}|| \tag{19}$$

gives the perpendicular distance between the quadratic surface and the query point $\mathbf{p}$.

B Derivation of the Decision Boundary Between Two Classes

The adaptive color space is based on the distance of a color sample $\mathbf{x}$ to the quadratic decision boundary which is formed between the two normally distributed classes. For convenience, we quote the derivation of the decision boundary equation given in [7]:
According to Bayes' Theorem a sample $\mathbf{x}$ should be assigned the class k which maximizes

$$P(k \mid \mathbf{x}) = \frac{P_k \, p(\mathbf{x} \mid \Theta_k)}{\sum_{j=1}^{K} P_j \, p(\mathbf{x} \mid \Theta_j)}, \tag{20}$$

given the class a priori probability P_k and the class parameters Θ_k.
Since the denominator is equal for all classes, it does not have to be calculated explicitely. It is enough to choose the class which maximizes

$$P_k \, p(\mathbf{x} \mid \Theta_k) = \frac{P_k}{\sqrt{(2\pi)^d \, |\mathbf{\Sigma}_k|}} \, \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)} \tag{21}$$

or which minimizes[3]

$$D_k^2 = -2\ln(P_k) + \ln(|\mathbf{\Sigma}_k|) + (\mathbf{x}-\boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k), \tag{22}$$

which actually represents a generalized distance of a data sample to the "distribution cloud".
To derive the equation of the decision boundary between two classes i and j, one starts with the fact that on this decision boundary the generalized distances D_i^2 and D_j^2 are equal.

$$\begin{aligned} &-2\ln(P_i) + \ln(|\mathbf{\Sigma}_i|) + (\mathbf{x}-\boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i) \\ &= -2\ln(P_j) + \ln(|\mathbf{\Sigma}_j|) + (\mathbf{x}-\boldsymbol{\mu}_j)^T \mathbf{\Sigma}_j^{-1} (\mathbf{x}-\boldsymbol{\mu}_j) \end{aligned} \tag{23}$$

Since covariance matrices are always symmetric, $(\mathbf{x}-\boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i)$ can be written as

$$\begin{aligned} (\mathbf{x}-\boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{x}-\boldsymbol{\mu}_i) &= \mathbf{x}^T \mathbf{\Sigma}_i^{-1} \mathbf{x} - \mathbf{x}^T \mathbf{\Sigma}_i^{-1} \boldsymbol{\mu}_i - \boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{x} + \boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \boldsymbol{\mu}_i \\ &= \mathbf{x}^T \mathbf{\Sigma}_i^{-1} \mathbf{x} - 2\boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{x} + \boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \boldsymbol{\mu}_i . \end{aligned} \tag{24}$$

Collecting the constants of (23) gives

$$a_0 = -2\ln(P_j) + 2\ln(P_i) + \ln(|\mathbf{\Sigma}_j|) - \ln(|\mathbf{\Sigma}_i|) + \boldsymbol{\mu}_j^T \mathbf{\Sigma}_j^{-1} \boldsymbol{\mu}_j - \boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \boldsymbol{\mu}_i \tag{25}$$

which further simplifies (23) to

$$\mathbf{x}^T \mathbf{\Sigma}_i^{-1} \mathbf{x} - 2\boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{x} = \mathbf{x}^T \mathbf{\Sigma}_j^{-1} \mathbf{x} - 2\boldsymbol{\mu}_j^T \mathbf{\Sigma}_j^{-1} \mathbf{x} + a_0 \tag{26}$$

$$0 = \mathbf{x}^T \mathbf{\Sigma}_j^{-1} \mathbf{x} - \mathbf{x}^T \mathbf{\Sigma}_i^{-1} \mathbf{x} - 2\boldsymbol{\mu}_j^T \mathbf{\Sigma}_j^{-1} \mathbf{x} + 2\boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{x} + a_0 \tag{27}$$

$$0 = \mathbf{x}^T \underbrace{\left(\mathbf{\Sigma}_j^{-1} - \mathbf{\Sigma}_i^{-1} \right)}_{\mathbf{A}} \mathbf{x} + 2 \underbrace{\left(\boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} - \boldsymbol{\mu}_j^T \mathbf{\Sigma}_j^{-1} \right)}_{\mathbf{a}^T} \mathbf{x} + a_0 . \tag{28}$$

Finally, the decision boundary turns out to be a quadratic surface

$$\mathbf{x}^T \mathbf{A} \mathbf{x} + 2\mathbf{a}^T \mathbf{x} + a_0 = 0 \tag{29}$$

with the parameters

$$\mathbf{A} = \mathbf{\Sigma}_j^{-1} - \mathbf{\Sigma}_i^{-1} \tag{30}$$

$$\mathbf{a} = \mathbf{\Sigma}_i^{-1} \boldsymbol{\mu}_i - \mathbf{\Sigma}_j^{-1} \boldsymbol{\mu}_j \tag{31}$$

$$a_0 = -2\ln(P_j) + 2\ln(P_i) + \ln(|\mathbf{\Sigma}_j|) - \ln(|\mathbf{\Sigma}_i|) + \boldsymbol{\mu}_j^T \mathbf{\Sigma}_j^{-1} \boldsymbol{\mu}_j - \boldsymbol{\mu}_i^T \mathbf{\Sigma}_i^{-1} \boldsymbol{\mu}_i . \tag{32}$$

[3] To come from (21) to (22) take the natural logarithm, multiply with -2 and omit $d \ln(2\pi)$.

References

[1] Gerald Friedland, Kristian Jantz, and Raul Rojas. SIOX: Simple interactive object extraction in still images. In *IEEE International Symposium on Multimedia (ISM2005)*, Irvine (California), December 2005.

[2] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut — interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (SIGGRAPH'04)*, August 2004.

[3] Alexander Ihlow and Udo Seiffert. Automating microscope colour image analysis using the Expectation Maximisation algorithm. In Carl Edward Rasmussen, Heinrich H. Bülthoff, Martin A. Giese, and Bernhard Schölkopf, editors, *Pattern Recognition: 26th DAGM Symposium*, volume LNCS 3175 of *Lecture Notes in Computer Science*, pages 536–543, Tübingen, Germany, August/September 2004. Springer.

[4] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[5] Richard A. Redner and Homer F. Walker. Mixture densities, maximum likelihood, and the EM algorithm. *SIAM Review*, 26:195–239, 1984.

[6] Jeff A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997.

[7] Earl Gose, Richard Johnsonbaugh, and Steve Jost. *Pattern Recognition and Image Analysis.* Prentice Hall, Upper Saddle River, NJ, 1996.

[8] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, New York, 2001.

[9] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition.* Academic Press, Amsterdam, 2nd edition, 2003.

[10] Todd Rowland. Companion matrix. From MathWorld – A Wolfram Web Resource, created by Eric W. Weisstein. `http://mathworld.wolfram.com/CompanionMatrix.html`.

[11] Christian Lennerz and Elmar Schömer. Efficient distance computation for quadratic curves and surfaces. In *2nd IEEE Conference on Geometric Modeling and Processing (GMP 2002)*, pages 60–69, Wako, Saitama, Japan, 2002.

[12] Christian Lennerz. *Distance Computation for Extended Quadratic Complexes.* PhD thesis, Universität des Saarlandes, December 2005.

Figure 1: A greenish-blue dyed plant cell surrounded by discolored cell tissue.*

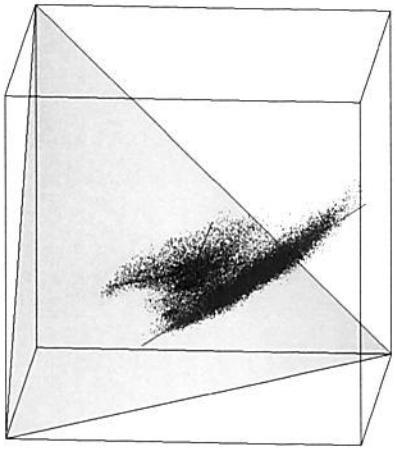

Figure 2: Class clusters in RGB.†
Blue: *"background"*, red: *"object of interest"*.

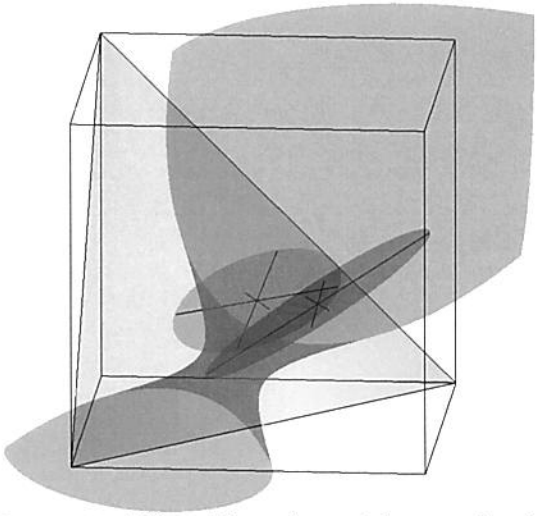

Figure 3: Ellipsoids and resulting quadratic decision boundary (one-sheet hyberboloid).†

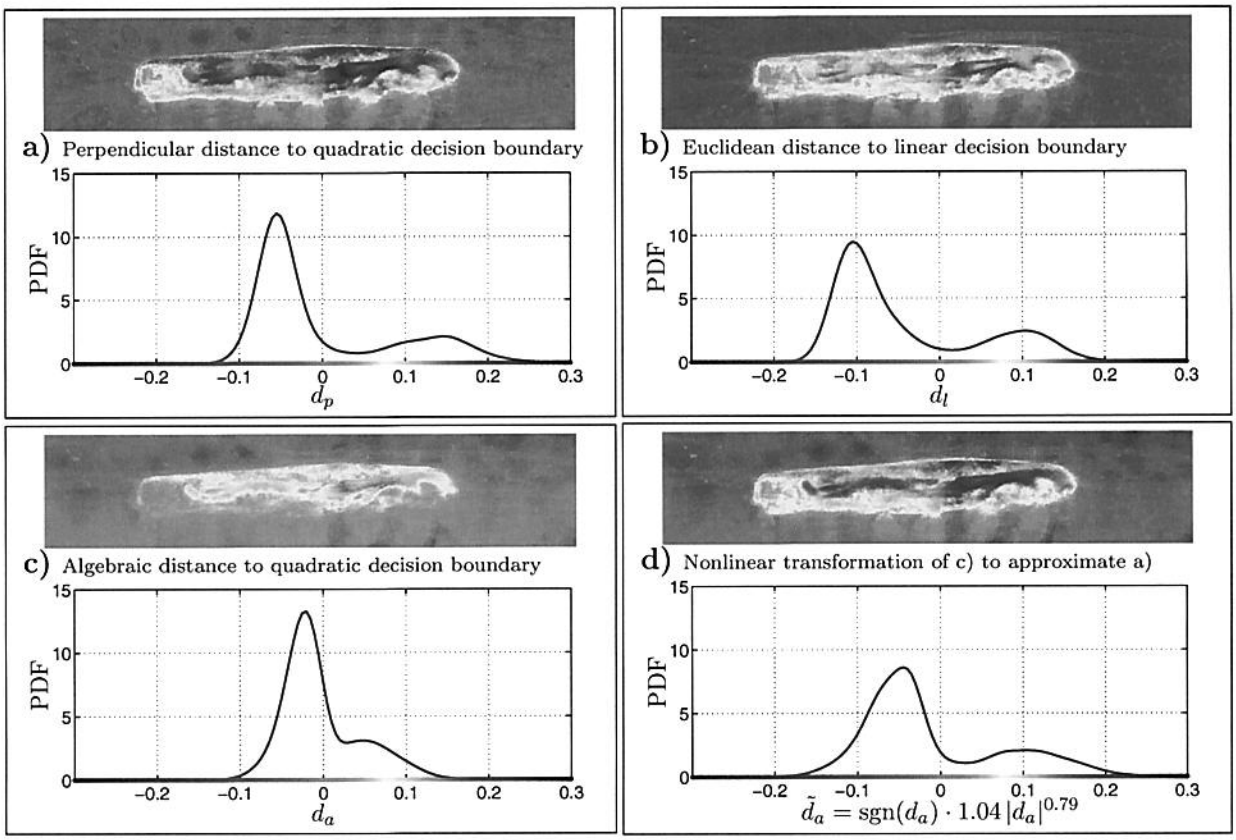

Figure 4: Results as pseudocolor representation and probability distribution.

*This image was provided by Patrick Schweizer from the Transcriptome Analysis Group at the Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben.

†The colored plane (Maxwell triangle) is drawn for illustrating the attribution of the RGB cube's vertices.

Figure 5: Dark hyphae growing on a plant leaf surrounded by discolored cell tissue.*

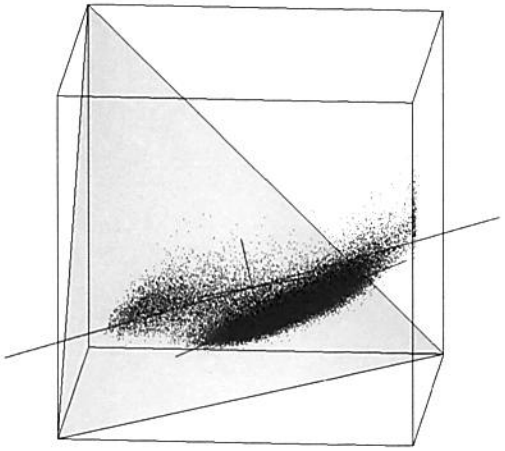

Figure 6: Class clusters in RGB.†
Blue: *"background"*, red: *"object of interest"*.

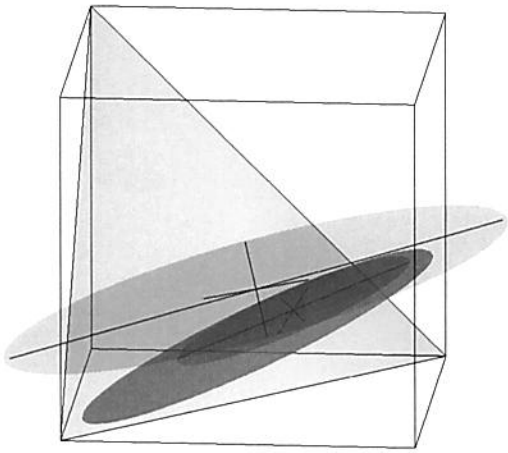

Figure 7: Ellipsoids and resulting quadratic decision boundary (ellipsoid).†

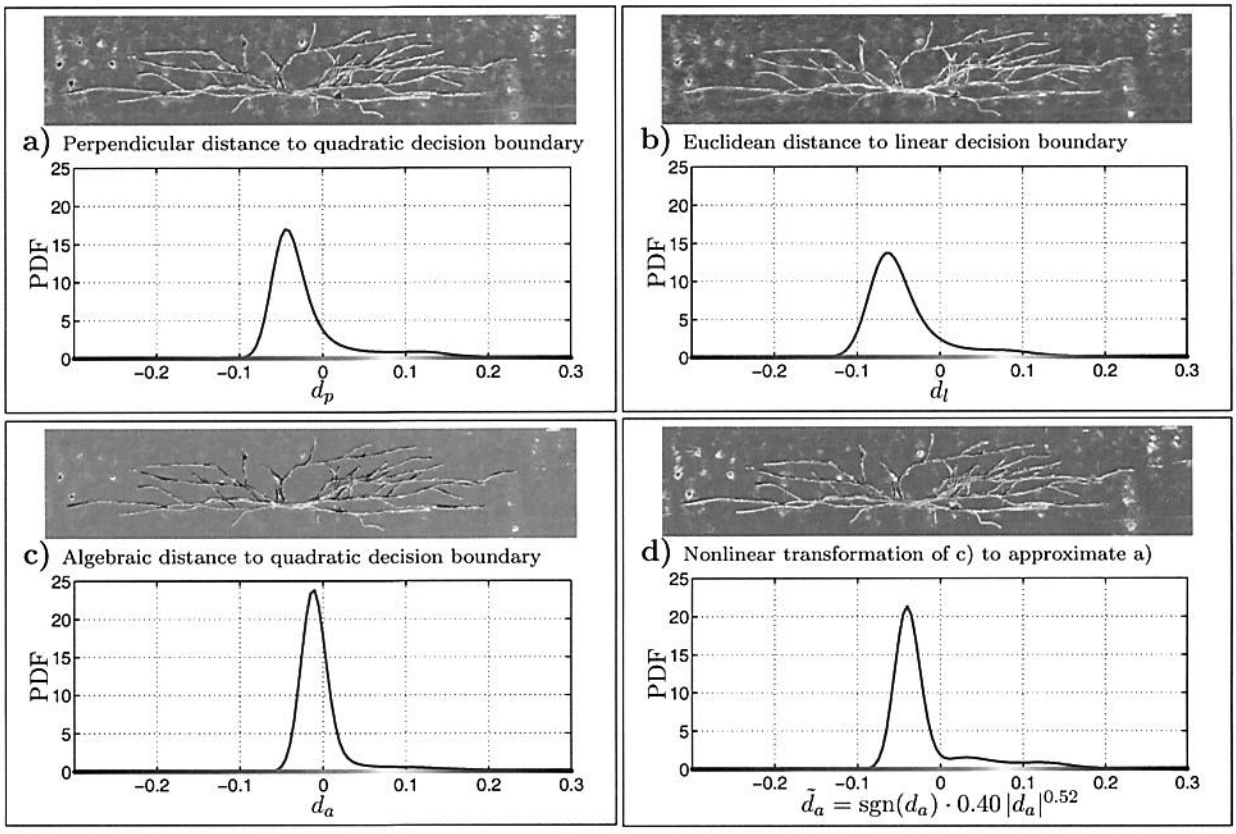

Figure 8: Results as pseudocolor representation and probability distribution.

*This image was provided by Patrick Schweizer from the Transcriptome Analysis Group at the Leibniz Institute of Plant Genetics and Crop Plant Research (IPK), Gatersleben.

†The colored plane (Maxwell triangle) is drawn for illustrating the attribution of the RGB cube's vertices.

Figure 9: Dyed hyphae growing on a plant leaf surrounded by discolored cell tissue.**

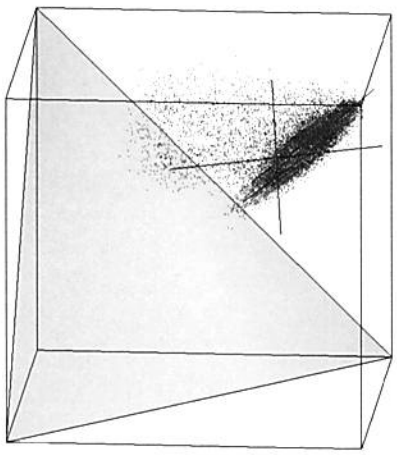

Figure 10: Class clusters in RGB.†
Blue: *"background"*, red: *"object of interest"*.

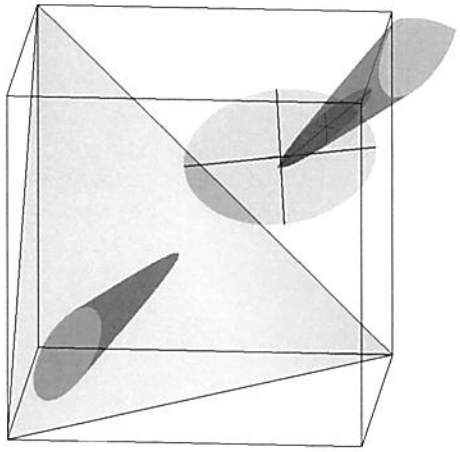

Figure 11: Ellipsoids and resulting quadratic decision boundary (two-sheet hyberboloid).†

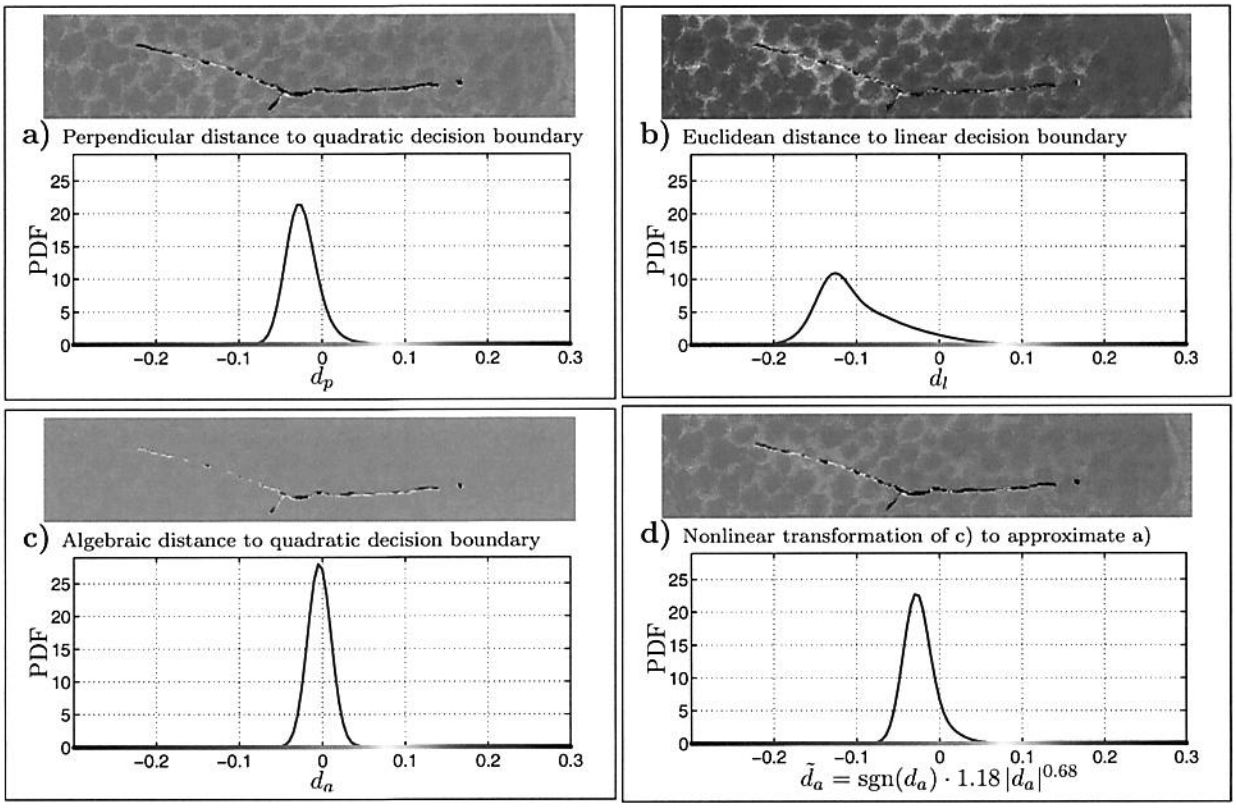

Figure 12: Results as pseudocolor representation and probability distribution.

**This image was provided by Katharina Goellner from the Plant Microbe Interactions Group at the Max Planck Institute for Plant Breeding Research, Cologne.

†The colored plane (Maxwell triangle) is drawn for illustrating the attribution of the RGB cube's vertices.