3-Way Modeling of Maize Kernels

using hyperspectral image analysis

by Morten Arngren & Christian Schack Pedersen

on January 29, 2009

Contents

1 Introduction

Near-infrared (NIR) as a tool for investigation of maize kernels has been studied extensively (Orman and Schuman, 1991 [9]; Baye et al., 2006 [2]; Jiang et al. 2007 [8] and others) with standard optical fiber technology both in reflectance and transmission mode. This has recently been expanded to include hyper-spectral imaging, which combine spatial and spectral information. NIR image analysis therefore has the possibility to relate the chemical information directly to its location.

Two recent articles have been published on the application of hyper-spectral NIR imaging on maize kernels (Cogdill et al. 2004 [4] and Weinstock et. al. 2006 [12]). Both articles on maize, and those in the review on hyperspectral imaging in food science[1], tries to predict various properties using 2-way methods such as PCA, PLS, PCR and MLR. None of the article utilizes any N-way models although some apply artificial neural networks (Table 2 in Gowen et al. 2007 [6]) for non-linear modeling.

2-way chemometric methods requires 2D data matrices. This poses a problem when using hyper-spectral NIR data as the raw-data have an underlying 4D structure: $X \times Y \times Spectra \times Kernels$. Often the pictures contain several kernels that are then segmented to obtain this 4D structure. The first dimension reduction is, in both articles and our work, to unfold the pixels dimensions into one, which yields a 3-D structure: $XY \times Spectra \times Kernels$.

The articles [4] [12] then reduce the 3D structure by taking the average of all kernel pixels, within each kernel, to obtain one average NIR spectrum for each kernel leading to the 2D matrix: $Spectra \times Kernels$. The drawback of this method is that a significant amount of spatial information is neglected, which partly removes the reason for using hyper-spectral imaging. The approach is refined a bit by Weinstock et al. (2006) [12], which first isolate the germ (because they investigate oil content) using various segmenting strategies. Both articles experiment with selecting specific wavelengths to reduce the amount of non-relevant information. Ideally there would be no information loss when applying N-way modeling.

This report is a proof-of-concept study to access the feasibility of using an N-way models to describe hyper-spectral NIR data of maize kernel. The data set are rather small and we have no further information about the maize kernels that could be used to make an N-PLS model or similar predictive model. If a 3-way model proves worthwhile, future studies could focus on expanding the work into testing 3-way predictive models. This would make it possible to estimate several parameters at once and relating those to a spatial distribution within the kernel.

1.1 Maize Kernels

Many cereals share the same compounds and basic structure. In our case of maize a single kernel can be divided into many different constituents on the macroscopic level as illustrated in figure 1.1.

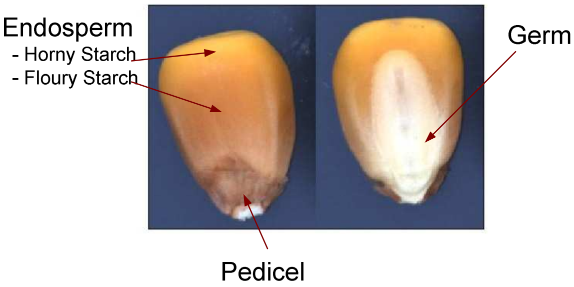

Figure 1.1: Structure of a maize kernel.

[1] For a review of hyper-spectral NIR imaging in food science refer to Gowen et al. (2007) [6].

The structural components of maize kernels can be divided into 4 classes denoted *Endosperm*, *Germ*, *Pericarp* and *Pedicel*. These components has different functions and composition as listed in table 1.1 leading to different spectral profiles as described below [2].

Endosperm

The endosperm is the main storage for starch, protein and water in cereals. Starch being the main constituent is a carbohydrate and consist of two different glucans named *Amylose* and *Amylopectin*.

The starch in maize grains is stored in granules, as a protein-carbohydrate complex. These complexes can be further divided into a floury and horny type depending on the composition of the protein matrix and the resulting superstructure of the complex. These two types of starch are typically mutually exclusive, but in maize grain they both appear as a special case as also illustrated in figure 1.1. Typical unprocessed IR spectra is seen in figure 1.2 as spectra 1 and 2.

Germ

The germ of a cereal is the reproductive part that germinates to grow into a plant. It is the embryo of the seed, where the scutellum serves to absorb nutrients from the endosperm during germination.

The germ has high oil content compared to the endosperm and has smaller concentrations of protein, sugars, lipids vitamins and minerals [3]. Typical unprocessed IR spectra is seen in figure 1.2 as spectra 3.

Pericarp & Pedicel

The pericarp is the skin or bran of the maize kernel and the pedicel is the flower stalk. Both primarily consist of complex carbohydrates and proteins. Typical unprocessed IR spectra is seen in figure 1.2 as spectra 4.

Chemical Com.	Pericarp	Endosperm	Germ
Protein	3.7%	8 .0%	18.4%
Oil	1.0%	0.8%	33.2%
Crude Fiber	86.7%	2.7%	8.8%
Ash	0.8%	0.3%	10.5%
Starch	7.3%	87.6%	8.3%
Sugar	0.34%	0.62%	10.8%

Table 1.1: Constituents of maize kernels [11].

NIR Spectral Analysis

NIR spectra of complex mixtures, such as a maize kernel is expected to have a few or no distinctive peaks due to the spectral overlap of the different components. By using hyper-spectral imaging we see that the NIR spectra of the maize kernel is distinctively different across the kernel (figure 1.2). This is a significant improvement compared to conventional NIR, which would just show the grand mean.

The spectra of the germ have a large absorption peak at appr. 1700 nm, which correspond to a strong oil absorption (refer to appendix A). This is expectable as the germ have a high concentration of oil (table 1.1). It is surprising that there is not an distinct absorption at 1390 nm.

The reflectance spectra of the endosperm, show absorption at 1200 nm (protein) and a strong continuous absorption between 1400 nm and 1650 nm, which correspond to several strong proteins absorptions. It could also include the starch absorption at 1700 nm that would have been shifted due to different binding environments. The other strong absorption of starch is at the limit or outside the spectral range used in this analysis.

[2] For a more detailed description of the general structure of cereals refer to Hoseney (1994) [7].

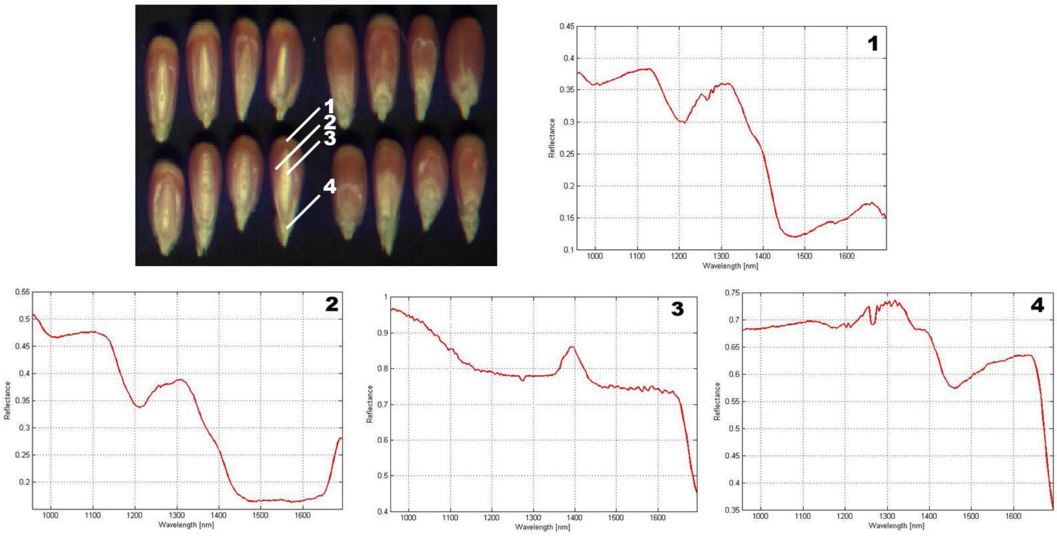

Figure 1.2: Typical unprocessed reflectance spectra for the different constituents.

2 Dataset

In our analysis a hyperspectral line-scan NIR camera sensitive from 900-1700nm was used to capture the image data of 8 maize kernels. The InGaAs sensor has a resolution of 320 spatial pixels and 165 spectral pixels and suffers from poor sensitivity in the initial bands from 900-950nm. As these bands are very noisy and contain negligible information, this small spectral range is omitted from our further analysis and the number of bands are reduced to 153 bands.

Images of both the germ and endosperm side of the kernels are acquired on a black background (NCS-9000) and appended into a single image as shown top left in figure 1.2. This means the final hyperspectral image $\mathcal{X}$ will have dimensions of $345 \times 290 \times 153$ bands. The kernels are furthermore not fresh from harvest and hence have a very low water content and are in addition free from any infections as well.

A set of small image processing steps must be conducted in order to obtain usable image data regardless of the subsequent model. These few steps are described below.

Image Correction

The raw acquired NIR intensities are highly influenced by the NIR light source and sensor background noise and requires correction to extract the actual spectral responses. Hence the entire image is divided by an acquired white reference (NCS-0300) image and subtracted from the dark current background image.

Y-Axis Averaging

As part of the image acquisition the distance between line scans can be specified in order to achieve square pixels. In our case this was set to twice the normal value for the purpose of noise reduction. This means every two Y-axis lines are averaged into one and the height of the hyperspectral image are hence reduced from 290 to 145 pixels.

Convert to Absorbance

The absorption of the individual constituents is linearly dependent on the amount of the constituent (Beer's law), which means that we can determine the relative spatial distribution of the constituents within the kernels. For Beer's law to apply the intensities are converted to absorbances by $-\log_{10}(\mathbf{x}) \rightarrow \mathbf{x}$.

3 Pre-Processing

Prior to the N-way modeling the hyperspectral data is subjected to a series of pre-processing steps in order to optimize the subsequent N-way modeling. The structure and order of the steps are illustrated in figure 3.1. In this section each of the most important steps are described in detail.

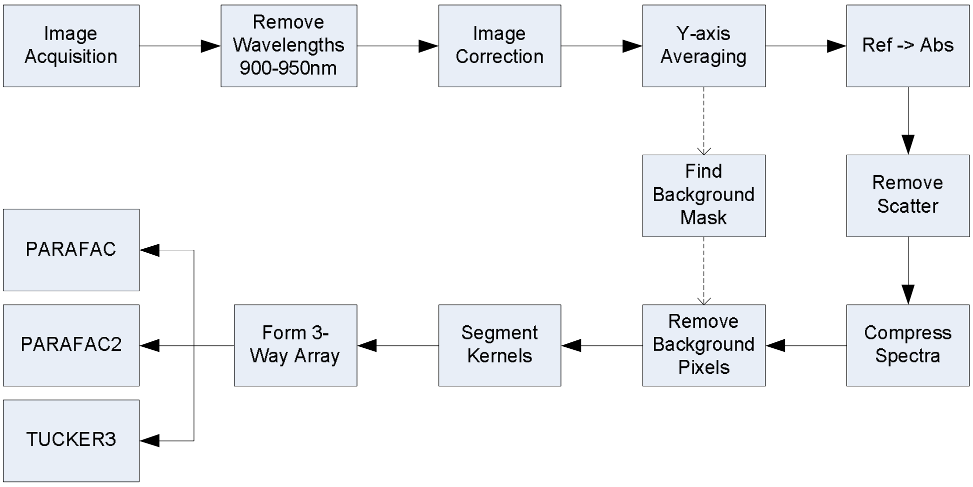

Figure 3.1: Flowchart of the individual pre-processing steps.

3.1 Scatter Correction

In a hyperspectral image acquisition system the acquired spectra will suffer from scattering due to the shape and the texture of the maize kernels. The raw spectra are dominated by this scattering, which masks the spectral information as shown in figure 3.2.

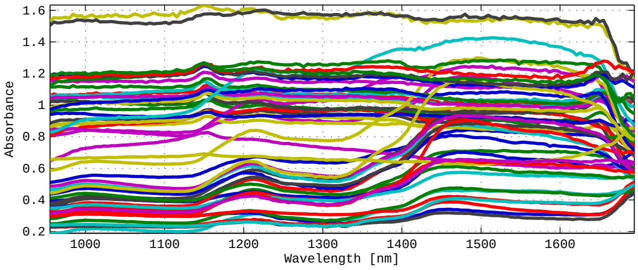

Figure 3.2: Scatter of the unprocessed absorbance spectra.

To solve this problem we have experimented with two different preprocessing steps: *Multiplicative Scatter Correction* (MSC) and *Savitzky-Golay* smoothing (SG). The algorithms supplied by the PLS-Toolbox were applied for all preprocessing operations.

Multiplicative Scatter Correction, adjust section

Multiplicative scatter correction corrects multiplicative and non-constant (linear) additive baseline effects in a single step. MSC offsets and adjusts each spectrum according to a linear fit of a reference spectrum (Geladi et al. 1985 [5]).

The reference spectrum is by default the mean spectrum of the whole picture, but can be selected as the median of a specified data set or even a selected reference spectrum. We tried several approaches to the selecting the MSC reference spectrum, but without success. This was partly due to the inhomogeneity of the maize kernels, both physically and with respect to orientation, and due to the presence of pixels being partly kernel and partly background.

The MSC correction is attractive as the original spectrum is retained, which eases chemical interpretation significantly and MSC is still a subject for further investigation.

Savitzky-Golay Smoothing

The Savitzky-Golay smoothing (SG) fits a polynomial (order) through a number of datapoints (width) and calculates the derivative (deriv) of the polynomial in the point. The function then proceeds in a similar fashion as a boxcar average. We tested several different settings and settled on

- Width = 7
- Order = 2
- Derivative = 1

The width was set low to minimize smoothing of the peaks. Both second and third order polynomial gave satisfying models. The second order polynomium was chosen to avoid overfitting, due to the low window width and the high redundancy of the spectra. The first order derivative resulted in good models and have also been applied previously to similar problems (Weinstock et al. 2006 [12]; Jiang et al. 2006 [8]; Baye et al. 2006 [2]).

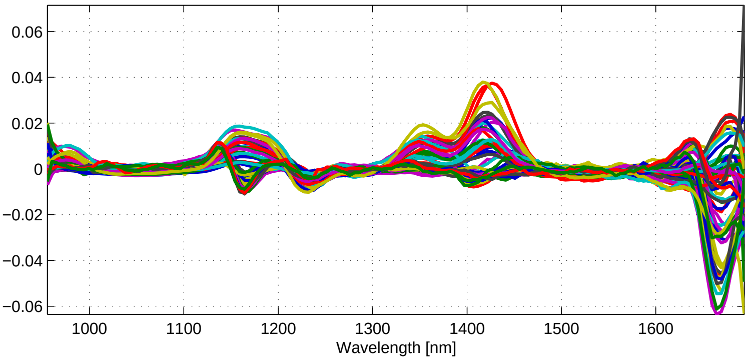

Figure 3.3: Scatter of the unprocessed absorbance spectra.

As we use the 1st order derivative in the Savitzky-Golay smoothing the original absorbance peaks will shift in the spectra and hence lead to a different chemical interpretation.

The original acquired spectra are highly non-negative due to the nature of the intensities. The SG scatter correction used here introduces negative components in the spectra and hence further leads to difficult chemical interpretation.

3.2 Compression of Spectra

The acquired spectra consist of 153 bands and has a high level of redundancy as shown in figure 3.3. This suggest the spectral information can be reduced from 153 dimensions without loss of important information.

The spectral information can be compressed by projecting the data onto directions spanning the highest variance. These orthonormal vectors $\mathbf{U} = \{\mathbf{u}_z\}_{z=1}^{Z}$ are determined as the principal components found by PCA [1].

Prior to the eigenvalue decomposition the spatial information is omitted and the 3-way array is unfolded to a matrix with the spectra as column vectors, i.e. $\mathbf{X} = \{\mathbf{x}_z\}_{z=1}^{Z}$, where $Z = 153$ bands. This means the compressed spectra can be expressed as

$$\tilde{\mathbf{x}} = \tilde{\mathbf{U}}^T(\mathbf{x} - \mu) \tag{3.1}$$

where $\tilde{\mathbf{U}}$ denotes the reduced rank matrix of eigenvectors and μ is the mean spectra. The compressed spectra are subsequently folded back to a 3-way array with fewer dimensions. The dimension reduction further means the compressed spectra will include negative elements and will lead to decorrelated data due to orthonormal projection.

For our hyperspectral image data subjected to a PCA the extracted eigenvectors (principal spectral components) are illustrated in figure 3.4 (only the first 6 first shown). The corresponding projected spectra (scores) according to (3.1) are depicted in figure 3.5.

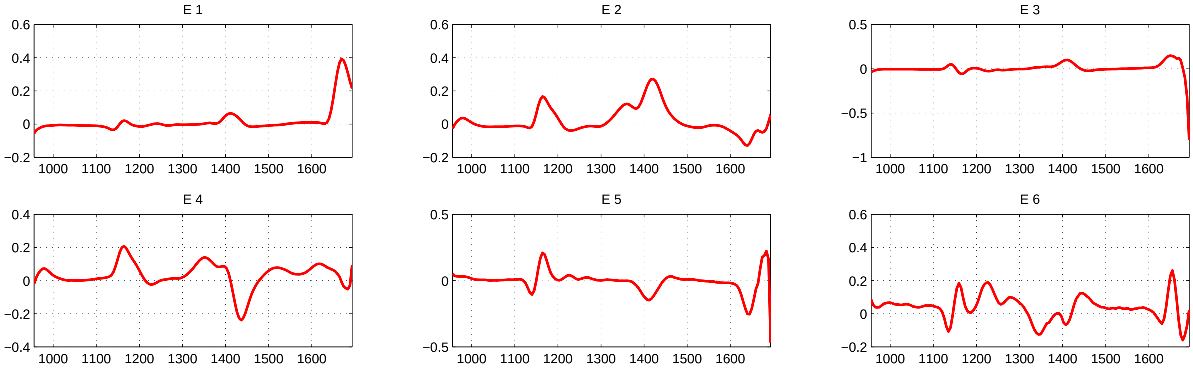

Figure 3.4: The 6 first principal spectral components (eigenvectors or loadings) with the highest variance.

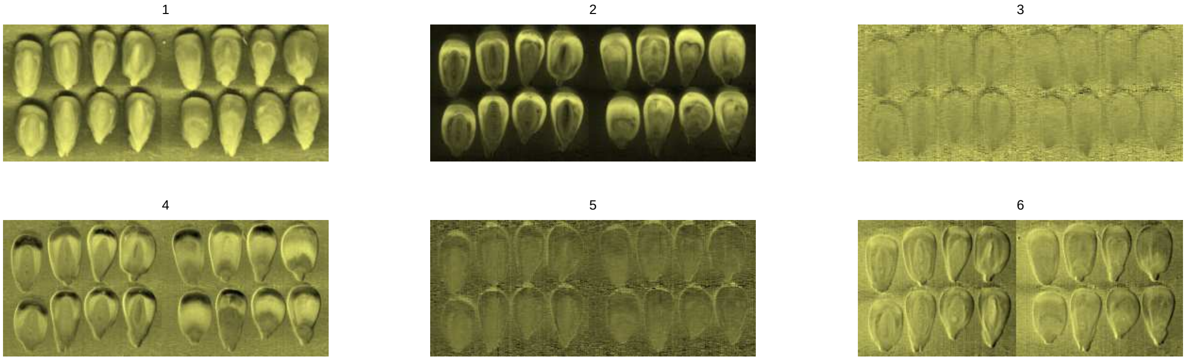

Figure 3.5: Score image of first 6 principal components.

From figure 3.5 it is clear to see how component 1, 2 and 4 holds important internal structure of the kernels, whereas the remaining components include more noise. The corresponding spectral loadings for these components are hence expected to contain most of the chemical information.

The 3rd components in figure 3.4 appear to consist primarily of spectral noise at appr. 1700nm located in the background, which is removed during the background segmentation. The contribution of the 3rd component is thus expected to diminish in the following pre-processing steps.

This simple PCA model is able to describe 3 different spectral components and how these are spatially located. This is a satisfying explorative model and we will use this as a reference when evaluating the performance of the different 3-way models. The 2nd and 4th spectral component are similar, which indicate that they describe related chemical compounds. Both components are associated with the endosperm, but describe two different parts of the endosperm. This could be interpreted as being the two different kind of starch, the floury (PC 4) and the horny (PC 2). The first spectral component are associated with the germ and are significantly different from the 2nd and 4th spectral component. This component could be interpreted as describing the oil content of the germ.

For our dataset of kernels a reduction to 10 principal components has in practice proven sufficient in describing the spectral variance. The compressed spectra are illustrated right in figure 3.6.

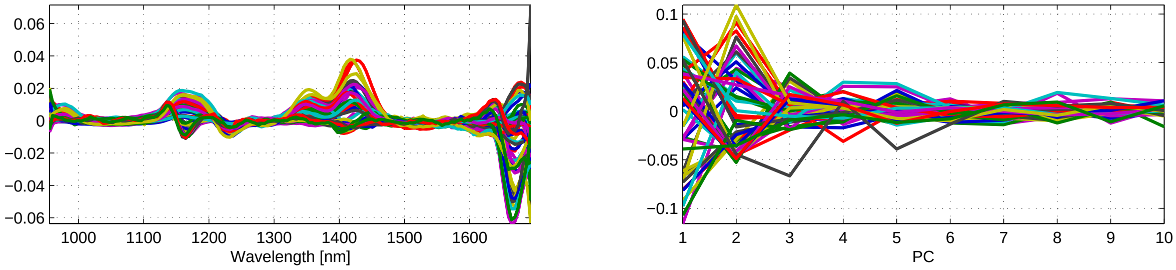

Figure 3.6: A subset of the spectra before and after compression, left and right respectively.

3.3 Remove Background Pixels

The background pixels has no relevance for our analysis of the maize kernels and are to be removed. In order to classify each pixel as background or not the 1st principal component of the reflectance image data (left in figure 3.7) is used to form a segmentation mask. With a simple positive/negative discriminant each pixel is easily classified to form the mask (right in figure 3.7).

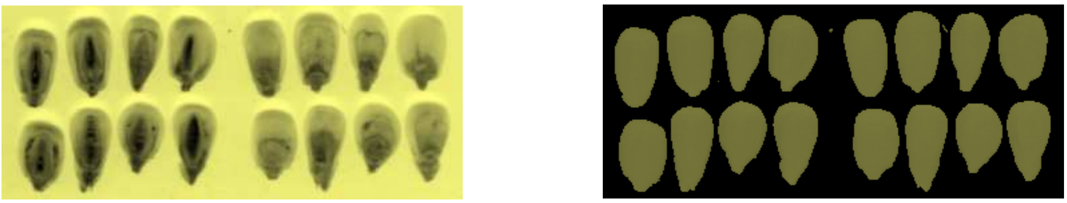

Figure 3.7: Left: 1st principal component (scores plot). *Right:* The segmentation mask.

A few distinct background pixels are misclassified as foreground and will be removed by hand during kernel segmentation. The saved segmentation mask is then applied subsequent to the spectral compression to remove the background pixels.

3.4 Segmentation of Individual Kernels

The individual kernels are segmented out by manually selecting a surrounding square corresponding to the largest kernel of size 37×65 pixels (refer to figure 3.7 right illustration). Using these box coordinates each of the kernels are extracted and numbered. The segmentation of each kernel is subsequently inspected and any overlapping parts of other kernels is manually removed by setting the value of those pixels to NaN.

The kernels are further divided into an germ set and endosperm set as illustrated left and right in figure 3.8 and are treated as two separate datasets.

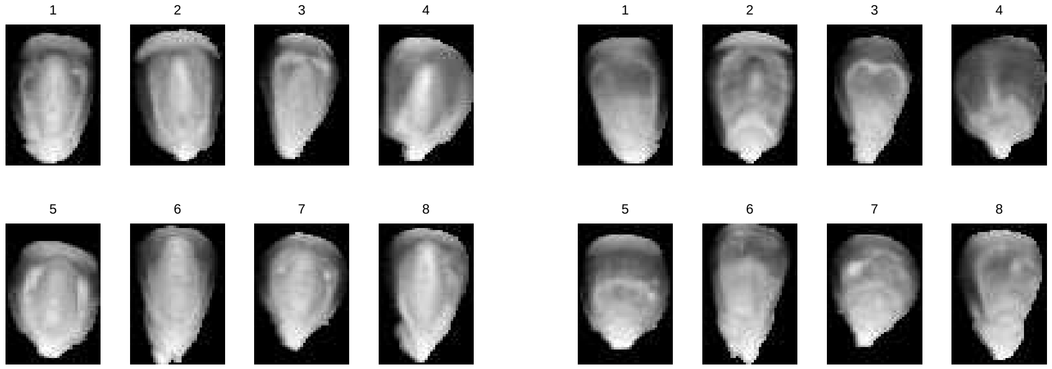

Figure 3.8: Segmented kernels for the germ side (*left*) and the endosperm side (*right*). Images are scaled individually for high contrast.

3.5 Form 3-Way Array

Each of the maize kernels are in their initial format each a 3-way array as described and including all kernels would form a 4-way array, i.e. $Kernel \times Spectra \times X \times Y$. As two of the modes in such a 4-way array represent spatial information we collapse these modes into a common XY mode and thus achieve a 3-way array on the form $Kernel \times Spectra \times XY$. This leads to a vector representation of the 2D image data, where the different regions of the kernels of different size might appear at shifted locations.

The resulting 3-way array will contain slabs with varying amounts of NaN elements, from none to all, due to the previous background segmentation. To remove these slabs with NaN and thus reduce the amount of empty information (and further to please PARAFAC2 and TUCKER3) we impose a threshold. The threshold is the percentage of active pixels and the resulting masks for different thresholds are shown in left in figure 3.9.

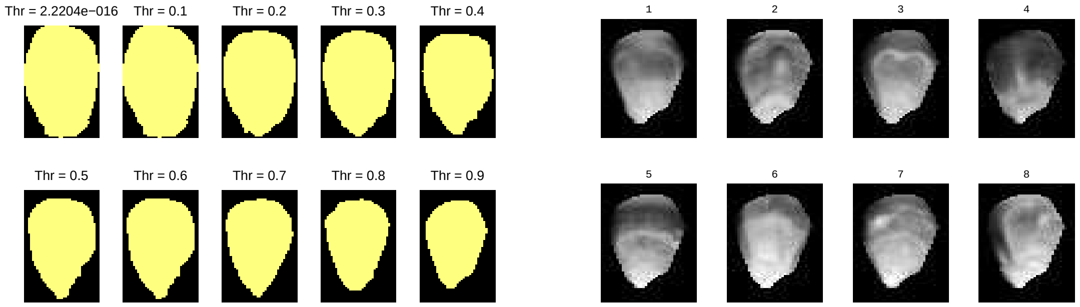

Figure 3.9: Left: Kernels masks for different thresholds. *Right:* Cropped kernel for a threshold of 0.9 (scaled for high contrast).

Setting the thresholds to zero leads to a mask where the structure of all kernels are kept, but includes slabs where not all kernels are present. In contrast a threshold of 1 extracts only the common pixels for all kernels and ensures no NaN values are in the dataset.

In selecting the optimal threshold it is important to avoid any 3-way model in capturing the non-common edges of the kernels for a low threshold. For this reason we have therefore chosen a threshold of 0.9 and thus continue with only common pixels, since we only have 8 kernels.

4 3-Way Modeling

Classical analysis of hyperspectral data is conducted considering only spectral or spatial data separately and many bold attempts has been made to combine them afterwards. Multiway analysis offers the opportunity to model the dependency across all modes simultaneously.

In this section the hyperspectral data will finally be subjected to three different 3-way models, i.e. *PARAFAC*, *PARAFAC2* and *TUCKER3*. The analysis will be quantitative and not describe any details in the underlying theory behind the models (refer to [10]). In addition the modeling will be unsupervised, as we have no labels associated with the dataset.

4.1 PARAFAC

The simplest 3-Way model is the linear PARAFAC model (PARAllel FACtors), where each of the elements in an IJK dimensional 3-Way array can be expressed by 3 loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as

$$x_{ijk} = \sum_{n=1}^{N} a_{in} b_{jn} c_{kn} + \epsilon_{ijk} \tag{4.1}$$

where $\mathbf{A} = \{a_{in}\}_{i=1,n=1}^{I,N}$, $\mathbf{B} = \{b_{jn}\}_{j=1,n=1}^{J,N}$, $\mathbf{C} = \{c_{kn}\}_{k=1,n=1}^{K,N}$ and N are the number of components. The residual noise are contained in the 3-way array ϵ. The model can also be visualized as illustrated in figure 4.1.

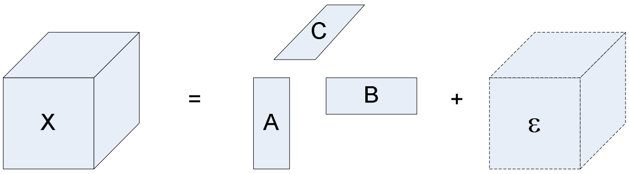

Figure 4.1: A 3-way PARAFAC model.

The optimization algorithm for estimating the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ loading matrices is based on the Alternate-Least-Squares (ALS) approach and a detailed description of this method is out of the scope of this report, refer to [10] for details.

As part of the parameters for the PARAFAC model a set of constraints can be specified for each of the loadings separately. For our 3-Way data array the 1st mode represents the kernel scores and must be non-negative in order to interpret them as concentration coefficients. The 2nd mode holds the spectral information, which after the SG scatter removal and PCA compression are no longer non-negative. Hence a non-negative constraint is this mode would not seem correct as it will limit the solution space to the positive orthant. The XY data being non-negative by nature is represented in the 3rd mode and suggest a non-negative constraint is feasible.

In summary the theoretical optimal set of constraints is $[101]$. However enabling a constraint does not necessarily mean the constraint will be active depending on the dataset. We therefore apply all combinations of constraints to the PARAFAC model, i.e $[000, 001, 010, 011, 100, 101, 110, 111]$.

We expect to capture 2-6 components describing the different constituents in the kernels (section 1) and thus subject the 3-Way data of endosperm side of the kernels to the PARAFAC model for $[2-6]$ components and all 8 combinations of the non-negative constraint. The stopping criteria was further set to allow for long simulation times and iterations in order to obtain reliable results. Each of the parameters sets are run twice to verify a unique solution.

The obtained PARAFAC models are analyzed by primarily evaluating the spatial distribution and the spectral differences in the loading matrices $\mathbf{B}$ and $\mathbf{C}$. Evaluating the core consistency is not relevant for this type of data, since the spectral and spatial mode are not highly dependent as for other types of dataset.

The most interesting and interpretable models achieved were the 3 and 4 component models with constraints $[101, 111]$, i.e. spectra unconstrained and full non-negativity constraint on all modes. The spatial distribution (loading 2) of the 4 different model are illustrated in figure 4.2. These basis kernel images depict the loading intensity for each kernel and are scaled for optimal contrast. The kernel images can therefore not be compared directly.

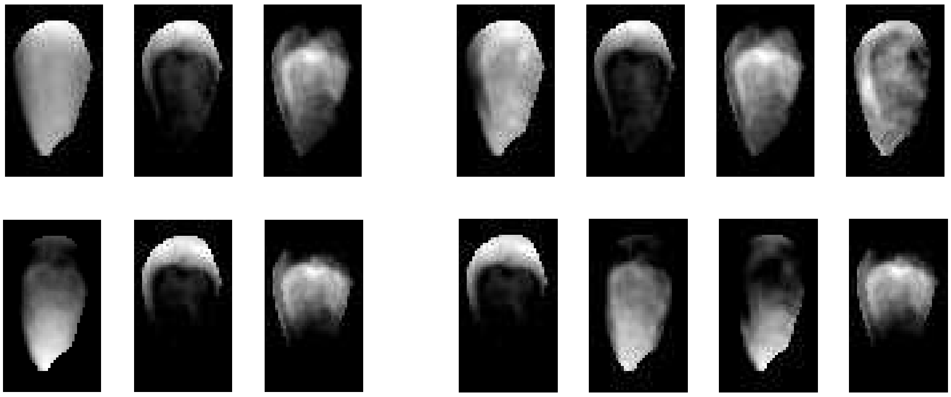

Figure 4.2: Top left: The 3 spatial loadings for con. $[101]$ (89.6%). *Top right:* The 4 spatial loadings for con. $[101]$ (91.7%). *Bottom left:* The 3 spatial loadings for con. $[111]$ (84.2%). *Bottom right:* The 4 spatial loadings for con. $[111]$ (85.9%). *(Explained variance in brackets).*

Based on these kernel basis images it is clear to see how the full non-negativity constraint best captures the underlying spatial structure of the kernels. The theoretical optimal constraint set $[101]$ (spectral mode unconstrained) reveals a 1st components for both models, which designates the entire kernels and thus holds little spatial information. Both the 4 component models has similar components indicating a split. The best model is thus the 3 component model with full non-negativity constraint (bottom left in figure 4.2), refer to appendix B figure B.1 for a collection of model views.

The corresponding spectra are extracted from the spectral loadings via the eigenvectors used for the compression earlier. Figure 4.3 depicts the uncompressed spectral loadings.

Initially it is clear to see how the spectral non-negativity constraint has an effect on the spectral peak at appr. 1700nm (top vs. bottom illustrations). For the 3 component models (left illustrations) this spectral difference results in the spatial difference seen in figure 4.2. The spectra also suggest the optimal amount of components is 3 as two of the spectra overlap completely in the bottom right illustration (4 components, constraint $[111]$). The spectral loadings hence also suggest the 3 components model with full constraint to be optimal.

The germ side kernels are also subjected to the PARAFAC for $[3, 4, 5]$ for constraints $[101, 111]$. The results indicate a similar optimal set of parameters as for the endosperm side kernels, i.e. 3 components with non-negativity constraint on all modes). Figure 4.4 shows the resulting spatial and spectral loadings.

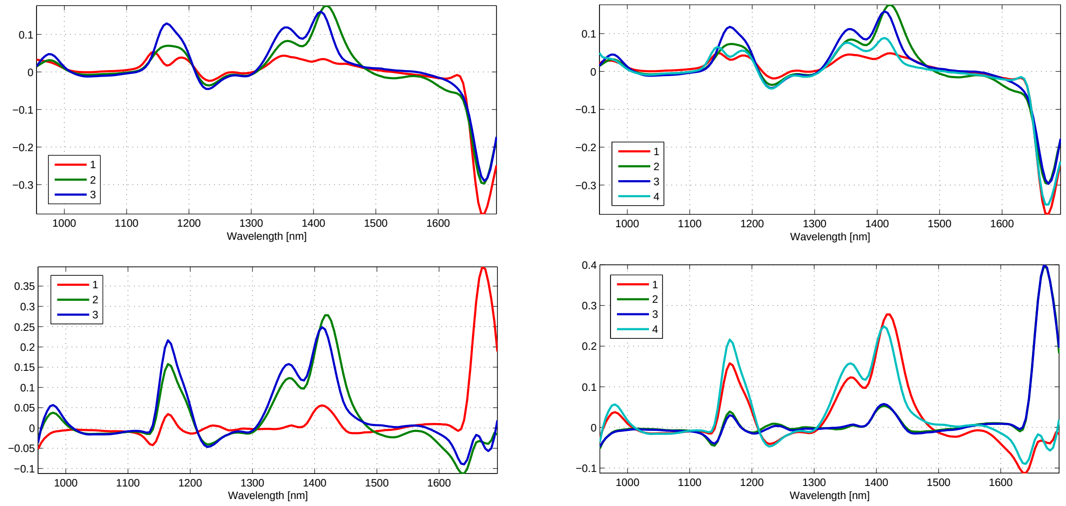

Figure 4.3: Top left: The 3 spectra con. [101] (89.6%). *Top right:* The 4 spectra for con. [101] (91.7%). *Bottom left:* The 3 spectra for con. [111] (84.2%). *Bottom right:* The 4 spectra for con. [111] (85.9%). *(Explained variance in brackets).*

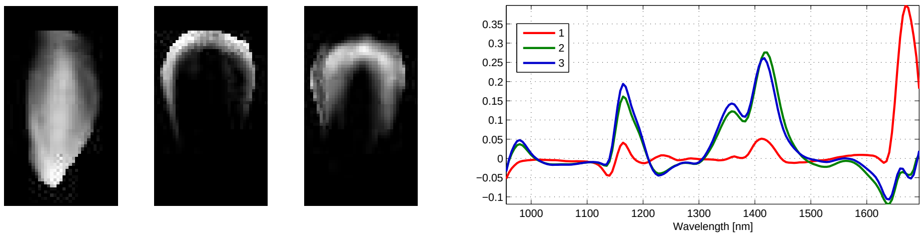

Figure 4.4: Spatial and spectral loadings for the germ side kernels for optimal PARAFAC model parameters, left and right respectively. *(Explained variance: 83.0%).*

The spatial loadings clearly reveals how the different parts of the kernels are found along with an individual spectral profile indicating the peaks discussed in section 1.

The prior analysis of the spectral compression in section 3.2 indicates that the chemical information is maintained in <10 principal components and hence the spectra were subsequently described by the 10 first scores primarily to reduce noise and to reduce the computational load. In order to analyze the effect of the spectral compression an uncompressed 3-way data array are subjected to the PARAFAC model for 3 components and constraints [101, 111]. Figure 4.5 illustrates the spatial and spectral loadings.

The full non-negative constrained model (lower illustration) has clearly captured the spatial structure of the kernels, but the truncated spectral signatures has no clear chemical meaning. In contrast the unconstrained model extract spectral profiles with relevant chemical information, but fails to distinguish the spatial distribution clearly. As part of the spectral compression the data is further decorrelated, which we suspect can have an effect on the modeling capabilites of the PARAFAC model. This requires further investigation.

4.1.1 Discussion

In the constrained PARAFAC models of both the germ side and the endosperm side we see two distinct types spectral loadings. We interpret one as describing the germ and partly also the pedicel and one describing the endosperm, as the spatial localization of these loadings overlap with the physical structures.

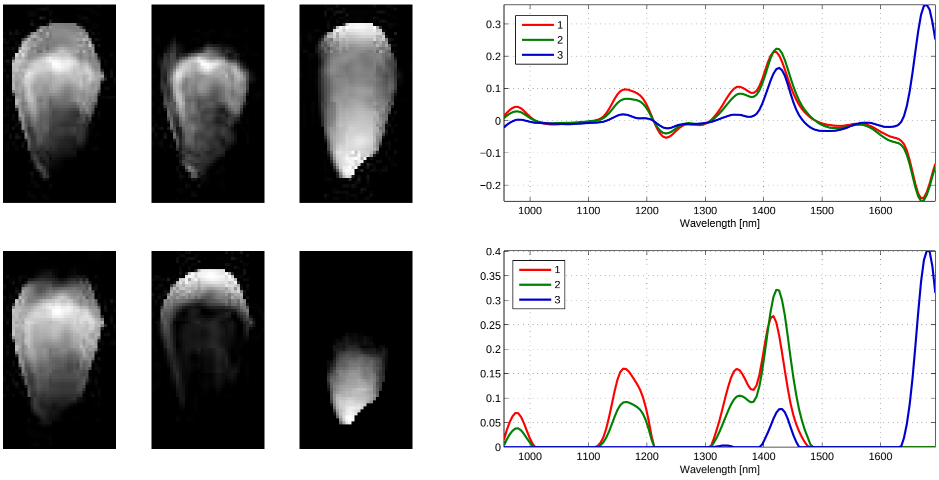

Figure 4.5: Spatial and spectral loadings for the endosperm side kernels with no spectral compression with and without spectral non-negativity constraint, upper (90.7%) and lower (72.5%) respectively. *(Explained variance in brackets).*

The germ loadings

The germ loading is dominated by a drop at 1680 nm which is caused by the absorption of oil at 1700 nm. (refer to appendix A). Although the raw spectrum of the germ and pedicel (figure 1.2, illustration 3 & 4) seem rather different the PARAFAC model has not been able to resolve this difference into two components. This could be due to the pedicel being removed from some of the kernels in the image preprocessing step.

The endosperm loadings

The endosperm loadings are rather similar, but with minor shifts and differences. This indicates that the loadings describe the same type of species. The two loadings are spatially separated. We interpret this as corresponding to two different types of endosperm, the horny and the floury endosperm. Based on a brief literature study [Watson 2003 [11] & Hoseney 1994 [7]] it seems probable that the floury type is the located nearest to the germ and the horny nearest to the epiderm (skin), but this would require further investigations for confirmation. The carbohydrates are in both cases starch and therefore the spectral difference reflects either the protein composition or in the superstructure of the protein-starch granules/complexes, or most likely a combination of both.

Score Plots

The score plots do not yield any significant information as we do not have any qualitative information about the kernels. If the kernels for example had been from different hybrids the score plot could have revealed if there was a significant difference between the hybrids. The only qualitative information we have is that they are of different size and shape and this information is partly destroyed in the process of removing the NaN pixels.

In general it is possible to distinguish 3 chemical components and their localization within the maize kernel with three component PARAFAC model. The performance of the PARAFAC model are very similar to the reference PCA model.

4.2 PARAFAC2

In forming the 3-way data array as described in section 3.5 we folded the spatial information modes X and Y leading to potential shifts in the common spatial mode XY. In terms of modeling these shifts in the

3-way data the PARAFAC2 model is attractive due to multiple loading assignment allowing individual modeling of each kernel.

The linear PARAFAC2 model is an extensions to PARAFAC and can be expressed as

$$x_{ijk} = \sum_{n=1}^{N} a_{in} b_{jkn} c_{kn} + \epsilon_{ijk} \tag{4.2}$$

where the second mode now has multiple slabs $\mathbf{B}_k = \{b_{jkn}\}_{j=1,k=1,n=1}^{J,K,N}$, one for each row in $\mathbf{C}$, subject to $\mathbf{B_k}^T\mathbf{B_k}$ is constant. This allows for individual modeling of the loading in mode $\mathbf{B}$. The model can also be visualized as illustrated in figure 4.6.

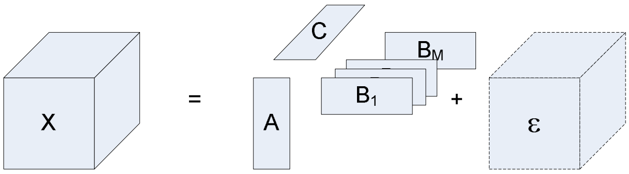

Figure 4.6: A 3-way PARAFAC2 model.

In evaluating the performance of the PARAFAC2 model we subject the endosperm side dataset for 3 components with and without non-negativity constraints on the spatial mode. The resulting kernel loadings for the unconstrained and non-negative model are depicted in figure 4.7 and 4.8 respectively as column as kernels.

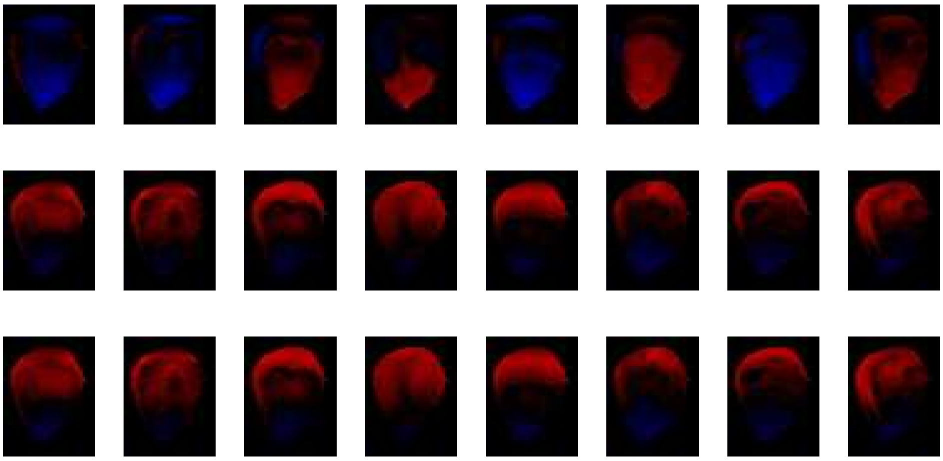

Figure 4.7: Spatial kernel loadings for a PARAFAC2 with 3 components all unconstrained (Pos/Neg, individually scaled).

Initially it can be seen how none of the models capture the sparse parts of the kernels as PARAFAC in figure 4.2. Part of the structure found is still relevant, but not with the same precision as PARAFAC. To summarize the PARAFAC2 model fails to capture the spatial shifts in the 3-way array correctly.

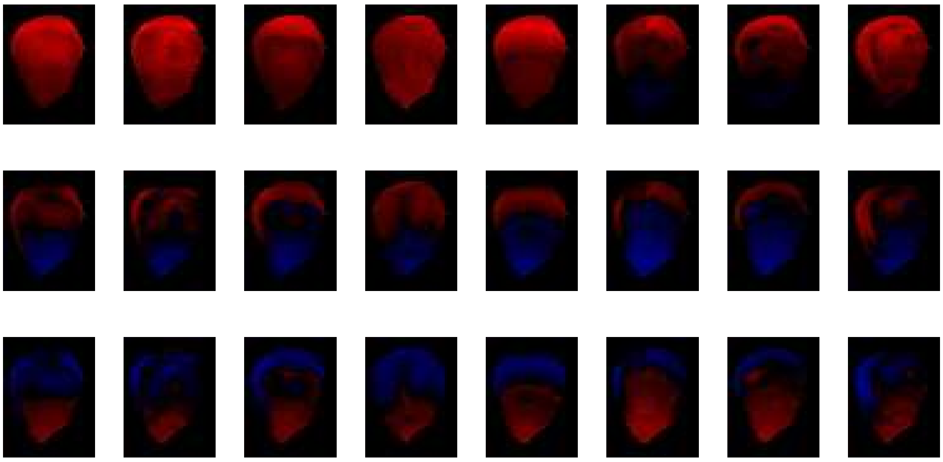

Figure 4.8: Spatial kernel loadings for a PARAFAC2 with 3 components and non-negative constrained sample and spectral mode (Pos/Neg, individually scaled).

4.3 TUCKER3

A more generalized 3-way model is the linear TUCKER3 model given by

$$x_{ijk} = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \sum_{n_3=1}^{N_3} a_{in_1} b_{jn_2} c_{kn_3} g_{n_1 n_2 n_3} + \epsilon_{ijk} \tag{4.3}$$

where a core array **G** is included to generalize the model. This leads to more flexible optimization where the core is not required to be a super diagonal as for PARAFAC models. The risk is however that the TUCKER3 model will tends to overfit the data as more parameters can be optimized. The model can also be visualized as illustrated in figure 4.9.

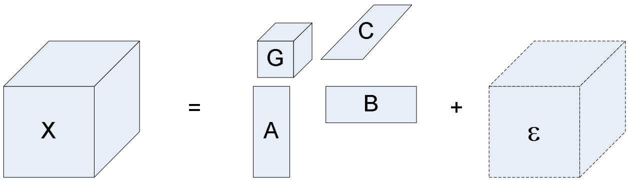

Figure 4.9: A 3-way TUCKER3 model.

The 3-way image data is subjected to the TUCKER3 model with different number of components with non-negativity on all modes. All models reveal similar results and for the $[3, 2, 3]$ (3 scores, 2 spectral and 3 spatial components) component model the spatial and spectral loadings are depicted in figure 4.10.

The captured spatial loading does not reveal a clear distribution of the any of the kernel constituents, whereas the spectral profiles are similar to those found with PARAFAC.

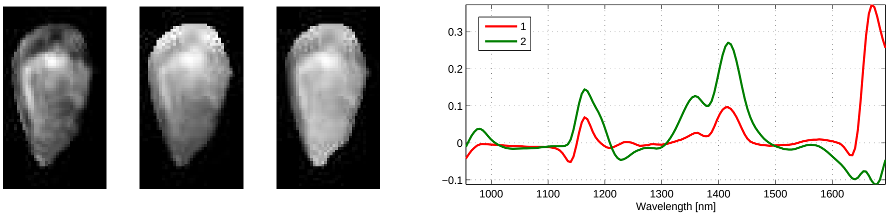

Figure 4.10: Spatial and spectral loadings for the germ side kernels for optimal TUCKER3 model parameters, left and right respectively. *(Explained variance: 88.5%)*.

5 Conclusion

This report is a proof-of-concept study to investigate the capabilities of 3-way models to describe hyperspectral imaging data and how such a model compare to a conventional 2-way model. We were able to make a PARAFAC model that had the same exploratory capabilities as our PCA reference model. To achieve this we had to impose non-negativity as a constraint on all loading. This is not logical as the spectral loading can be both negative and positive due to Savitsky-Golay preprocessing an PCA compression. Further work needs to be done to understand why this is so and which implications it has for the model.

Experiments with PARAFAC2 and TUCKER3 models did not improve the model.

The model made with a compressed dataset using 10 principal components gave a slight improvement in the separation of the spatial distribution and more distinct spectral loadings compared to the uncompressed model.

The PCA-model and the PARAFAC model gives similar spectra loadings and spatial distribution of those. Which of these is optimal is difficult to access as we lack external information. What can be concluded is that the PCA model requires much less computing power, but requires a 2D data structure, which is make it difficult to extend the PCA model into a predictive model, as further dimensional compression is needed. The PARAFAC model requires a lot of computing power and can be rather complicated both interpret and optimize. The 3D data structure is on the other hand well suited for extending into a predictive model.

6 Future Work

We have demonstrated that PARAFAC is able to model the data satisfactory. This gives a good basis for further modeling. To do this we need a dataset with more samples and more external information, both qualitative and quantitative. Qualitative information about different hybrids, growth conditions and damages could be investigated exploratively using PARAFAC. Quantitative information about the starch, protein and oil content could be used to make 3-way predictive models, such as N-PLS. Ideally the two types of informations could be combined to see the influence of various parameters on the distribution and content of nutrients.

References

[1] Morten Arngren. Modelling cognitive representations. Master's thesis, DTU Informatics, Intelligent Signal Processing, April 2007.

[2] T. M. Baye, T. C. Pearson, and A. M. Settles. Development of a calibration to predict maize seed composition using single kernel near infrared spectroscopy. *Journal of Cereal Science*, Vol.43 Issue.2:236–243, 2006.

[3] H.-D. Belitz, W. Grosch, and P. Schieberle. *Food Chemistry (3rd edition)*. Springer Verlag, 2004.

[4] R. P. Cogdill, C. R. Hurburgh, and G. R. Rippke Jr. Single-kernel maize analysis by near-infrared hyperspectral imaging. *Transactions of the American Society of Agricultural Engineers*, Vol. 47(1):311–320, 2004.

[5] P. Geladi, D. MacDougall, and H. Martens. Linearization and scatter-correction for near-infrared reflectance spectra of meat. *Applied Spectroscopy*, Vol.39 Issue.3:491–500, 1985.

[6] A.A. Gowen, C.P. O'Donnell, P.J. Cullen, G. Downey, and J.M. Frias. Hyperspectral imaging - an emerging process analytical tool for food quality and safety control. *Trends in Food Science & Technology*, Vol.18 Issue.12:590–598, 2007.

[7] R. Carl Hoseney. *Principles of Cereal Science and Technology*. American Association of Cereal Chemists, 1994.

[8] H. Y. Jiang, Y. J. Zhu, L. M.Wei, J. R.Dai, T. M.Song, Y. L. Yan, and S. J. Chen. Analysis of protein, starch and oil content of single intact kernels by near infrared reflectance spectroscopy (nirs) in maize (zea mays l.). *Plant Breeding*, Vol.126 Issue.5:492–497, 2007.

[9] Bruce A. Orman and Robert A. Schumann Jr. Comparison of near-infrared spectroscopy calibration methods for the prediction of protein, oil, and starch in maize grain. *J. Agric. Food Chem.*, 39 (5):883–886, 1991.

[10] Age Smilde, Rasmus Bro, and Paul Geladi. *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley, August 2004.

[11] S.A. Watson. *Corn: chemistry and technology*. Am. Assoc. Cereal Chem., St Paul, Minn., USA, 1987.

[12] B. André Weinstock, James Janni, Lisa Hagen, and Steven Wright. Prediction of oil and oleic acid concentrations in individual corn (zea mays l.) kernels using near-infrared reflectance hyperspectral imaging and multivariate analysis. *Appl Spectrosc*, 60(1):9–16, Jan 2006.

A Appendix: Food Absorption Wavelengths.

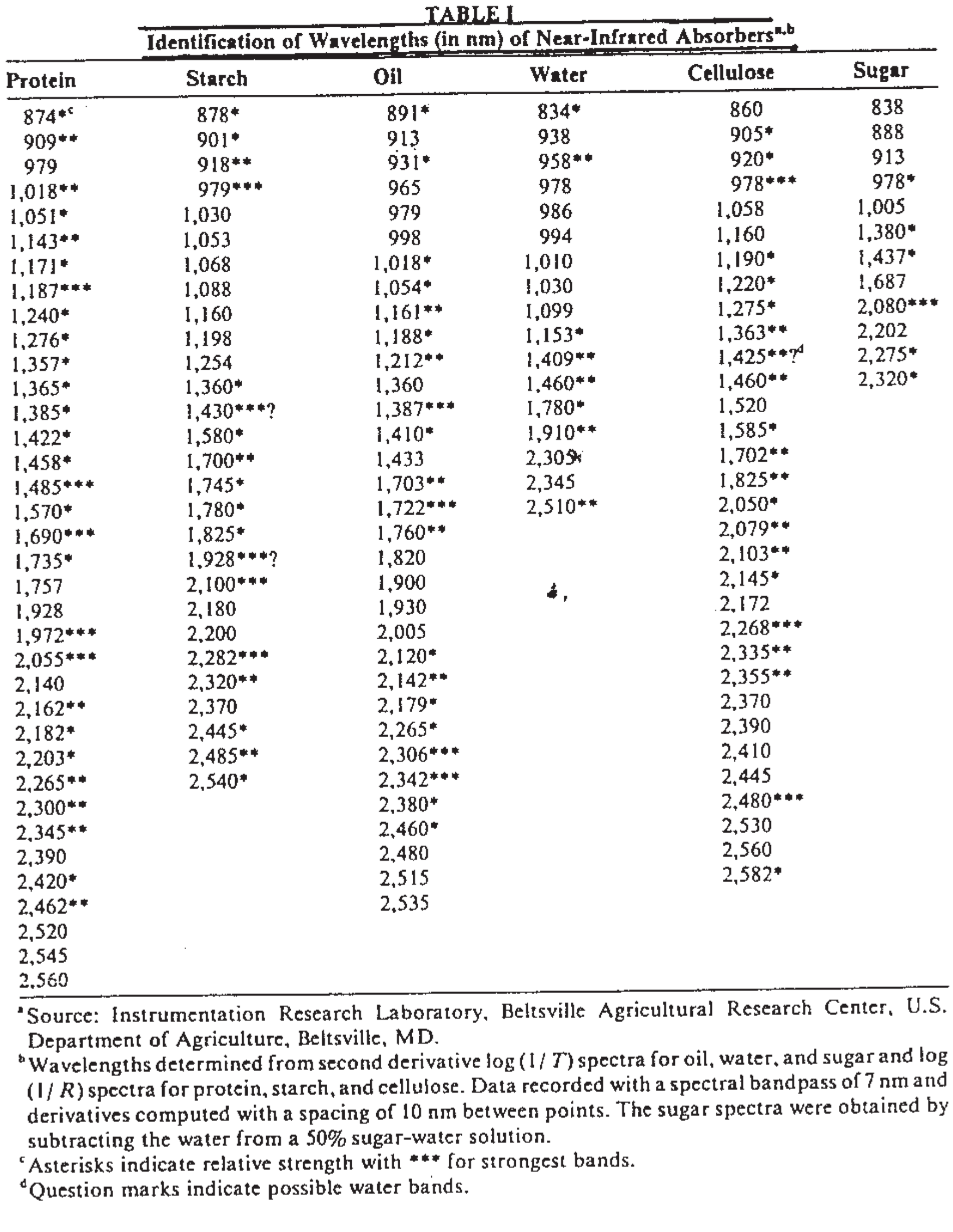

TABLE I

Identification of Wavelengths (in nm) of Near-Infrared Absorbers[a,b]

Protein	Starch	Oil	Water	Cellulose	Sugar
874*[c]	878*	891*	834*	860	838
909**	901*	913	938	905*	888
979	918**	931*	958**	920*	913
1,018**	979***	965	978	978***	978*
1,051*	1,030	979	986	1,058	1,005
1,143**	1,053	998	994	1,160	1,380*
1,171*	1,068	1,018*	1,010	1,190*	1,437*
1,187***	1,088	1,054*	1,030	1,220*	1,687
1,240*	1,160	1,161**	1,099	1,275*	2,080***
1,276*	1,198	1,188*	1,153*	1,363**	2,202
1,357*	1,254	1,212**	1,409**	1,425**?[d]	2,275*
1,365*	1,360*	1,360	1,460**	1,460**	2,320*
1,385*	1,430***?	1,387***	1,780*	1,520	
1,422*	1,580*	1,410*	1,910**	1,585*	
1,458*	1,700**	1,433	2,305*	1,702**	
1,485***	1,745*	1,703**	2,345	1,825**	
1,570*	1,780*	1,722***	2,510**	2,050*	
1,690***	1,825*	1,760**		2,079**	
1,735*	1,928***?	1,820		2,103**	
1,757	2,100***	1,900		2,145*	
1,928	2,180	1,930		2,172	
1,972***	2,200	2,005		2,268***	
2,055***	2,282***	2,120*		2,335**	
2,140	2,320**	2,142**		2,355**	
2,162**	2,370	2,179*		2,370	
2,182*	2,445*	2,265*		2,390	
2,203*	2,485**	2,306***		2,410	
2,265**	2,540*	2,342***		2,445	
2,300**		2,380*		2,480***	
2,345**		2,460*		2,530	
2,390		2,480		2,560	
2,420*		2,515		2,582*	
2,462**		2,535			
2,520					
2,545					
2,560					

[a] Source: Instrumentation Research Laboratory, Beltsville Agricultural Research Center, U.S. Department of Agriculture, Beltsville, MD.

[b] Wavelengths determined from second derivative log (1 / T) spectra for oil, water, and sugar and log (1 / R) spectra for protein, starch, and cellulose. Data recorded with a spectral bandpass of 7 nm and derivatives computed with a spacing of 10 nm between points. The sugar spectra were obtained by subtracting the water from a 50% sugar-water solution.

[c] Asterisks indicate relative strength with *** for strongest bands.

[d] Question marks indicate possible water bands.

Figure A.1: Source: Unknown.

B Appendix: PARAFAC Model.

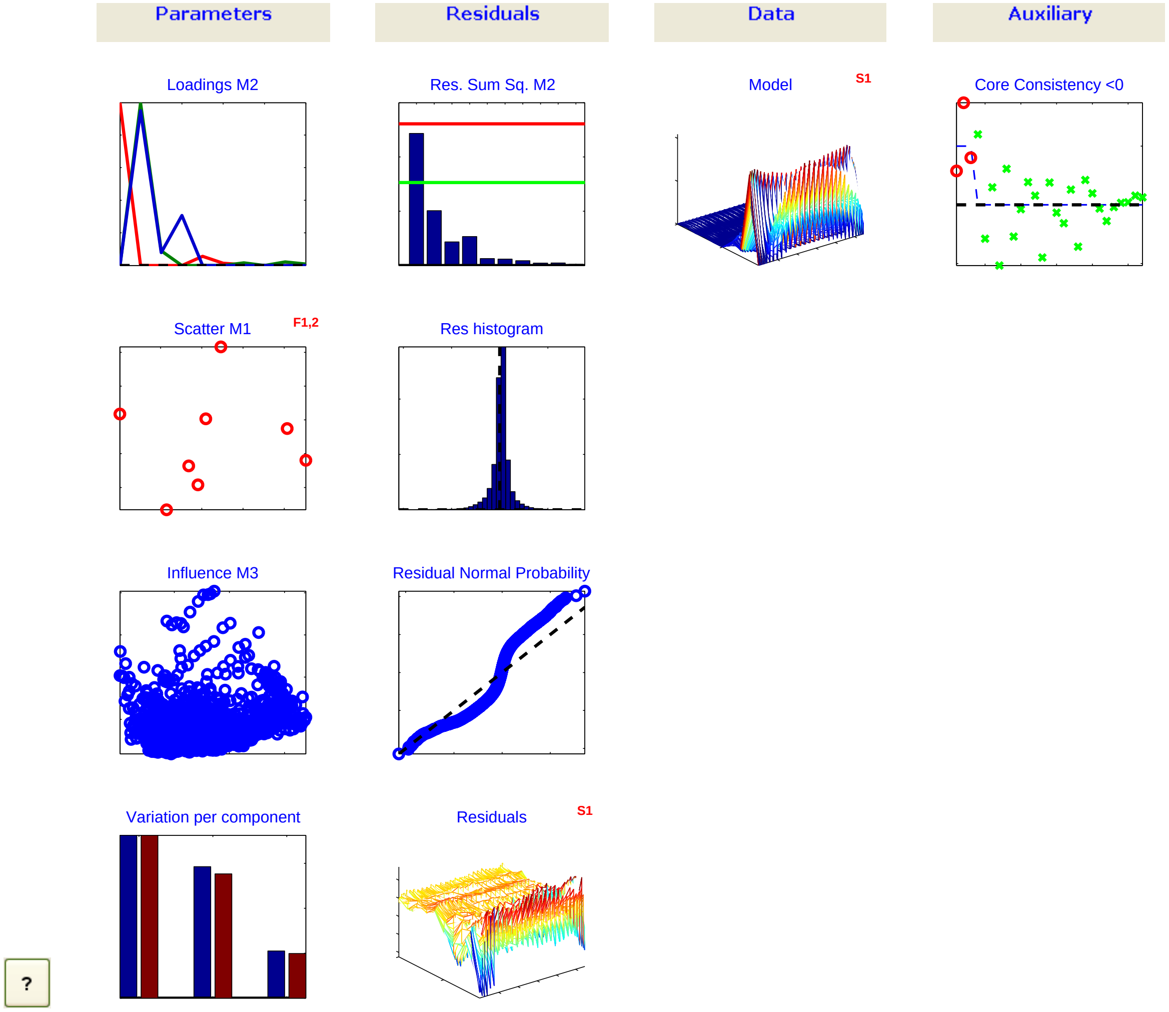

Figure B.1: Model view of the PARAFAC model with 3 components and non-negativity constraint of all modes for endosperm kernels.

C Appendix: MATLAB code.

MATLAB code - 3-Way Analysis of Maize Kernels.

```
% 3-Way Modelling of Maize Kernels, Matlab code
%
% Copyright 2009, Morten Arngren and Christian Schack Pedersen

clear;

path(path, '..\..\..\Matlab-FU\Toolbox\Arngren');
path(path, '..\..\..\Matlab-FU\Toolbox\ENVI');

%% Initialise
initPlot;

wvOut      = [1:12];         % Exclude wavelengths from 900-950nm, very noisy
Crop       = 'all';          % Default size if none specified
doPlot     = 1;
Pre        = 'rfl';          % Absorbance 'abs' or Reflectance 'rfl'

%% Model Parameters

kernelID = [9:16];

Scatter  = 'SG';         % SG, MSC, MD
SpecComp = 1;            % Compress spectras via PCA
NoPCA    = 10;           % PC => 20 gives no additional significant information.

% Savitzky-Golay
SG_Width = 7;
SG_Order = 2;
SG_Deriv = 1;

maskThr = 1;             % Thresold for background mask
nanThr  = 0.9;             % Threshold for active pixels

empty = NaN;

%%

% Cube filename, comment those not needed
fname  = ('..\..\..\Camera\Images\KVL_data\Maize_1');
fname2 = ('..\..\..\Camera\Images\KVL_data\Maize_2');

% Read cube, ref and dark image
cubeRef       = enviread('..\..\..\Camera\Images\KVL_data\Ref');
cubeDark      = enviread('..\..\..\Camera\Images\KVL_data\Dark');
[cubeRead wv] = enviread(fname);
cubeRead2     = enviread(fname2);

% Locate saturated pixels
cubeSat(cubeRead, 2^14-1, doPlot);

[cubeL wv Size] = prepareCube(cubeRead,  wv, cubeRef, cubeDark, Crop, Pre, wvOut, 0);
cubeR           = prepareCube(cubeRead2, wv, cubeRef, cubeDark, Crop, Pre, wvOut, 0);
```

```
% Append two cubes, e.g. front and back of maize
figure(99); [cube Size] = appendCube(cubeL, cubeR);


%% %%%% Pre-process


%% - Mean oversampling
% Data is oversampled by a factor of 2, i.e. eacvh Y-line is meaned into 1.

cube   = (cube(1:2:end,:,:) + cube(2:2:end,:,:))/2;
Size.Y = Size.Y/2;

figure(1), viewCubeRGB(cube, [1 89 150], 'rgb');


%% Remove background by extracting mask

% Apply PCA on reflection data prior to any pre-processing.
[E D PcaIM] = cubePCA(cube, wv, [], Size, 2, 1);
mask2D      = segmIM(PcaIM(:,:,1), maskThr, 1, 0);


%% Convert to absorbance
% Must do early to avoid negative spectra value leading to complex values.
cube = -log10(cube);


%% Remove scattering
% Pre-process spectra to remove scatter effects

% View scattering prior to removal
a = reshape(cube, [Size.X*Size.Y Size.Bands])';
figure(20), plot(wv, a(:,1:1000:end)); grid on; axis tight;
  ylabel('Absorbance'); xlabel('Wavelength␣[nm]');

% Unfold cube
S = reshape(cube, [Size.X*Size.Y Size.Bands])';

% Savitzky-Golay
if strcmp(Scatter, 'SG')
  [SG, cm] = savgol(S', SG_Width, SG_Order, SG_Deriv);
  cube     = reshape(SG, [Size.Y Size.X Size.Bands]);

  figure(21), plot(wv, SG(1:1000:end,:)'); grid on; axis tight;
    xlabel('Wavelength␣[nm]');
end

% Multiplicative scatter/signal correction
if strcmp(Scatter, 'MSC')
  win = mean(cube,3);
  win(27:67,19:32,:) = 1;
  figure(99), viewCubeRGB(win, [1 1 1], 'rgb');

  S_          = reshape(cube(27:67,19:32,:), [41*14 Size.Bands]);
  tmp         = preprocess('default','MSC␣(mean)');
  [datap, sp] = preprocess('calibrate', tmp, S_);
  SMS          = preprocess('apply', sp, S');
  SMS          = SMS.data;

%   [SMS, alpha, beta, xref] = mscorr(S',[], 1);

  cubeTmp = reshape(SMS, [Size.Y Size.X Size.Bands]);
  meanIM  = squeeze(mean(cubeTmp, 3));

  figure(22), plot(wv, SMS(1:50:end,:)'); grid on; axis tight;
    ylabel('Absorbance??'); xlabel('Wavelength␣[nm]');
end


%% - Reduce dimensions of spectra

if SpecComp
  [E D PcaIM] = cubePCA(cube, wv, [], Size, 6, 1);

  % Extract only significant dimensions
  cube = PcaIM(:,:,1:NoPCA);
end

%% Set background pixels to NaN
fprintf('|-␣Remove␣Background...');

imTmp          = empty*ones([Size.Y Size.X]);
imTmp(mask2D) = 1;
imTmp          = imTmp(:,:,ones(1,size(cube,3)));
cube           = cube.*imTmp;

fprintf('Ok\n');

%% Segment Kernels
fprintf('|-␣Segment␣Kernels...');

if 0
  % - Segment single kernels
  [kernels Crop] = extractKernels(cube, [38 66]);
  NoKernels      = size(kernels,2);
  save Kernel_Coords Crop NoKernels;
else
  load Kernel_Coords;
  for i = 1:NoKernels
    kernelAll{i} = cropCube(cube, Crop{i}, 0);

    % Remove empty NaN slabs
    kernelAll{i}(1,:,:) = [];             % Remove 1st row in image
    kernelAll{i}(:,1,:) = [];             % Remove 1st column in image
  end
end

% Process kernels manually, remove overlap
kernelAll{3}(:,end-2:end,:)  = empty;
kernelAll{11}(:,1:2,:)       = empty;

% Extract size of kernels
[Size.Y Size.X Size.Bands] = size(kernelAll{1});
```

```
% Divide into front and back
for i = 1:length(kernelID)
  kernels{i} = kernelAll{kernelID(i)};
end
NoKernels = size(kernels,2);

% View individual kernels
figure(10), set(gcf,'Name','Kernels');
  viewKernels(kernels);

fprintf('Ok\n');

%% - Form 3-way array
fprintf('|-␣Form␣3-Way␣Array...');

% Vectorise X and Y dimensions
K3W = zeros(NoKernels, Size.Bands, Size.X*Size.Y);
for i = 1:NoKernels
  K3W(i,:,:)   = reshape(kernels{i}, [Size.X*Size.Y Size.Bands])';
end
fprintf('Ok\n');

K3Ws = K3W; id = 1; j = 1;
clear actID;
for i = 1:size(K3Ws,3)
  a = squeeze(K3Ws(:,:,id));
  if 1-sum(sum(isnan(a))) / (NoKernels*Size.Bands) < nanThr;
    K3Ws(:,:,id) = [];
  else
    actID(j) = i;
    j        = j + 1;
    id       = id + 1;
  end
end

% View individual kernels after truncation by transforming back to images.
K3Wtmp = NaN*ones(NoKernels, Size.Bands, Size.X*Size.Y);
IM     = zeros(NoKernels, Size.Bands, Size.X*Size.Y);
for i = 1:size(actID,2)
  K3Wtmp(:,:,actID(i)) = K3Ws(:,:,i);
  IM(:,:,actID(i))     = 1;
end
for i = 1:NoKernels
  kernelsTmp{i} = reshape(squeeze(K3Wtmp(i,:,:))', [Size.Y Size.X Size.Bands]);
end

figure(11), set(gcf,'Name','Kernels␣after␣truncation');
  viewKernels(kernelsTmp);

%% %%%%% Models

%% 3-Way PARAFAC with sparse data

C = [{'101'}];            % Constraint

NoCom                           = 3;
opt                             = parafac('options');
opt.plots                       = 'off';
opt.init                            = 50;
opt.stopcriteria.iterations         = 5000;
opt.stopcriteria.relativechange = 1e-9;
opt.stopcriteria.absolutechange = 1e-9;

if str2num(C{i}(1)), opt.constraints{1}.type = 'nonnegativity'; end
if str2num(C{i}(2)), opt.constraints{2}.type = 'nonnegativity'; end
if str2num(C{i}(3)), opt.constraints{3}.type = 'nonnegativity'; end

% Run PARAFAC
MP3s = parafac(K3Ws, NoCom, opt);

100*(1-MP3s.detail.ssq.residual/MP3s.detail.ssq.total); % Expl. variance

% Tranform Sparse 3-way array back to images
figure(30), set(gcf,'Name','Spatial␣Loadings'); clear IM;
for j = 1:NoCom
  IM{j} = NaN*ones(Size.Y*Size.X,1);
  for i = 1:size(actID,2)
    IM{j}(actID(i)) = MP3s.loads{3}(i,j);
  end
  IM{j} = reshape(IM{j}', [Size.Y Size.X]);
  subplot(1,NoCom,0+j), viewCubeRGB(IM{j}, [1 1 1], 'rgb');
end

% View Spectra
spec = E(:,1:NoPCA)*MP3s.loads{2};
figure(33), set(gcf,'Name','Spectra␣Loadings');
  plot(wv, spec); grid on; axis tight; legend('1','2','3','4','5','6','7','8');
  xlabel('Wavelength␣[nm]');

% Score plot
figure(33), set(gcf,'Name','Score␣Plot'); hold on;
  for i = 1:NoKernels
    plot(MP3s.loads{1}(i,1),MP3s.loads{1}(i,2), 'or', 'LineWidth', 1); grid on;
    text(MP3s.loads{1}(i,1),MP3s.loads{1}(i,2), num2str(i), 'FontWeight', 'bold');
  end
  hold off;

%% PARAFAC2

NoCom = 3;

option = parafac2('options');
option.constraints{1}.nonnegativity = 0;
option.constraints{2}.nonnegativity = 1;
option.constraints{3}.nonnegativity = 1;

% Run PARAFAC2
MP3s_2 = parafac2(permute(K3Ws, [3 2 1]), NoCom, option);

% Tranform Sparse 3-way array back to images
figure(35); clear IM;
for k = 1:NoKernels
  a = MP3s_2.loads{1}.P{k}*MP3s_2.loads{1}.H;
  for j = 1:NoCom
    IM{j} = NaN*ones(Size.Y*Size.X,1);
    for i = 1:size(actID,2)
```

```
      IM{j}(actID(i)) = a(i,j);
    end
    IM{j} = reshape(IM{j}’, [Size.Y Size.X]);
    subplot(3,NoKernels,(j-1)*NoKernels+k), viewCubeRGB(IM{j}, [1 1 1], ’rgb’); axis off;
  end
end

% View Spectra
spec = E(:,1:NoPCA)*MP3s_2.loads{2};
figure(36), plot(wv, spec); grid on; axis tight;

%% TUCKER3

NoCom = [3 3 3];

opt                   = tucker(’options’);
opt.plots             = ’off’;
opt.init              = 50;

opt.stopcrit(1)       = 1e-9;
opt.stopcrit(2)       = 1e-9;
opt.stopcrit(3)       = 5000;
opt.stopcrit(4)       = 3600;
opt.constraints{1}.nonnegativity = 1;
opticonstraints{2}.nonnegativity = 1;
opt.constraints{3}.nonnegativity = 1;
opt.constraints{3}.nonnegativity = 1;

% Run TUCKER3
MT3s = tucker(K3Ws, NoCom, opt);

% Tranform Sparse 3-way array back to images
figure(31), set(gcf,’Name’,’Spatial␣Loadings’); clear IM;
for j = 1:NoCom
  IM{j} = NaN*ones(Size.Y*Size.X,1);
  for i = 1:size(actID,2)
    IM{j}(actID(i)) = MT3s.loads{3}(i,j);
  end
  IM{j} = reshape(IM{j}’, [Size.Y Size.X]);
  subplot(1,NoCom,0+j), viewCubeRGB(IM{j}, [1 1 1], ’rgb’);
end

% View Spectra
spec = E(:,1:NoPCA)*MT3s.loads{2};
% spec(:,1) = -spec(:,1);
% spec(:,2) = -spec(:,2);
figure(32), set(gcf,’Name’,’Spectra␣Loadings’);
  plot(wv, spec); grid on; axis tight; legend(’1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’);
  xlabel(’Wavelength␣[nm]’);

% Score plot
figure(33), set(gcf,’Name’,’Score␣Plot’); hold on;
  for i = 1:NoKernels
    plot(MT3s.loads{1}(i,2),MT3s.loads{1}(i,3), ’or’, ’LineWidth’, 1); grid on;
    text(MT3s.loads{1}(i,2),MT3s.loads{1}(i,3), num2str(i), ’FontWeight’, ’bold’);
  end
  hold off;
```