Multivariate Image Regression (MIR) for Quantitative Predictions

- Prototype Software Implementation and Selected Industrial-Technological Pilot Studies

Doctoral Thesis

by

Thorbjørn Tønnesen Lied

Telemark University College (HiT)

and

the Norwegian University of Science and Technology (NTNU)

Porsgrunn, Norway

November 2000.

Contents

List of Papers

This thesis is based on the following papers which will be referred to by the Roman numerals I-V:

I. K. Esbensen, T.T. Lied, K. Lowell and G. Edwards. *Principles of Multivariate image Analysis (MIA) in remote sensing, technology and industry*. (Submitted for publication, 2000).

II. T.T. Lied, P. Geladi and K. Esbensen. *Multivariate Image Regression (MIR): implementation of image PLSR – first forays*. Journal of Chemometrics, 14 (2000) pp. 585-598

III. T.T. Lied and K. Esbensen. *Principles of MIR, Multivariate Image Regression I: regression typology and representative application studies.* (Submitted for publication).

IV. T.T. Lied and K. Esbensen. *Principles of MIR, Multivariate Image Regression II: Cross Validation – what you see is what you get.* (Submitted for publication, 2000).

V. T.T. Lied et. al. *Image-Analytical Quantitative Monitoring of Heterogeneous Mixture Processes: Angle Measure Technique (AMT) vs. Multivariate Image Regression (MIR)* (Submitted for publication 2000).

Acknowledgements

I wish to thank all my colleagues, both inside and outside of the Applied Chemometrics Research Group (KF) at Telemark University College (HiT) and Tel-Tek, who have contributed to the present work.

I also wish to dedicate my special thanks to:

- Professor Kim H. Esbensen who believed in me and gave me the opportunities and challenge to get involved in this work. His engagement and efforts have been beyond what can possibly be expected.
- All the project partners at MATFORSK for supporting me with data, literature and exchange of thoughts and ideas.
- Professor Paul Geladi at University of Umeå for his valuable introduction to the world of KERNEL algorithms.
- Maths Halstensen for the opportunity of sharing problems and ideas.
- Inger Hedvig Matveyev for assistance in image acquisition and personal support at hard times.
- Magnar and Mette Ottøy for personal support and encourage when I was in doubt.
- Jun Huang for his contributions and efforts in AMT.
- My parents for their never-ending believe in me.
- And last, but not least: Lise, Nicolai and Victoria who waited for me to finish and gave me the strength to carry on, and for showing me there is more to life than principal components.

Introduction

In large and important sectors of modern production, there is an increased demand for on-line or at-line information. Both consumers and governmental regulations require that producers can *document* the quality of their products. The more precise measurements can be, the more potentially valuable they are for the producers in their endeavours to fulfil these demands.

Usually, however, precision has its price. In many measurement systems samples have to be removed from the production line to be analysed in a laboratory. When the producer needs continuous surveillance of the products, off-line methods are inefficient and expensive. Searching for on-line measurement systems capable of extracting the information in demand will always be a topic of interest.

It has been said in many occasions that "*An image says more than a thousand words*". In many cases this is true regarding the very promising area of multivariate image analysis, but there are indeed also potentially many problems regarding teaching electronic components to interpret these digital signals the same way humans can extract information from visual input. Not only are humans extremely adept at recognising patterns and objects, but our vision system is also very compensative when variations occur. These superior human abilities can only be matched with very great difficulties (if at all) by the present powerful image-analytical methods, some of which make up the major parts of the present thesis.

Thus, in many cases, image analysis strive to imitate the human way of interpreting visual information. Because of the complexity involved, it is my belief that digital analysis of visual information should often be adapted to what may be totally different ways of treating the data, ways which may initially seem odd to humans.

For people without chemometric experience, the methods discussed below may then well seem unfamiliar. But because a computer is doing the job, we must design the job in a way in which the computer can handle the complex world of image data, so as to use its facilities to the maximum. Based on the results in this thesis, it is my intention that the methods discussed will prove to be useful for future implementations in real-world, image analytical measurement systems.

Scope of project

The present work has been financed by the Norwegian Research Council (NFR) as a Strategic Program (SIP), project No. 51120/100. The title of the project was *"Efficient control and assurance of product quality in food-, feed- and process industry."* It was organized as collaborative project between the Norwegian food research institute (MATFORSK), Telemark University College (HiT) and Telemark Technological Research and Development Centre (Tel-Tek) who has been my formal employer.

The overall objective, out of which my project was about one third, was to develop rapid and reliable methods for quantification and measurement systems within food science and technology. Hence the extensive use of food-related examples in this thesis. The two other Ph.D. projects in this collaboration were Jens Petter Wold[1] and Bjørn-Helge Mevik[2].

In the beginning of the project it was the intention that I especially should study multivariate image texture analysis (MIX) which is explained elsewhere in the present thesis. For this I should build a large texture filter database, and apply these to experimental data which were to be analysed on an adequate software system, MIR (Multivariate Image Regression) . This latter was going to be developed by a forth participant in the project, who unfortunately had to leave the project in midterm due to personal matters. This lead to that I had to do the MIR programming myself, leaving less time for the filter database.

As it turned out, however, the principles of MIX were not all that new as we first believed. Therefore, efforts were made to focus on other aspects of multivariate image regression (MIR) for quantitative measurements. Thus when working on the last paper (V), a totally new (as far as I know) approach for combining spectral and spatial information in regression analysis was developed. Therefore I believe it is safe to say that the task of the overall project objectives still have been achieved, and that an appropriate amount of new knowledge has been brought to the world.

Prior to the work on this thesis, I was educated as an Charted Engineer in Environmental Technology, with some basic knowledge to chemometrics, some programming experience and 10 years with photography as my major hobby. In order to do the present Ph.D. research I have had to study more on chemometrics, signal- and image processing and –analysis, programming and multivariate image analysis (MIA) and –regression (MIR). In addition to this, I feel that working with the practical industrial and technological examples have given me considerable understanding in the field of MIA and MIR.

Theory

This chapter describes some of the important aspects of multivariate image analysis (MIA) and –regression (MIR). Most of the theory herein is based of the work of others, the rest was initially published in the papers I-V. References will be given where appropriate.

The multivariate Image

The term "*multivariate image*" [3] is here used to describe any digital image consisting of a multiple of spatially consistent channels. One channel may represent a colour or a different part of the electromagnetic spectrum, a different imaging technique, a specific time etc [4]. When treating different spectral channels, the terms *multi- spectral* or *hyper-spectral* is often used. The actual number of channels is not important for the methods described in this thesis, but it is crucial for an image analytical measurement system that the number of channels and choice of spectral wavelengths are *optimised* with regard to both quantification and measurement precisions. If e.g. the imaging techniques used does not manage to distinguish different species in a quantification system, the methods described in the present thesis will not work.

Independent on the technique used to acquire a multivariate image, it can be visualised as a 3D OOV[5] matrix where two of the ways are objects (M x N pixels) and the third way is variables[6], e.g. spectral channels. In Figure 1 the variable way is vertical and the object ways are in the horizontal plane.

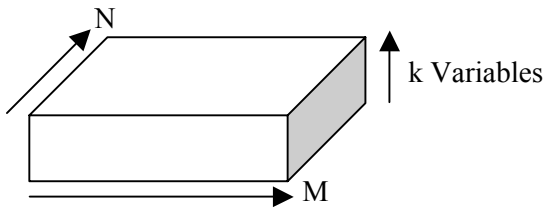

Figure 1. The Multivariate Image as a 3D matrix

Because the techniques described below treats each pixel as a separate object independent of its neighbours, the 3D matrix is usually re-arranged into a 2D OV matrix where each channel is a (M x N) long vector[4]. On this matrix, 2-way algorithms can be applied, as described below.

MIA: Image PCA

Multivariate Image Analysis [7, 8] was introduced by Esbensen and Geladi in 1989[9]. However, the 2-way calculation core aspects of MIA , the principal component decomposition, was initially described already in 1933 [10].

Principal Component Analysis (PCA) is a data transformation that decomposes the matrix X into scores T, loadings P and residuals E, so that

$$\mathbf{X=TP' + E}$$

This can be interpreted so that the structural information is organised in the scores T, the noise is gathered in the residuals E and the loadings P contains the transformation information.

Basically, MIA calculates the loadings as eigenvectors from the cross product matrix, the covariance matrix or the correlation matrix of the re-arranged multivariate image. If the re-arranged image is called X, the three different matrixes are given by:

Matrix	**Definition** [4]
Cross Product	$Z = X'X$
Covariance	$Z_{cov} = [1/(N-1)]\ X_{mc}'X_{mc}$ where X_{mc} is mean centred and N is the number of objects
Correlation	$Z_{cor} = [1/(N-1)]\ X_{mcw}'\ X_{mcw}$ Where X_{mcw} is X_{mc} [1/s], where s is column-wise standard deviation of X_{mc}.

From one of these matrixes, loadings can be calculated using Singular Value Decomposition (SVD). When loadings are available, scores are calculated using

$$\mathbf{T=XP}$$

The unused information in X will be placed in E, so that

$$\mathbf{E=X-TP'}$$

Figure 2 shows the different matrixes involved in the calculations.

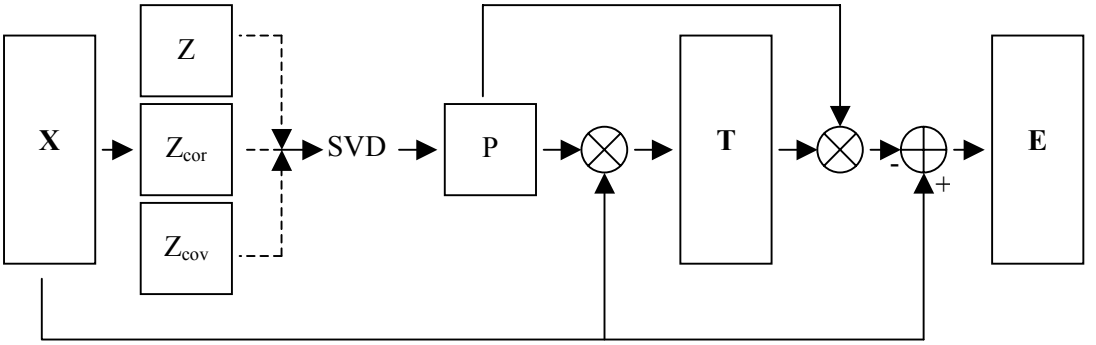

Figure 2. The MIA PCA decomposition diagram

The most important benefits of the PCA transformation is that the structural information in X is organized in a decreasing manner in T, so that the most dominating structure is found in the first component, the second most dominating structure in the second component etc. This way, structural information spread over a large number of original, "raw" variables in X is compressed into a few principal components in T.

Secondly, and also important, is the orthogonal feature of the components. Because every component in T is orthogonal to all other components, it gives well-reasoned meaning plotting them against each other in score plots. This very often gives an empirical, increased insight into the relevant data covariance structures, i.e. both the inter-variable as well as the inter-object (spatial) relationships.

Score plots are actually 2D histograms where each way represents one component. Every (x,y) position in the score plot contains the number of objects with score-pairs corresponding to this score-pair. The larger this number is, the more objects display *identical* score-combinations in the two components.

Figure 3 [11] shows an example of a typical MIA scoreplot, where colour codes are used to illustrate the frequency of score pairs in each position. This is done to enhance the visual inspection, as the human eye is more able to distinguish colours than grey-levels [12]. In this type of plot there is a truly remarkable potential information to be gained if guided by the proper MIA strategy. Paper I is devoted entirely to formulating a general, complete and flexible multivariate image analysis strategy, the core of which was originally stated to have comprised a central part of Geladi and Grahn [4].

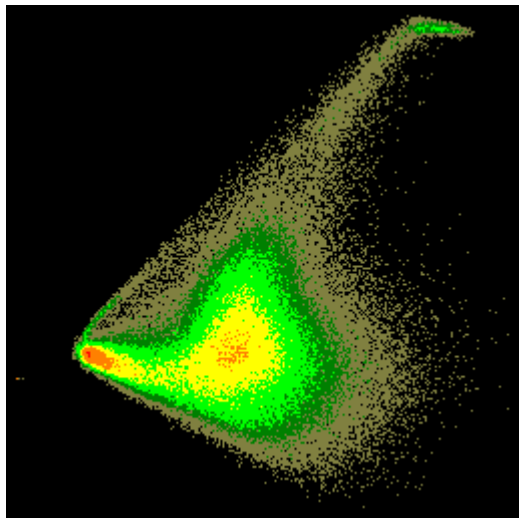

Figure 3. Example MIA score plot showing component 1 vs. component 2. This figure is from paper I, where the original data is presented in full detail.

This type of score plot and a technique called "*MIA-mask brushing"* described below, are the most important tools that MIA has added to the multivariate image analysis realm.

Brushing is a technique which correlates the score plot with the corresponding original raw image, i.e. the scene space. With the MIA-mask brushing tool, it is possible to map information *between* the two spaces. By drawing a mask or a Region of Interest (ROI) in the score plot and having the corresponding pixels highlighted in the scene space, the user is able to identify (and perhaps classify) objects with similar score combinations, independently of position and form in the scene space. The brushing technique is demonstrated on pages 28-29.

In addition to the score plot, score data and raw data can be visualised in scene space as grey-level images or colour-composition RGB images. In the present implementations the brushing works also from scene space to score space.

Image PLS; Algorithms and Tools

Multivariate Regression concerns calibrating models between two sets of data. In addition to the X-matrix introduced above, there is also an Y-matrix, or –vector, containing the dependent variable(s). The goal of the calibration is to establish a regression model between X and Y, so that in the future, Y can be calculated (or predicted) from X. In most cases, X will typically contain some multivariate measurements of a phenomenon, while Y will contain the information that are sought, e.g. some kind of state (e.g. concentration) of the phenomenon.

The idea of Multivariate Image Regression (MIR) was initially introduced in 1991 by Geladi and Esbensen [13], and strategies were published by Esbensen, Geladi and Grahn the year after [14]. Initially, MIR was based on Principal Components Regression (PCR). PCR uses PCA in the decomposition of X, and creates a regression model based on the scores from PCA. PCR is thus *unguided* in its decomposition, principally opening for the possibilities of a sub-optimised decomposition.

A different approach, Partial Least Squares Regression (PLS) [15-16] overcomes this problem. PLS performs Y-guided decomposition of the X matrix, so that Y-related information in X also gets priority in deriving T. A potential drawback with this approach lies in problems related to a noisy Y, which can cause overfitting of models[17], if not properly validated [18].

In the original algorithms[17], PLS is performed on the entire X and Y matrixes. This is no problem when the number of objects and/or variables are small. In MIR, however, the opposite is the case. Carrying around with 10^6-10^7 pixels or more in a calibration, the available computer memory may easily become an issue. In such cases, the Kernel algorithm presented by Lindgren and Geladi [19] is a powerful way of reducing memory consumption in the calibration. Because of this, the Kernel PLS approach was chosen in the present MIR implementation (II). Kernel PLS is based on the work by Höskuldson [20]

Figure 4 shows the outline of the Kernel PLS approach. Because PLS is iterative, the X and Y matrixes needs to be updated after each component has been calculated. In Kernel PLS, however, this updating is done on the smaller Kernel matrixes; X'X, X'Y and Y'Y. The master Kernel matrix, the X'YY'X is used for the decomposition. In the present implementation, SVD is used to extract loading weights W, X'X gives X-loadings P, X'Y gives Y-loading Q and Y'Y is used for calculating the explained variance in Y. MATLAB code for this algorithm is listed below.

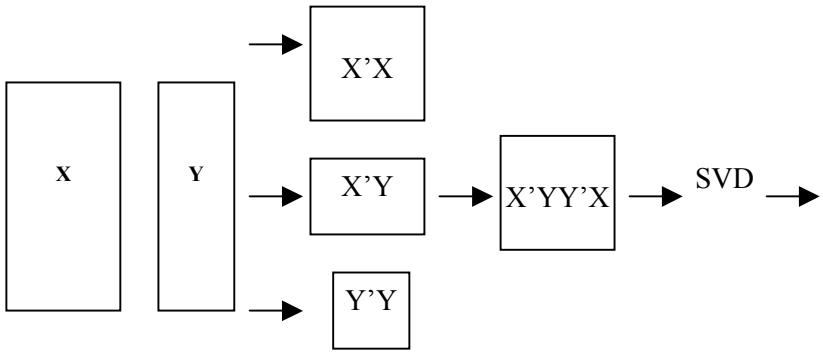

Figure 4. Basic overview of the Kernel matrixes in the Kernel Algorithm for PLS. The sizes of the matrixes are only dependent on the number of varialbes, not the number of samples (pixels).

```
function [W,P,Q,SX,SY,B]=kernelpls(XpX,XpY,YpY,A)
%KERNELPLS [W,P,Q,SX,SY,B]=kernelpls(XpX,XpY,YpY,A)
%  Calulates the PLS of X on Y using the Kernel approach.
%  INPUT:
%     XpX:  X'*X
%     XpY:  X'*Y
%     YpY:  Y'*Y
%     A: Scalar   number of components
%  OUTPUT:
%     W: (k by A) loading wights
%     P: (k by A) X-loadings
%     Q: (j by A) Y-loadings
%     SX:   (j by A) Explained X-variance
%     SY:   (j by A) Explained Y-variance
%     B: (k by A) regression coefficients
%  (C)1999 Thorbjørn T Lied,
%  Ref: Lindgren, Geladi, Wold:"The Kernel Algorithm for PLS"

% Initialize return variables
W=zeros(0);
P=zeros(0);
Q=zeros(0);
SX=zeros(0);
SY=zeros(0);
B=zeros(0);
I=eye(size(XpX));        %Identity matrix used for uptates
Kernel=XpY*XpY';         %Kernel Matrix
oXpX=XpX;                %Saving the initial XpX
oSSX=trace(XpX);
oSSY=trace(YpY);
for i=1:A
   [u,s,v]=svd(Kernel);
   w=u(:,1);
   wXw=w'*XpX*w;
   scale=inv(diag((wXw))); %used to scale p and q
   p=(w'*XpX)'*scale;
   Q=(W'*XPY)'*SCALE;
   W=[W w]; Q=[Q,q];   P=[P,p];
   %Update kernel and covariances:
   Iwp=I-w*p';
   XpX    = Iwp'*XpX*Iwp;
   XpY    = Iwp'*XpY;
   Kernel = XpY*XpY';
   b=W*inv(P'*W)*Q';
   B=[B b];
   YpY=b'*oXpX*b;
   SX=[SX 100-trace(XpX)/oSSX*100];
   SY=[SY trace(YpY)/oSSY*100];
end
```

Program Listing 1. MATLAB code for the Kernel PLS [19]. MATLAB implementation: T. T. Lied.

As with PCA, the components of PLS are orthogonal. The all-important score plot interpretation features of MIA is hence also available in MIR. In addition to the t-t plots, MIR offers t-u plots, ŷ-y plots (predicted vs. measured y), Figure 5 and Figure 7, or any other combination of t, u, x, y and ŷ that may be of interest. The same data can also be visualized in scene space in either grey-level images or colour-composition RGB images, papers II-V.

Combining different data in plots and images can reveal valuable information about the model. Plotting $\hat{y}_i$ vs. $\hat{y}_j$ [1] will show the degree of difference between the two (Figure 6 and Figure 8). The more pixels not lying on the diagonal, the larger is the difference. Showing e.g. $\hat{y}_i$ and y together in an image can also reveal *where* the model is accurate, and where there are problems. This can be seen as a function of colours in a RGB image. Divergence from grey tones indicate over- or under prediction (Figure 10 to Figure 12). Figure 9 shows the (calibrated) Y-variance plot from the banana example used here (X=Figure 13 and Y=Figure 36, below).

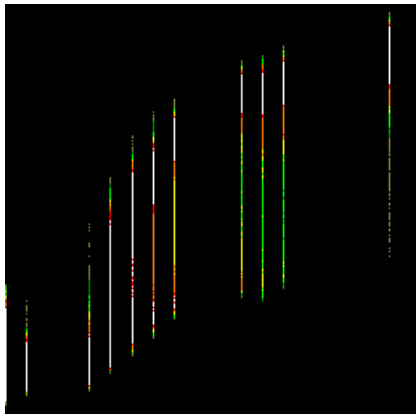

Figure 5.Pred$_1$ vs. measured ($\hat{y}_1$ vs. y)[1] from the banana case.

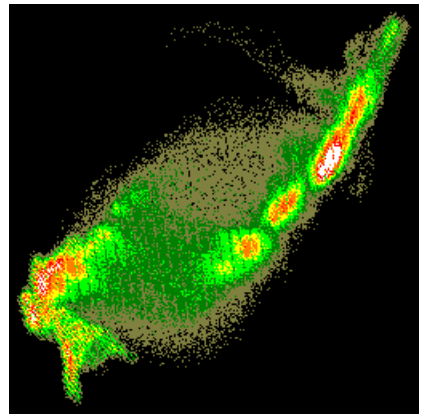

Figure 6.Cross plot of $\hat{y}_1$ vs. $\hat{y}_2$ [1] from the banana case.

[1] The subscripts i and j represents the number of components used to predict ŷ.

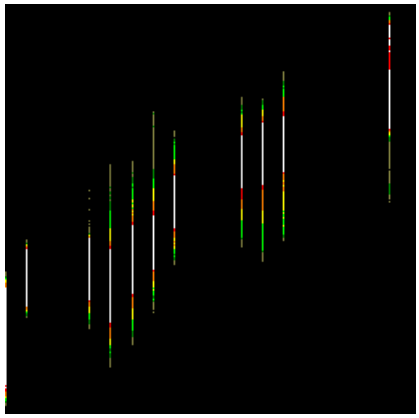

Figure 7.Pred$_2$ vs. measured ($\hat{y}_2$ vs. y) from the banana case.

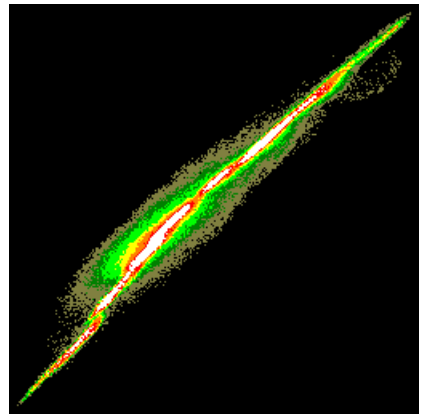

Figure 8.Cross plot of $\hat{y}_2$ vs. $\hat{y}_3$ from the banana case.

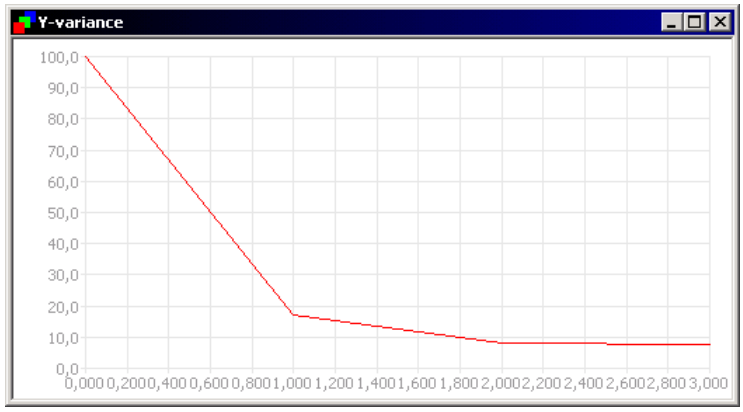

Figure 9.Residual Y-variance (cal) from the banana case.

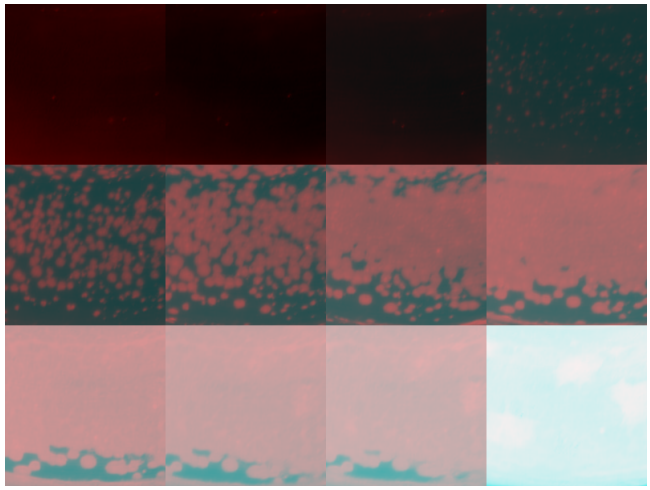

Figure 10. Predicted and measured ($\hat{y}_1$ and y) from the banana case. Red indicates over-prediction while cyan indicates under-prediction.

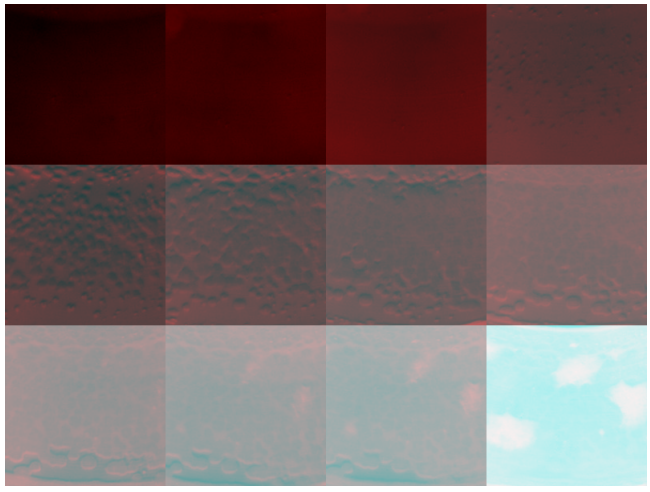

Figure 11. Predicted$_2$ and measured ($\hat{y}_2$ and y) from the banana case. Red indicates over-prediction while cyan indicates under-prediction

Within all the current figures (Figure 5 to Figure 12), except in the variance plot (Figure 9), the brushing technique also found in MIA can be applied to increase the interpretation potential even further. This will not be demonstrated here, however.

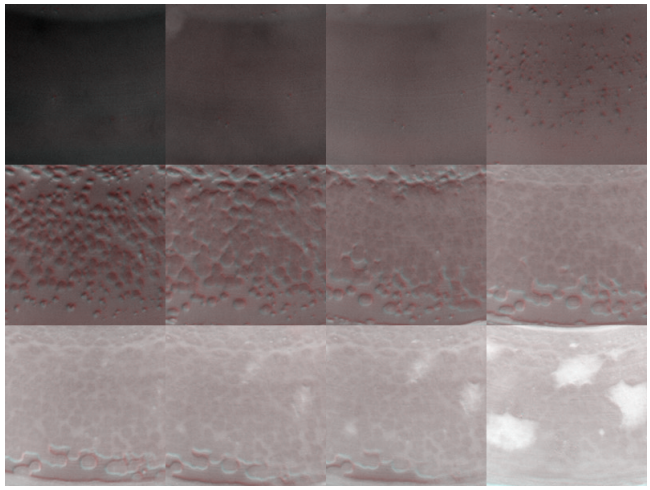

Figure 12. Predicted 3 and 2 ($\hat{y}_3$ and $\hat{y}_2$) from the banana case. Red indicates pixels where component 3 adds information, while component 2 is stronger where the pixels have cyan colour.

MIX: Data Pre-treatment

MIX, Multivariate Image teXture analysis, is an approach for introducing spatial information into the decomposition. Variables derived from the raw variables based on e.g. textural information, can be used as additional variables in the decomposition [21, 22, 23]. In cases where e.g. the spectral information in the raw data is insufficient for a calibration, textural derivatives may sometimes add the little extra needed in the calibration. In other situations more dedicated texture approaches may be needed.

With several variables available for derivation, and a large number of possible filters available for each, MIX has the potential for explosive data growth. Means for reducing this may come in handy when looking for an optimal solution. As a general approach, the following procedure is recommended:

• Decompose X with MIA

• Apply filters considered to be useful for extracting additional information either on the original X-images or in *score images* from MIA. Append variables to X, which is now termed X*.

• Calibrate X* with MIA or MIR (whichever appropriate for the objective at hand.)

• Examine loadings (from MIA) or loading weights (from MIR) to decide which (new or old) variables in X* should be used in the future, based on which variables actually helped *improve* the image modelling, relative to X alone.

The approach outlined above should be able to assist the filter selection procedure, as well as minimizing the amount of variables. Usually, textural derivates are quite different from their "parents", so using scores as the basis for filtration can often be a fast way of finding the "best" combination of filters. MIX is treated in papers II and III.

In paper V a somewhat different MIX-approach is suggested, where the results from MIR (see The Ŷ-Histogram section in the chapter on Quantitative Measurements, p. 39) are combined with the highly texture- and pattern correlated AMT-transformed spectra[24, 25, 26] in a PLS model containing both spectral and spatial data, MIR+AMT+PLS.

Data Arrangement

In traditional 2-way calibration, a calibration data set is created by concatenating a number of relevant samples into the X-matrix. For PLS the Y-vector or matrix is established the same way. To calibrate a reliable PLS model, it is of importance that all the different phenomena in X and in Y are well represented. The data should span the model as much as possible within the experimental domain.

This is the case also for MIA and MIR, only here also w.r.t. the image Region Of Interest (ROI). If the purpose of the calibration is to be able to extract some kind of information in future images, as many available images as possible should be merged into a larger calibration X-image, in e.g. an image-grid as described in detail in paper III. Figure 13 shows an example of a calibration image-grid.

Only if the entire ROI ("experimental design") is included in a single scene, making such image-grids is unnecessary. This situation is usually found only in remote sensing applications, but here it may also be useful to create a grid where different objects, e.g. water, snow, ice, forest, sand etc. are isolated in separate grid-cells for the training data set.

The Y-image

The Y-image used in a PLS calibration situation should be designed so that the predicted images from future X-observations can be used to extract the sought information from X. The nature of Y is thus very much *problem-dependent*. Paper III discusses three different principal modes of Y-images. These are in the paper called $\mathbf{Y}_{\mathbf{discrim}}$, $\mathbf{Y}_{\mathbf{grid}}$ and $\mathbf{Y}_{\mathbf{total}}$. Figure 14 shows a schematic overview of these three modes. Y-image examples are shown in Figure 30 (p.32) and in Figure 36 (p. 35).

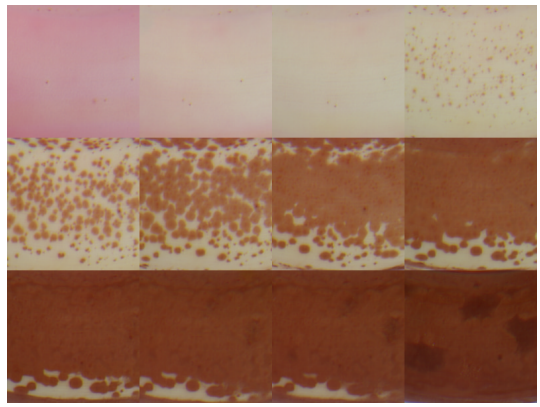

Figure 13. Example of image-grid with 4 by 3 sub-images. The dataset is treated in paper III.

The Y-image must be of identical spatial dimensions as X, but the number of variables may be different. If there are two or more variables in Y, the Kernel PLS[19] will act as a PLS2[17] algorithm.

In a quantification or classification system, different classes in X should be given different values in Y. Because PLS1 usually gives better precision for separate classes than PLS2 [18], one PLS1 model will often be created for each class. In the training data set, all occurrences of the class in focus are given high values (white) in Y, all other classes are given low value (black) in Y. When predicting X-images in the future, the current class will be brightened, while other classes will be darkened. In paper III this is called $\mathbf{Y}_{\mathbf{discrim}}$. This contrast adjustment can later be used for further feature extraction, paper V.

In designing the training set for a quantification system involving mixing of different classes, it is recommended to create a image-grid where the sub-images are of pure classes. Then the MIR algorithm will be able more optimally to separate the class of interest.

In homogenous mixtures, where e.g. the colour is varying with time, concentration etc., the training set should also span a number of concentrations in form of a grid, and the corresponding Y-image-grid should contain these concentrations in the corresponding grid-cells. This situation is called $\mathbf{Y}_{\mathbf{grid}}$ in paper III.

The $\mathbf{Y}_{\mathbf{total}}$ mode is the situation where every pixel in X has an individual representation in Y. in remote sensing applications this would correspond to complete ground-knowledge, down to every pixel. Also when creating models between different imaging techniques, the Y_{total} mode is relevant.

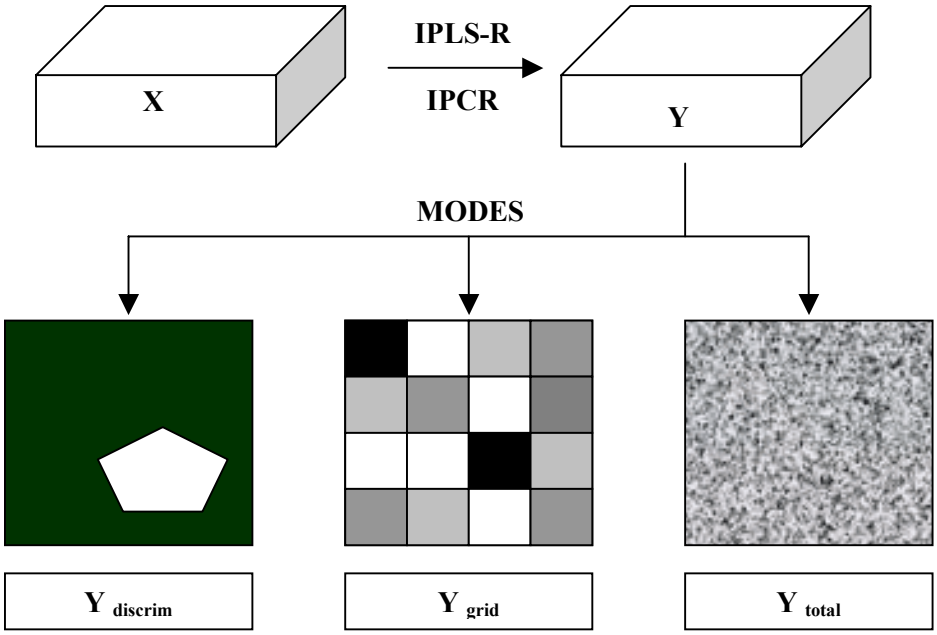

Figure 14. The three different Y-image modes: $Y_{discrim}$, Y_{grid} and Y_{total}. In all three modes additional Y-variables can be used in a MIR PLS2 case.

Interactive Prototype Software Implementation

Introduction

At the time I started studying the field of MIA and MIR, the (to me) only available PC software was the DOS-based ERDAS v. 7.5 (1991), where the MIA components were originally programmed by P. Geladi. This had to be run on an old computer (486) because of a NUMBER 9 CPU-card with capture- and display functionality. The software was slow when calculating, brushing and displaying data. There were no simultaneous display of images and plots, they all had to be visualized sequentially, once at a time. This system would in the long run lead to nothing but irritation, knowing of the potential in a Windows-based system. Therefore it was decided that in order to work efficiently with the data and methods, a different system was required.

Design specifications

Some design specifications were initially set up for the software system prior to any programming work. Based on the experiences with the ERDAS software, it was decided that the software should be designed for operation in MS Windows with the possibilities of displaying an "*unlimited*" number of images and plot simultaneously.

Further it was a requirement that the brushing technique should work both ways, not only in score space as was the case in ERDAS. Visualization of X, Y, T, U, and $\hat{Y}$ should be possible in both scene space and score space, and it should be possible to combine the different matrixes in plots and images.

Limitations

Because of limitations in time and manpower, some restrictions were made regarding the development of the prototype. The algorithms to be implemented were confined to PCA and Kernel-PLS, and they should both be available both in the global- as well as in the local-model mode. These limitations also forced the selection of a Rapid Development System with proper pre-made functionality for image data disk- and memory management. This was found in LabVIEW from National Instrument and their IMAQ Vision toolbox where most of the required tools were available.

In the present thesis it was not found time to optimise the software with respect of speed and memory consumption, but for small data sets (few variables) it is usually still reasonably fast. LabVIEW does not (for the time being, anyway) offer the

memory-management functionalities available in e.g. C++. Hence, the software uses a "large" amount of time and memory when doing calculation. Especially cross-validation on large data sets may tend to be rather slow.

Functionality

Figure 15 shows the UI-dialog for the implemented prototype.

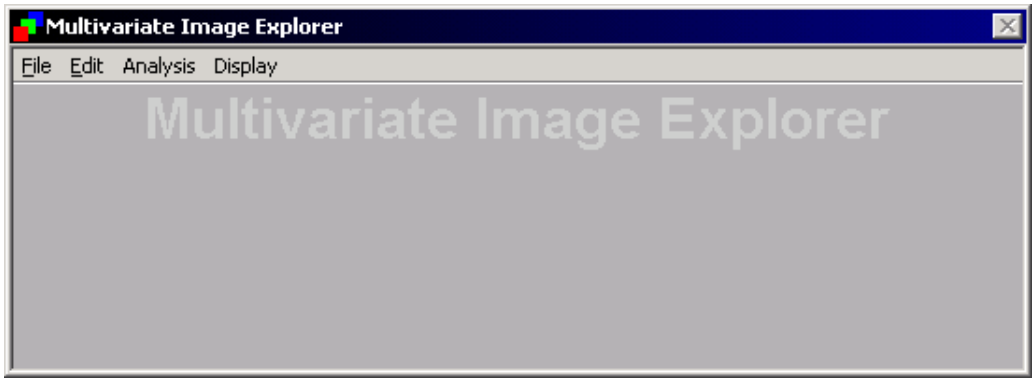

Figure 15. Main UI Dialog Window for the Multivariate Image Explorer prototype software.

The File Menu

At the present time, only BITMAP (*.BMP) images are supported, so images in other formats have to be converted to BITMAP before read into the system. If reading 24-bit bmp-files, however, these are automatically split into three colour channels.

Any number of channels may be read sequentially into both X and Y, and these are kept as 8-bit (unsigned integer) channels in memory until computations of correlation matrixes and loadings, which are computed in 64-bit double precision floating values.

For convenience, all loaded images can be stored together in one file. This is an internal file format containing X and Y channels in vectors, as well as raw data dimensions. The size of this file is equal to the sum of sizes of all images loaded. Later, when working on the same dataset, the entire dataset can be read in one operation.

As with data, MIA and MIR results can be stored to disk. Both scores, loadings, predicted images, X-and Y variance, loadings and loading weights are stored in one file. If the calibration is time-consuming, saving the results can save time if the user needs to study the results later. Also, when predicting new images, this is done from a model on file (currently not available in the program).

The Edit Menu

The active image can be copied to the clipboard and later pasted into a different program, e.g. a word-proseccor, software for image treatment etc.

The Analysis Menu

The Analysis Menu contains five choises: MIA (Image PCA), MIR (Image PLS), Local MIA, Local MIR and Cross validation. The local models, if selected, is based on a mask drawn in either score- or image space.

The Display Menu

This has 11 choises: Image, Cross-plot, Loading plots (includes loading weights), RMSEP-, PRESS- and CV-plot, X- and Y-variance, close (all open) windows, ROI (brushing) Tools and finally Select Fill Colour (the colour used for brushing in RGB images).

The Display Image Dialog (Figure 16) and the Display Cross-plot Dialog (Figure 17) alows the user to select among the avaliable matrixes and variables for what to plot.

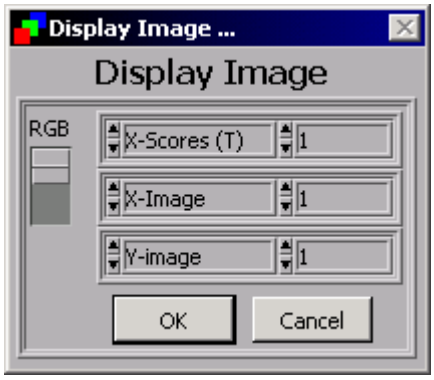

Figure 16. The Display Image Dialog allows the user to select among several data to show in image space.

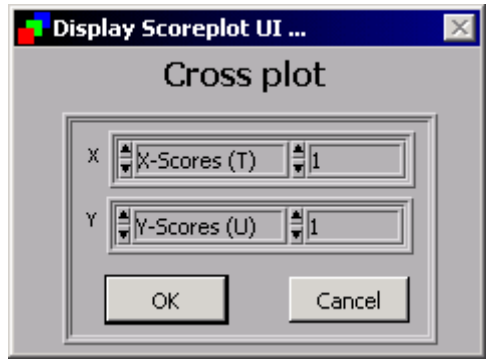

Figure 17. The Display Cross Plot Dialog allows the user to select what data to plot in x- and y-direction in the cross plot.

Explorative Analysis of Multivariate Images (MIA)

Before starting out on a regression calibration, an explorative analysis of the multivariate images using MIA usually gives valuable information about the structure in the data. If MIA easily separates the classes in X, there is probably no need for additional variables, and the regression model should be stable. If there are problems, however, MIX and other additional features can be of use.

The topics discussed in this chapter are treated extensively in paper I. In this outline, a different data set will be used to show the principles however. This dataset forms an integral part of paper V.

Figure 18 shows the raw image of the current example. This image is acquired with a SILVACAM video camera, which is a false colour composite NIR-R-G camera. The image represents a mix of three types of vegetables; green peas, maize and carrots. The image represents a mixture which contain approximately one third of each component.

Figure 18. RAW image of a three-component mix in NIR, Red and Green channels

Two MIA score plots from this classification problem is shown in Figure 19 and Figure 20. In this example only three channels were used, so there are three components available of score plots. In the MIA decomposition of this example, component three has the strongest discriminative effect. The first principal component is mainly used to describe the black-white contrast properties of the image. A small bright dot in the left part of the scoreplot (low t_1 score values) in Figure 19 represents the black border in the right and lower part of the image in Figure 18. The right side of the score plot (high t_1 score values) contains pixels with high values, e.g. reflections. Score 1 represents the overall intensity contrast of the image in the current example.

Such a general content-component will often be found when decomposing multivariate images.

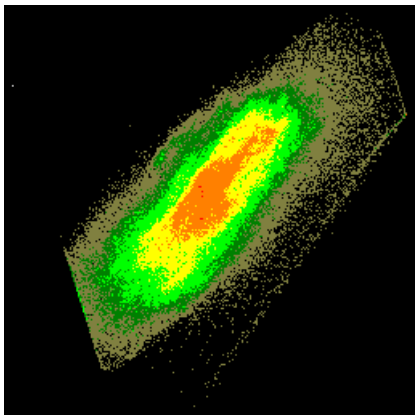

Figure 19. t_1 vs. t_2 scoreplot. Note that there is no obvious differentiation between the three species in this plot.

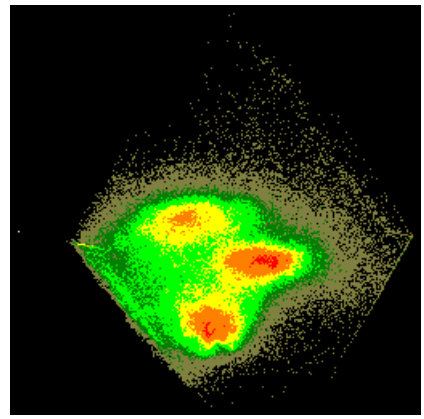

Figure 20. t_1 vs. t_3 score plot. Note good separation between the three species along the third image PC.

In Figure 21 the lower class in the t_1-t_3 score plot has been masked with a ROI. This mask has been projected to image space in Figure 22 with brushing. Here, it can be seen that the class outlined is the carrot class. The two other classes can be selected in the same manner. The middle class represents the maize (Figure 23) while the upper class (high t_3 values) represents the peas (Figure 24).

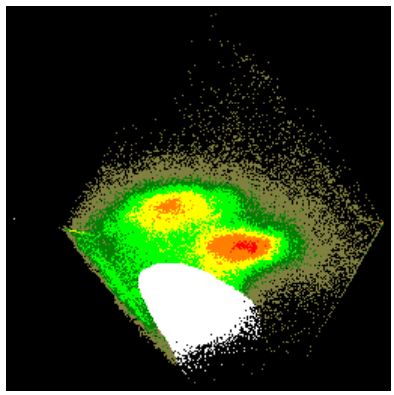

Figure 21. t_1 vs. t_3 score plot. The carrot class has been marked by a ROI. The corresponding mask in image space is shown in Figure 22.

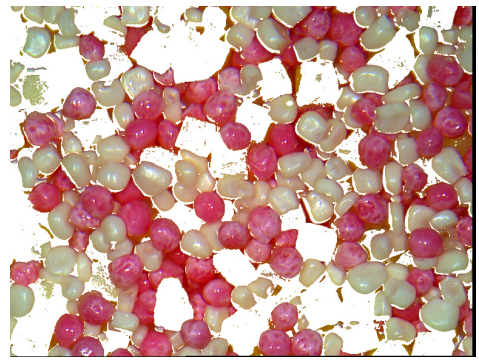

Figure 22. The carrot class selected in score space in Figure 21 shown in image space.

According to the separative effect of the third component in this example, it can be concluded that additional variables are not required for regression purposes. This is in agreement with what could be expected, considering the spectral differences between

the classes. In cases where the spectral distinctions are smaller, pre-processing etc. may be required.

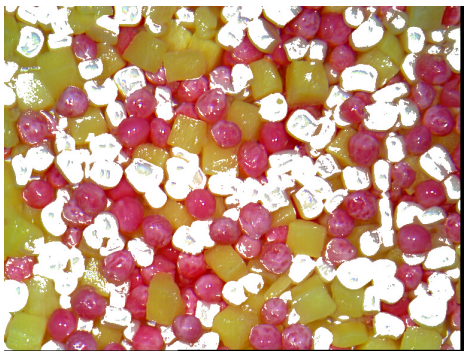

Figure 23. The maize-class selected in a t_1 - t_3 score plot shown in image space.

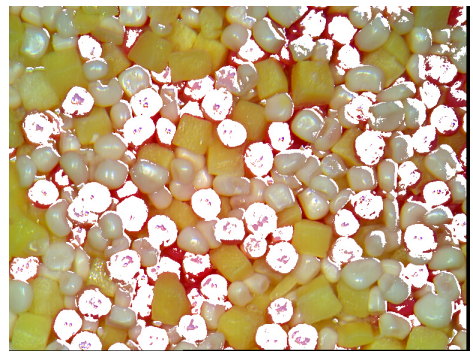

Figure 24. The pea-class selected in a t_1 - t_3 score plot shown in image space.

In addition to score plot inspection, plotting the loadings can give valuable information about the channels (i.e. the variables) and how they are used to build the scores. Below the loading plots corresponding to Figure 19 and Figure 20 are shown in Figure 25 and Figure 26 respectively.

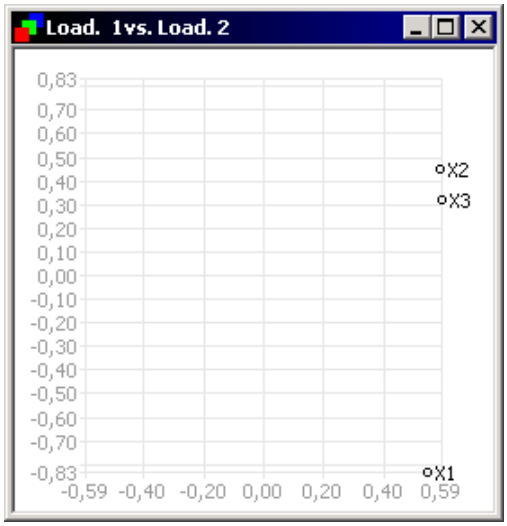

Figure 25. The p_1 vs. p_2 loading plot for the example in Figure 18. Note almost equal loading for all components in the first component, the general contrast-component.

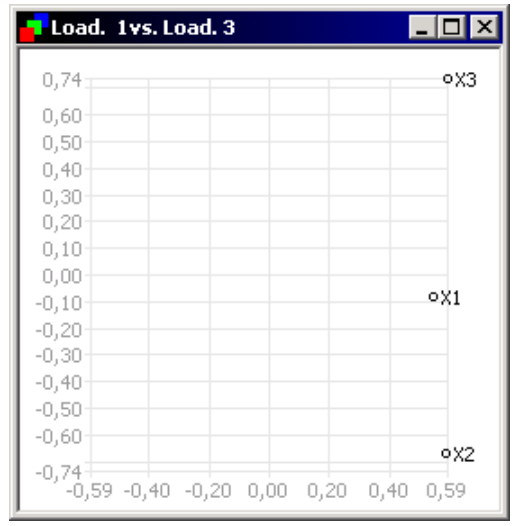

Figure 26. p_1 vs. p_3 loading plot. Note that X_1 lies close to zero on the third component, hence it is not used to seperate the classes in this example.

Figure 26 tells that variable 1 has a very small loading in the third component. Since it was the third component that was used to separate the classes, this means that variable 1 (the NIR channel) does not contribute to the classification. This can be understood

looking at variable 1 in image space, as shown in Figure 27, where it is obvious that the NIR channel does not separate the three different classes in any significant degree. In the current example, using the available NIR-channel to classify the three vegetables is a waste, and different sensors should be considered. A traditional RGB camera would probably be more suitable in the current case. By way of contrast, the PC3 is seen to be a manifestation of X3/X2, i.e. a green/red differentiation, Figure 26.

Figure 27. Variable 1, the NIR-channel in the example from Figure 18.

For assisting the interpretation of the score plots, it is valuable to also look at the score images. A score image is a score component visualized ("backfolded") in scene space. Figure 28 shows component 1 while Figure 29 shows component 3. Note how the first component is used both to describe the overall greylevel span of the image and to describe the 3D properties of the vegetables with highlights and shadows, while the third component describes an almost flat landscape, but where the three classes have highly discriminatory values. The essence of MIA comes about only if/when it is understood that the score plot t1-t3 (Figure 20) comes about by plotting Figure 28 against Figure 29.

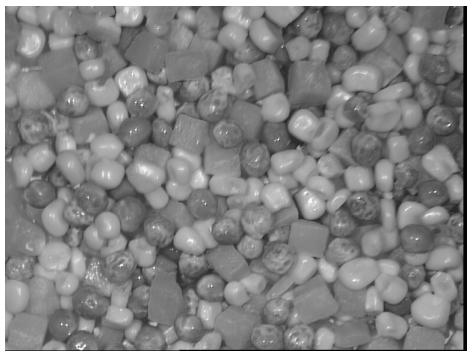

Figure 28. Score image of component 1.

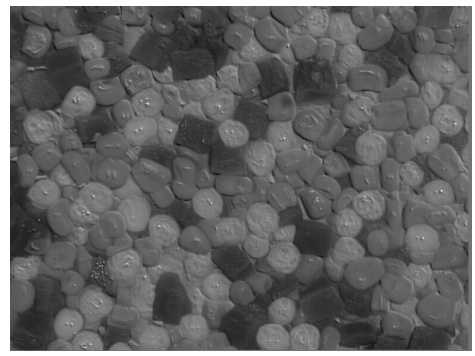

Figure 29. Score image of component 3.

Multivariate Image Regression (MIR)

As mentioned above, MIR involves building a regression model between two sets of image data. There may be several reasons for doing this[4]:

- Matching one imaging technique with another
- Matching remotely sensed images with "ground truth"
- Matching images in one spectral region with images in a different spectral region, e.g. UV and NIR.
- Classification for quantitative measurements
- Quantitative <y-image prediction
- Other, more problem-dependent, regression situations (generic).

To illustrate a typical quantitative measurement system, the example used in Figure 18, the three vegetables, will be used. For in-depth MIR reading, see papers II and III and for more examples see paper V.

In a production plant involved with heterogeneous mixtures of particles, it is important to be able to measure the fraction of each component in the mixture at different production stages. Because of segregation and other problems occurring when mixing particulate matter of different physical characteristics, being able to compare the mix with its specifications is of critical interest to the producer.

In this section calibration models will be created from training data sets, and these will be examined. In the section on Quantitative Measurements (p. 37) the results from the following example chosen here for illustration, will be used for calibrating the concentrations of the different species.

Figure 30 gives an overview of the principles involved in setting up a MIR regression model for quantitative image measurements of a heterogeneous mixture.

The principles visualized in Figure 30 have been designed to enhance the contrast in the predicted Ŷ-images from mixture images. In the illustration a X-training image is constructed from pure class-images of the classes involved, here named A, B and C. For each class an Y-image is generated which will be used to maximize the intensity *difference* between the pertinent class and its surroundings.

In a PLS1 MIR situation, each class will be represented with an individual Y-image. This has the maximum value (white) in every pixel in Y where the current class is

found in X, and the minimum value (black) in all other pixels. For an unsigned, 8-bit image these values are 255 (white) and 0 (black), respectively.

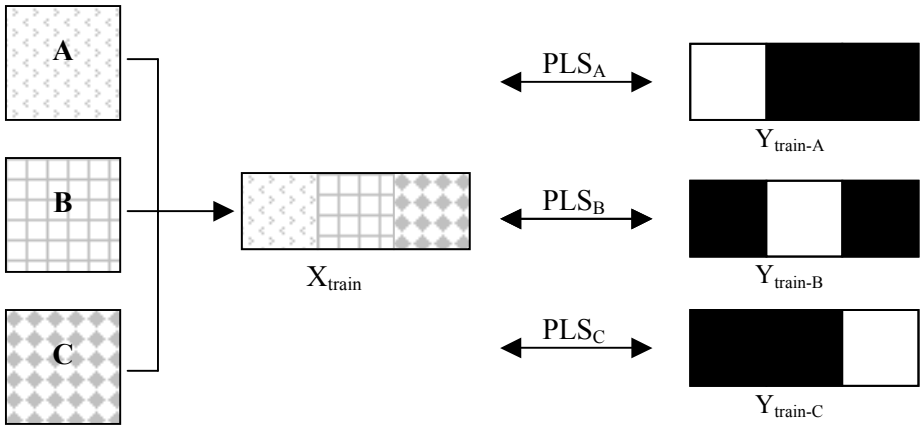

Figure 30. Illustration of MIR discrim set-up for quantitative measurements of classes A, B and C in heterogeneous mixtures. A separate model is calibrated for each class.

In this illustration, three models are created, one for each class. Even though it is possible to use PLS2 to create one general model for all classes, prediction will usually be improved by using separate models [18].

Figure 31 shows the X-training image for the vegetables example. Note that the three classes are kept in separate sub-images (ref Y_{grid}, paper III). This has been done to ease the Y-image construction. If it is not possible to acquire training images of pure classes, generating the Y will be more complex.

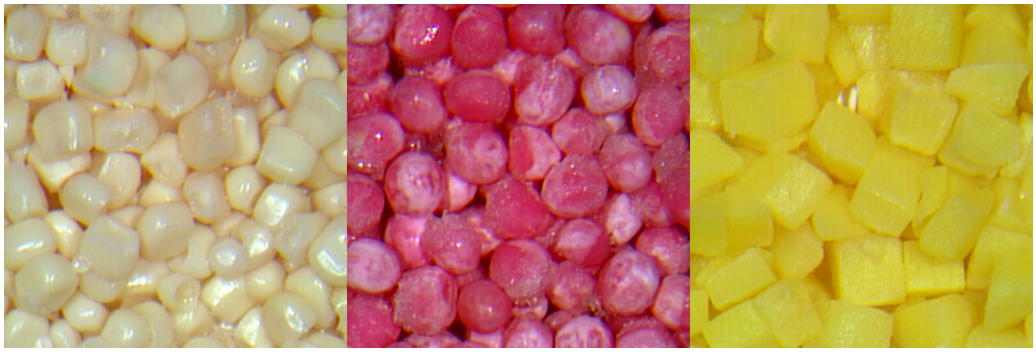

Figure 31. X-training image for the mixed vegetables example. From left to right: maize, peas and carrots. The SILVACAM false colour camera was used, thus the "strange" colours.

The corresponding Y-images that should be used for the three separate models are equal to the ones illustrated in Figure 30, and is hence not repeated. The corresponding t_1-t_2 scoreplot for each model is visualized in Figure 32. From this figure it is evident that the maize-model has more problems separating the three classes than the two other

models. This can also be seen in Figure 33, Figure 34 and Figure 35 where the predicted images are shown.

Also notice that the carrot model only uses one component to distinguish the classes, while the other models use two. This can be seen in Figure 32, where the separation in the rightmost score plot is done in the horizontal direction, in contrast to the middle and leftmost models which use a diagonal (combination of 1st and 2nd component).

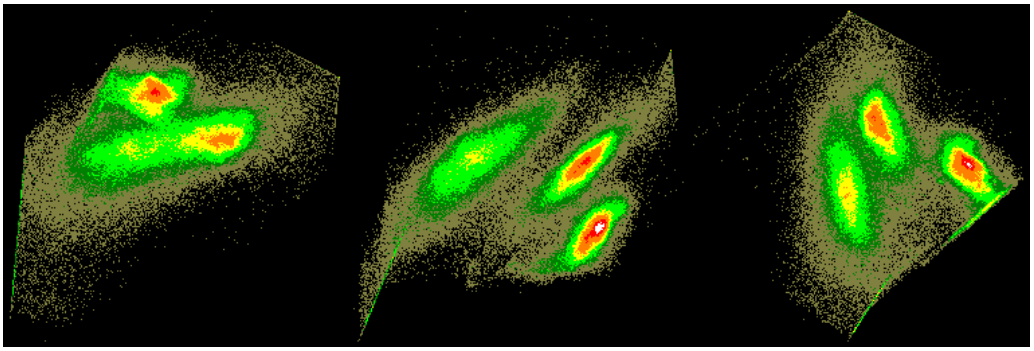

Figure 32. t_1 vs. t_2 for the three vegetable models designed to predict, from left to right, maize, peas and carrots.

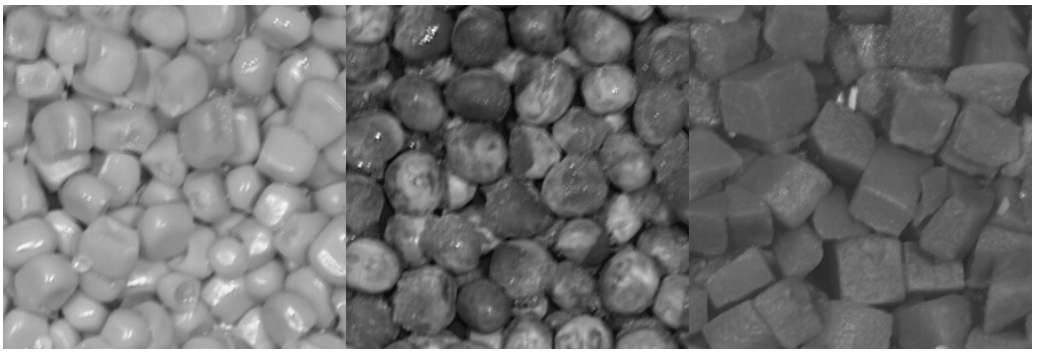

Figure 33. Predicted image from maize- model using 2 components. Note the similarities between maize and some of the peas (which are frost covered).

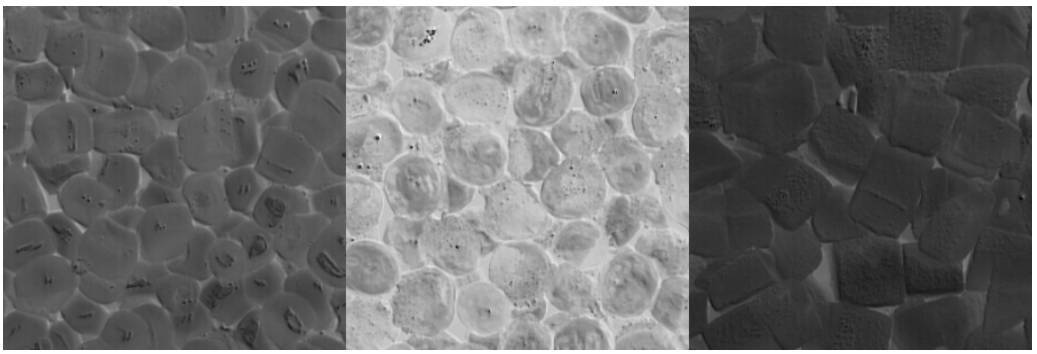

Figure 34. Predicted image from pea-model using 2 components. Note good separation between peas and the other classes. Also note the (somewhat smaller) difference between maize and carrots.

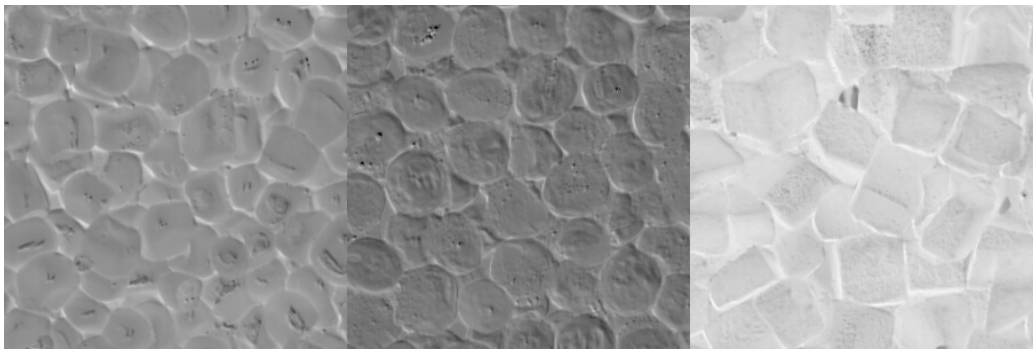

Figure 35. Predicted image from carrot-model using 1 component. Note good separation between carrots and the others, also note differences between maize and peas.

Figure 34 and Figure 35 should be compared with Figure 29 (the 3rd score image of the MIA example). This comparison should demonstrate that MIR, in the last two models, successfully has guided the decomposition against a classification, and is no longer concerned with variations in intensity (reflections, shadows etc.) as was the case in the first MIA components.

The images shown in Figure 33 - Figure 35 is not the answer to the question "how much of each class". It is merely a *pre-processing-step* on the way to this answer. How these images are treated further, will be discussed in the chapter Quantitative Measurements (p. 37).

In the outline above a heterogeneous mixture example has been used for illustration. Because of the obvious benefits regarding complete acquisition control and "ground truth knowledge", these examples are easy to set up. It should be noted, however, that the *principles* above may very well be transferred to remote sensing problems involving satellite imagery, as these images also can be considered of being of the *heterogeneous mixture* type as well as many other similar types of the same nature.

In cases *not* involving heterogeneous mixtures, different approaches may be needed for setting up a MIR calibration. If the measured signals show a shift in intensity or colour as a result of variations in the medium being imaged, a more continuous model involving more Y-value levels will be required, ref Y_{grid} and Y_{total} (paper III).

For an example involving a more complex model, refer to Figure 13 which shows the banana example X-training image from paper III. This example treats the problem regarding deciding the degradation status of bananas as a function of time. In the corresponding Y-image, each sub image contains the storage time in number of days since purchase. Hence this Y-image has 12 different value levels, as shown in Figure 36.

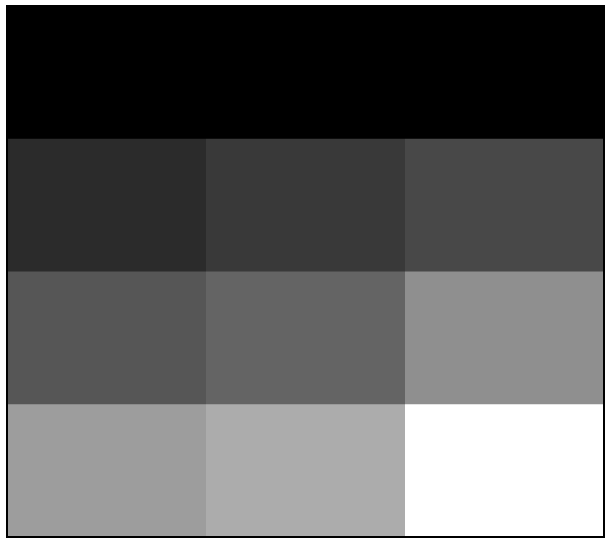

Figure 36. The Y image of the banana problem shown in Figure 13.In this Y-image, the different sub-image greylevels correspond to the number of days the banana has been stored. The brighter value, the longer storage period.

In the current image (Figure 36), the grey level values have been maximized in contrast ("contrast-stretched") to enhance visual inspection. Because the original image contains values between 1 and 20, it would not be possible to distinguish the different sub-images by visual inspection..

Figure 37 shows the t_1-t_2 scoreplot for the banana example, which clearly has a complex structure. This highly *non-linear* model obviously needs some help to be able to predict the degradation as a function of time. This is detailed in paper III.

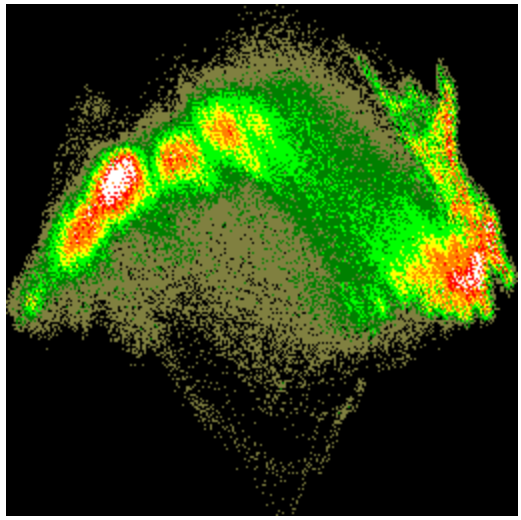

Figure 37. t_1-t_2 score plot from the banana MIR example.

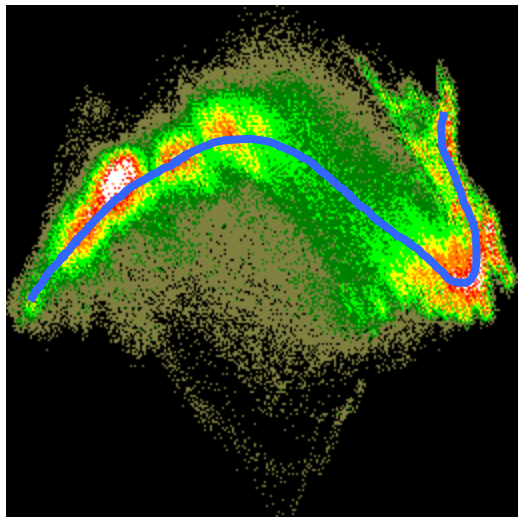

Figure 38. The t_1-t_2 scoreplot from the banana MIR example. The curve follows the time-line(from right to left) of steadily increasing decomposition over 20 days.

In Figure 38 the blue curve marks the time-line in the banana example in the t_1-t_2 score plot, which shows a highly non-linear trace. Considering there are actually two processes involved, this is not difficult to understand. Initially the banana is going through a ripening process where the colour changes from green to yellow. This process is followed by a deterioration process where the colour changes from yellow to dark brown.

Quantitative Measurements: Extended MIR-predictions

For quantitative measurement problems, especially when dealing with homogeneous mixtures and remote sensing problems, MIR can be used as a powerful pre-processing step which essentially *enhances* intensity differences among classes. However, trying to predict one value directly from a complex image will seldom be effective. In addition to MIR, some sort of post-processing must also often be applied. In the current chapter three different pre-processing approaches will be presented; two univariate and one multivariate. These methods are discussed in detail in paper V.

The basis for the current discussion is the predicted image, as shown in Figure 33 - Figure 35. As mentioned in the previous chapter, these figures show that each class is associated with a distinct grey-level interval. In a mixture analysis, the number of pixels in each interval should hopefully correlate with the concentration of the corresponding class in some way. Three different approaches for this correlation will be presented below.

Thresholding

The thresholding technique tries to find the optimum grey-level value that will split the predicted image in two, where the modelled class will get the value one, and all other classes will get the value zero. If this is done successfully, calculating the mean value for the thresholded image will return the concentration as a fraction between 0 and 1. This is thus a very direct method for calculating an estimate of the concentration.

One problem with this method is finding the "best" value to threshold at. For assistance in selecting this value, the histogram-plot[12] of the predicted image can be valuable. Figure 39 shows the histogram-plot of the predicted image in Figure 35. In the histogram-plot, grey-level 180 was selected to be optimal. Figure 40 shows Figure 35 after thresholding, and the listing below shows both the thresholding command in MATLAB, as well as calculation of the mean value.

```
» BW=im2bw(A,180/255);
» f=mean(BW(:))
f = 0.3367
```

Program Listing 2. Thresholding of image A at 180, and calculation of mean value.

As can be seen from Program Listing 2, the concentration of the current class is found to be 0.3367, or 33.67%. Because the class covers exactly 1/3 of the image, the correct

value would be 0.3333, or 33.33%. For the current example, this must be said to be within an acceptable prediction error.

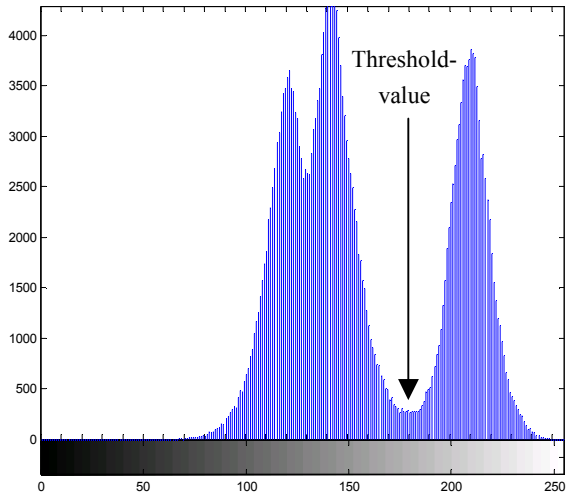

Figure 39. The histogram-plot of Figure 35 (predicted image using the carrot-model) is a valuable tool for selecting the optimum point for thresholding. The histogram is an overview of the number of pixels at each grey-level value.In the current image the local minimum at about 180 was selected. Figure 40 shows the result after splitting at this level.

When it comes to the other two classes, however, there is a strong overlap, as can be seen in the histogram in Figure 39. For these classes, finding split-points would be difficult, to say the least. For these classes, the two other models must be used.

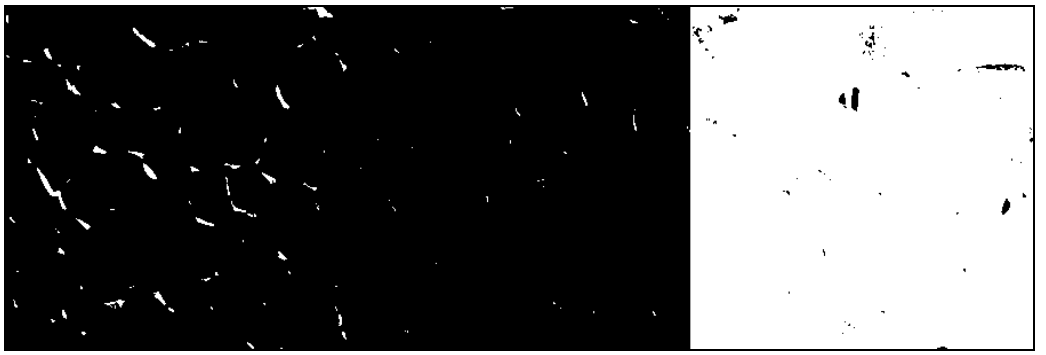

Figure 40. Predicted image using the carrot model in Figure 35 after thresholding with "optimum" value.

Figure 40 shows that some parts of the class are not included (they are left black), while on the other hand, parts of the other classes have been included in the class. This phenomenon will almost always occur, and is mostly related to highlights and shadow effects. Unless totally eliminated with large, diffuse light sources, this can give problems in the modelling. Also consider the predicted image using the maize-model

(Figure 33). This model has great difficulties separating between maize and peas, as can be seen in Figure 41. In this case, the thresholding approach will be problematic.

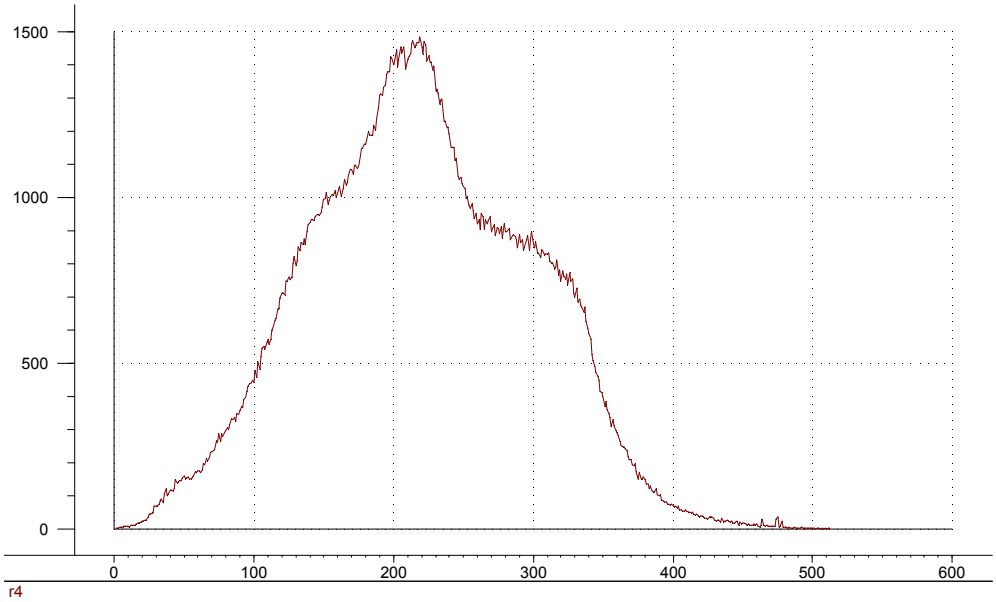

Figure 41. Histogram of predicted image using the maize-model. The image contains 1/3 of each vegetable. Finding the optimum threshold value is obviously very difficult, and large prediction errors are to be expected.

The mean value

Instead of thresholding the predicted image, the mean value can be calculated for the predicted image "as is". This will in some cases reduce the influence of noise in the image, averaging the effects from highlights and shadows etc. Especially when dealing with a two-component mix this is rather straight forward. If three or more classes are involved, each with a separate grey-level interval, correlating the mean value with concentration becomes somewhat more complicated, quickly becoming useless.

The Ŷ-Histogram

Because the histogram of the predicted image can be seen as a *spectrum* in which the bin heights correlate with the concentration of classes associated with the different positions, the histogram can be used as X-data, in principle in the same way as e.g. a NIR-spectrum in traditional PLS calibration[17]. Figure 42 shows three histograms of pure classes predicted with the same model. Figure 43 shows the histogram for a 1/3 + 1/3 + 1/3 mix.

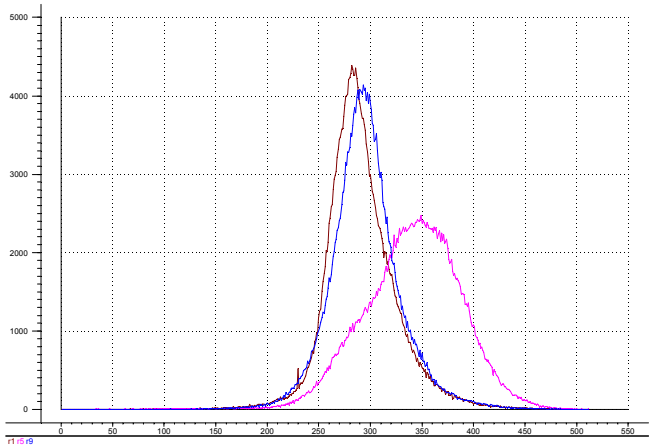

Figure 42. Histogram for maize, peas and carrots using the carrot model. Note the strong overlap between the two leftmost (non-carrot) classes.

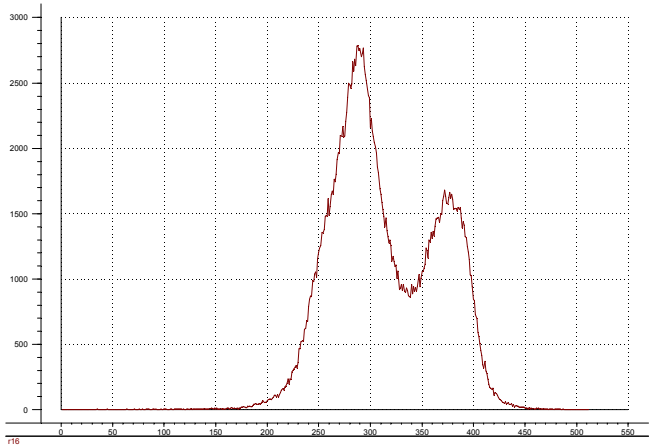

Figure 43. Histogram of predicted mixed image, 1/3 of each class. Carrot-model.

Figure 44 shows how MIR and 2-way PLS relates to the multivariate Image X, the predicted Ŷ-image and its histogram. The MIR model being used for Ŷ-prediction has been established earlier using the approach outlined in Figure 30. Likewise, the PLS-model used to predict the final value(s) has been established on the basis of a calibration set of several histograms with known y-values.

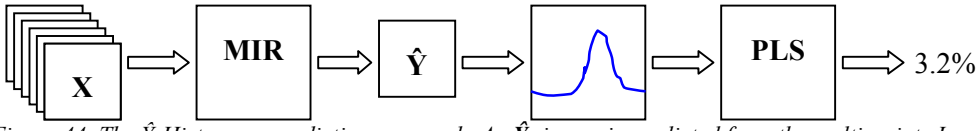

*Figure 44. The Ŷ-Histogram prediction approach. An **Ŷ** -image is predicted from the multivariate Image **X** using an existing MIR model. The histogram of the **Ŷ** -image is then used for prediction using a traditional 2-way PLS model.*

Examples

Using histograms as shown in Figure 42 and Figure 43 for PLS calibration against concentration is now straight forward. Below, in Figure 45 and Figure 46, is shown standard PLS calibration and cross-validation results when predicting the carrot concentration in carrot-predicted mixed images. The prediction error (RMSEP) is evaluated to be 4.76 measurement units [%] using one component. Results for the other vegetable classes predicted with representative models are shown in Figure 47- Figure 50. See paper V for results from different examples.

These prediction results pertain to our development of a generic image analysis-based mixing process monitoring facility, which is also directed towards other 2- and 3-component mixing systems in paper V.

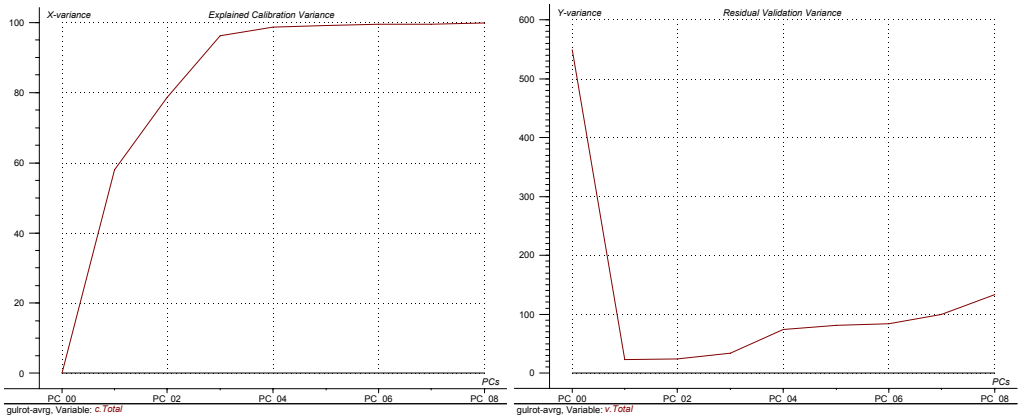

Figure 45 Explained X-variance (cal.) (left) and residual Y-variance (val.) (right). Carrot model.

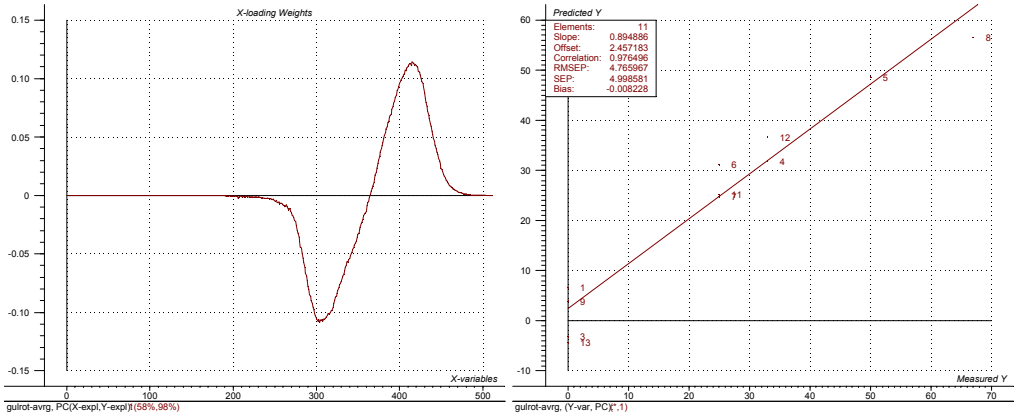

Figure 46. Loading Weights (1-3) (left) and Predicted vs. Measured (val.) (right) 1 comp. Carrot Model.

Table 1. Cross validation results of PLS Carrot-model on histograms from MIR-predicted Ŷ-images.

Carrot-Model				
# Comp	Slope	Offset	Correlation	RMSEP
1	0.895	2.457	0.976	4.766

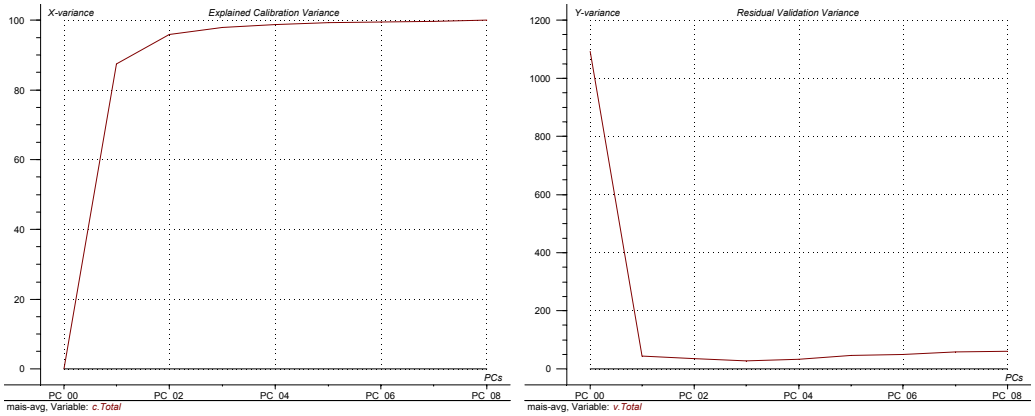

Figure 47 Explained X-variance (cal.) (left) and residual Y-variance (val.) (right). Maize model.

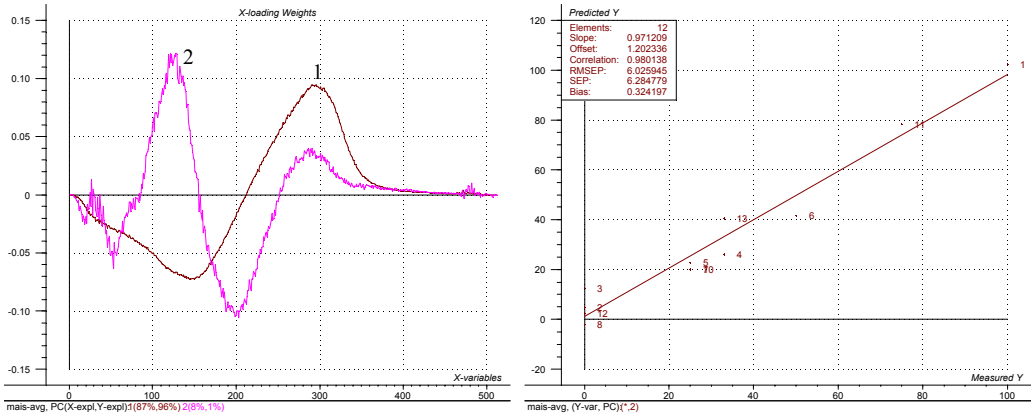

Figure 48. Loading Weights (1-3) (left) and Predicted vs. Measured (val.) (right), 2 comp. Maize Model.

Table 2. Cross validation results of PLS Maize-model on histograms from MIR-predicted Ŷ-images.

Maize-Model				
# Comp	Slope	Offset	Correlation	RMSEP
2	0.971	1.202	0.980	6.026

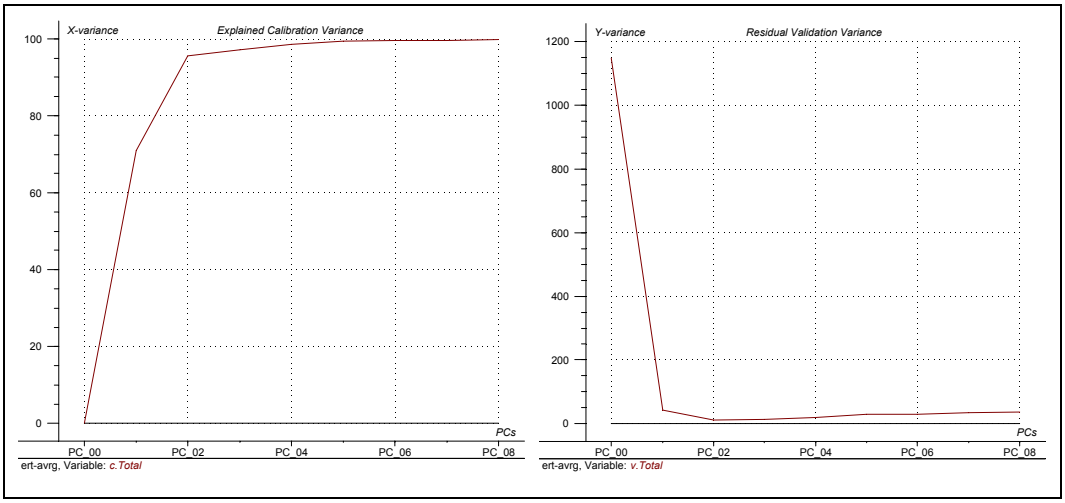

Figure 49 Explained X-variance (cal.) (left) and residual Y-variance (val.) (right). Pea model.

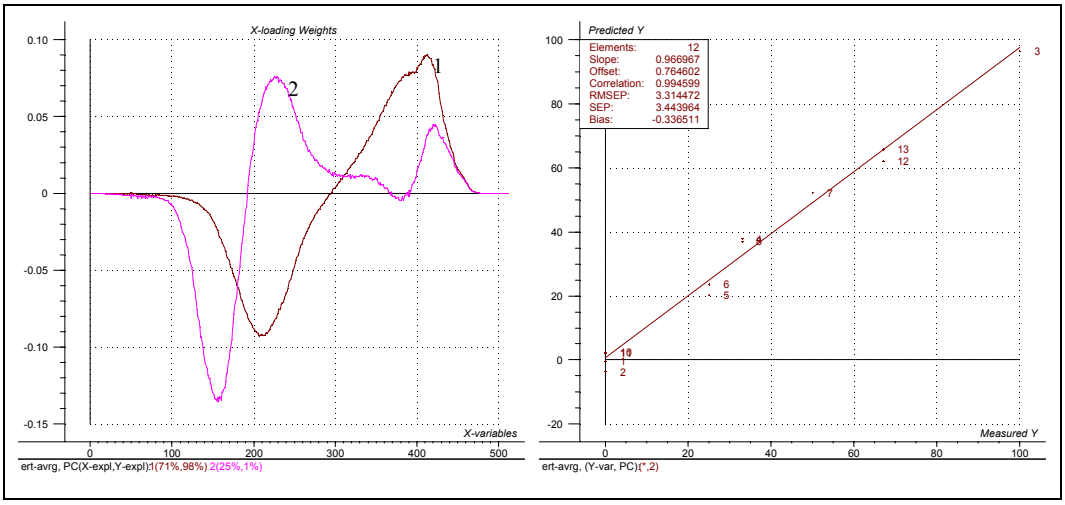

Figure 50. Loading Weights (1-3) (left) and Predicted vs. Measured (val.) (right), 2 comp. Pea Model.

Table 3. Cross validation results of PLS Pea-model on histograms from MIR-predicted Ŷ-images.

Pea-Model				
# Comp	Slope	Offset	Correlation	RMSEP
2	0.967	0.765	0.995	3.315

Preliminary Conclusion

The results above (Table 1 to Table 3) as well as those reported in paper V must be said to be very promising for the future of MIR in image based measurement systems.

Using standard PLS on the histograms from MIR-predicted images should now be transferable to a large number of similar applications in which quantitative prediction is on the agenda.

Extending MIR with AMT

In some cases *combining* the MIR^+ histogram with AMT spectra makes it possible to combine spectral and spatial information in one calibration model in a very powerful manner. In the examples studied in paper V, only marginal improvements were found, and are hence not reported. The main reason for this is the already strong regression models developed using MIR^+ and AMT separately in the present examples. In future problems, however, this combination can prove to be very useful.

Multivariate Image Cross validation

In any multivariate model to be used for prediction, it is important to know the predicting powers of the model. This is usually done by estimating the prediction errors as a measure between known and predicted values. A popular prediction measure is RMSEP (Root Mean Square Error of Prediction) which is defined as

$$RMSEP = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_{y,ref})^2}{n}}$$

where $\hat{y}_i$ refers to the predicted value, and $y_{i,ref}$ is the known value [18].

The procedure of testing prediction performance is known as *validation*. To perform this optimally, at least two sets of data are required, one for calibration and one for validation. When a model has been established, using the calibration set, the validation set is subsequently used for predicting the ŷ-values of the validation set for comparison, e.g. according to equation 1.

At least two variations for this type of validation exist, one is known as "*test set validation"*, the other as *"cross validation"*. In test set validation, a completely new, independently sampled data set is acquired, in addition to the calibration set. This demands that an *identical sampling procedure* is used for *both* data sets [18].

If this is not feasible, a different, less optimal, approach will have to be resorted to. *Cross validation* extracts a *,* set from the calibration set before building the model on the remaining complement of data. The extracted data is now used for validation. This approach may take several different forms, but all are closely related, in that they must correspond to one specific number of so-called *segments* in the list: 2,3,4,5....N, where N stands for the total number of objects in the original calibration set. After prediction errors have been estimated for the one left-out segment, it is replaced in the modelling base and a new model is created in which a different segment is being kept out of the modelling etc. This is continued until every segment, and object, has been used for the validation, hence the term *cross* validation [27].

To get realistic validation estimates, it is important that the calibration and validation datasets represent two independent samplings from the target (parent) population. The degree of difference between them should reflect the variations that can be expected associated with the future measurement situation in which the regression model is to be

used for prediction purposes [18]. It is easy to see that test set validation is the only approach which honours all these requirements.

In 2-way chemometrics there are steadfast different opinions regarding how exactly to divide the data in calibration and cross validation sets or segments [17]. From so-called full cross validation (leave one out) on the one hand, to two-segment, so-called "test set switch" on the other; the latter represent a singularly unsatisfactory choice of terminology, as there is no "test set" present at all. It is always possible to use any intermediate number of segments from the list: 2,3,4 ....N. The relationships between test set validation and these systematics of cross validation remain an area of some confusion and intense debates in conventional 2-way multivariate calibration[18]. In multivariate image analysis, however, distinct and special considerations are required.

There are two major characteristics in image data that are rarely found in 2-way data. Most striking is the number of “objects”. In a conventional video image (~500x700 pixels), there are more than 350.000 “objects”, i.e. pixels, all in the range [0..255]. Removing any single object from this amount of data is not going to change the model adequately to perform any useful validation (to say the least!) [18]. Also, calculating 350.000 sub-models, full cross validation, is nonsensical.

Secondly, and much more important to consider, is the large *redundancy* that exists in image data. Pixels lying close together in the image space are likely to represent the same image-object, and therefore often have closely similar values. Two-segment data sets, for example in which every second pixel, is allocated for the training – and test sets respectively, would necessarily produce two almost identical images, clearly leading to inferior validation. This would correspond to some spatial (image space) segmentation scheme. With knowledge of object selection traditions in 2-way data analysis, the reader might well alternatively ask: “Why not simply use random sampling?” This would correspond to a notion of a fair "blind", automated segmentation strategy. Again, consider the very large amount of data (pixels) present. Sampling 50% randomly out of 350.000 objects would most likely again simply produce two practically identical datasets. This is where segmented cross-validation would be truly beneficial, based on e.g. 10% segments or similar.

Multivariate image analysis often requires reconsideration of the strategies employed for selecting relevant data sets for calibration and validation. A new a strategy called “*guided random sampling*” is suggested in paper IV. In guided random sampling the *user* decides how the data is to be divided into the pertinent sets. This is neither done randomly, nor by a pre-specified "blind" number of segments (representing e.g. 10% or

otherwise), but with very specific respect for the *empirical data covariance structure* present (in the score feature space). A different angle from which to attack the data segmentation problem is required. Following the MIA experience this angle is to be found in the score-space.

Paper IV discusses the topic of cross validation of multivariate images in pertinent detail. It is there shown that segmenting the data in calibration- and validation sets can be performed in score space. It is not, however, indifferent which score plots to use for the segmentation. Paper IV shows that in strongly correlated score plots, e.g. in the primary components, the t-u plots etc., the strong structure in the data will result in large differences between the different segmented sub-data sets. When validating, the model created with one type of data will try to predict an often very different type of data, giving non-representative validation results.

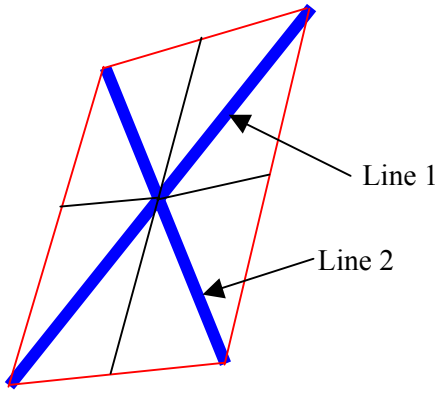

Figure 51. Two lines (1 and 2) are drawn in a score plot by ***the image analyst****. These lines are used as the skeleton of the Maltese Mask, which is calculated based on the two lines.*

In paper IV a new procedure for delineating a mask in score plots is introduced. By drawing two intersecting lines, a double Maltese Cross is generated (Figure 51), which can be used to segment the data in 2, 4 or 8 segments. In designing this mask, efforts have been made to ensure maximum spanning of the covariance structure of the data, hence when 2 or 4 segments are combined, they are always pair-wise opposite with respect of the centre (intersection) point of the mask, Figure 52 in the well-known form of two alternative Maltese Crosses .

The mask, when drawn in a suitable *high*-order score plot, will produce validation segments that span the data sufficiently representative, but still with some structural differences intact, the consequence being a realistic validation of the predictive capabilities of the model. Figure 52 shows the Maltese cross delineated in a high-order score plot (t_4-t_5) with corresponding image space mask projections. The data visualized here is presented in detail in paper IV.

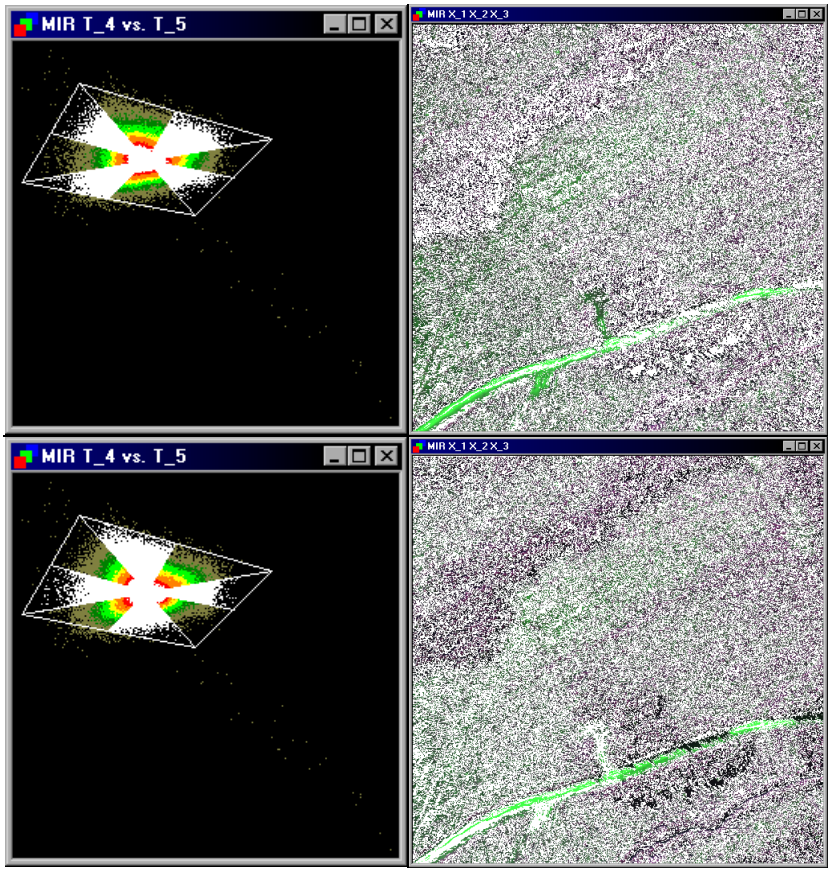

Figure 52. Maltese Cross delineated in a t_4-t_5 score plot with corresponding scene space image projections. The mask is shown in white in both score- and image spaces. Data details are found in paper IV.

Discussion

An image says more than a thousand words. For image-analytical measurement systems, these thousand words often needs to be reduced to a single, or a few quantitative values. To be able to do this, some means of *data reduction* and conceptual *reorganisation* of the objectives are often required.

Multivariate Image Analysis (MIA) has the power of reducing the original many channel multivariate images to a few bilinear components containing all the important structural information of the data (if *correctly validated* models are used only). This is the important data reduction characteristic, which has benefited chemometrics ever since its start almost 30 years ago.

Multivariate Image Regression based on PLS multivariate calibration (MIR/ MIR$^+$) can be *guided* to focus on the specific problem in question, and is thus often the most direct method for ultimately automated measurement systems (again if *correctly validated* models are used). It has been shown in this thesis, however, that neither of these methods can stand alone in every case.

When the Multivariate Image techniques presented in this work are applied to real-world, complex issues, there is sometimes also a requirement both for dedicated pre-processing (AMT, MIR$^+$) as well as post-processing. As an example, we have developed the MIR$^+$-approach (paper V), in which post-calibrating an initial MIR-model based on the histogram of the primary predicted image pixel values, have proven to be an effective method to quantify the "contents" of two- and three-component simple mixing images. This novel method, *extended MIR-modelling (MIR$^+$)*, should certainly be applicable to a much larger range of applications than the pilot-studies covered by this work.

It may perhaps be argued that the same type of histogram can be used also for calibration without going through an initial MIR for prediction, i.e. based on the original channels directly - which is a standard feature in classical image analysis. This may be the case in very simple situations, but it is no problem to demonstrate that it is downright impossible to find a *single spectral channel* which can be used to identify, classify and quantify the objects in even only moderately complex image data. As a general approach, the suggested new procedure should be much more reliable than working on raw data alone.

Relation to earlier work

Papers I - V constitute various further developments of MIA and MIR, based on the seminal works by Geladi and Esbensen in the late 80'ies/early 90'ies, followed by the fundamental Geladi & Grahn first textbook on MIA. These five papers gives a complete reference to all the works from these authors within chemometrics. The individual papers discuss these relationships in the pertinent detail.

There has been very little additional, external work within the realm of *multivariate image analysis*, as defined by the scope set out by the above three founders, since. I have tried, to the best of my ability, to give full credit to this in papers I - V. Note that the broad and important areas on image analysis within the field of remote sensing and related disciplines is not included in the present overview, neither in the individual papers I - V. This was a very deliberate choice made by the Applied Chemometrics Research Group.

In themselves papers I - V form a logical development, which will be appreciated when preceded by the present introduction to the thesis.

Concluding Remarks

Summary of the most important achievements in the thesis; (.) denotes the five papers included (I - V).

Methodological developments:

Systematics of applied MIA (I)

"*Stand alone"* MIR-implementation (II, III, IV)

Typology of MIR objectives and corresponding methods (III)

Systematics of applied MIR (III)

World's first image analysis validation facility (IV)

Applications:

Generic *remote sensing application* (Forest Montmorency) (I)

Food science and - technology examples, several types (III, IV, V)

Powder science and - technology examples (V)

Industrial application examples, several types (III, IV, V)

Collaborations:

MATFORSK, Ås

Centre de Recherce en Géomatique, Université Laval, Quebec

Chemometric Research Group, University of Umeå, Sweden

POSTEC (Powder Science and Technology), Tel-Tek, Porsgrunn

IDE-CON, Porsgrunn

Future Work

This thesis has exclusively been oriented towards developing *prototype* software and *ditto* key exemplifications and applications. Therefore MATLAB and LabVIEW ("G") was chosen for the essential development work, as argued in paper II. The "stand alone" version of MIA/MIR runs as an independent sw system however, although further development of a C++ version may be the target for future commercialisation efforts.

The MIR validation facility today only includes cross-validation (paper IV). It would be more satisfactory also to have had the time needed for including a complementary *test set validation* feature. This will have to wait for a possible future C++ version.

I am happy with the degree to which it has been possible to address basic food science and - technology problems in this thesis, in relation to the collaborative context of the Ph.D. stipend involved. There is however a host of potential broader technological application areas ahead - some of which it has only been possible to touch upon in papers I - V.

Much fascinating work remains for the next generation MIA/MIR researchers and - users.

References

[1] J.P. Wold. *Rapid quality assessment of meat and fish by using near-infrared spectroscopy, autofluorescence spectroscopy and image analysis.* Doctor Scientarium thesis (2000), Agricultural University of Norway, Ås. ISBN 82-575-0413-0, ISSN 0802-3220

[2] Bjørn-Helge Mevik (2000) *Statistical Methods for Handling Unwanted Variation in Production Processes, Using Raw Material Measurements*; Series in dissertations submitted to the Faculty of Mathematics and Natural Sciences, University of Oslo, No. 94. ISSN 1501-7710

[3] P. Geladi, E. Bengtson, K. Esbensen and H. Grahn. *Image analysis in chemistry I. Properties of images, greylevel operations, the multivariate image.* Trends in Analytical Chemistry, **11** (1992) 41-53.

[4] P. Geladi and H. Grahn. *Multivariate Image Analysis.* (1996). John Wiley and Sons, Chichester, UK pp. 316 ISBN 0-471-93001-6

[5] P. Geladi. *Analysis of Multi-Way (Multi-Mode) Data.* Chemometrics Intell. Lab. Syst. **7** (1989). 11-30

[6] P. Geladi, S. Wold and K. Esbensen. *Image analysis and Chemical Information in Images.* Analytica Chemica Acta, **191** (1986) 473-480g.

[7] P. Geladi, H. Isakson, L. Lindqvist, S. Wold and K. Esbensen. *Principal Component Analysis of Multivariate Images.* Chemometrics Intell. Lab. Syst. **5** (1989) p. 209-220

[8] P. Geladi, H. Grahn, K. Esbensen and E. Bengtsson. *Image Analysis in chemistry II. Multivariate image analysis.* Trends in Analytical Chemistry, **11** (1992) 121-130.

[9] K. Esbensen and P. Geladi. *Strategy of Multivariate Image Analysis (MIA).* Chemometrics Intell. Lab. Syst. **7** (1989) 67-86.

[10] Hotelling H. *Analysis of a complex of statistical variables into principal components.* J. Educ. Psycol., **24** (1933) 417-441, 498-520.

[11] K. Esbensen, G. Edwards and N.R. Eldridge. *Multivariate Image Analysis on Forestry Applications Involving High Resolution Airborne Imagery.* Proc. of The 8th Scandinavian Conference on Image Analysis-SCIA '93, Tromsø, May 25-28 1993. pp. 953-963

[12] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* (1993) Addison-Wesley Publishing Company, Inc. Reading, USA. ISBN 0-201-60078-1 pp. 716.

[13] P. Geladi and K. Esbensen. *Regression on multivariate images: Principal Component Regression for modelling, prediction and visual diagnostic tools.* J. of Chemometrics **5** (1991) 97-111.

[14] K. Esbensen, P. Geladi and Hans Grahn. *Strategies for Multivariate Image Regression.* Chemometrics Intell. Lab. Syst. **14** (1992) 357-374.

[15] S. Wold, H. Martens and H. Wold. *The multivariate Calibration problem in chemistry solved by the PLS method*. (1983) Proc. Conf. Matrix pencils, (A. Ruhe, B. Kågstrøm, eds), March 1982. *Lecture Notes in Mathematics,* Springer Verlag, Heidelberg, 286-293

[16] H. Martens and S.Å. Jensen. *Partial Least Squares regression: A new two-stage NIR calibration method.* (1983) Proc. 7th World Cereal and Bread Congress. Prague June 1982. (Holas and Kratochvil, eds.) Elsevier Publ., Amsterdam, 607-647.

[17] H. Martens and T. Næs. *Multivariate Calibration.* (1989). John Wiley & Sons Ltd. Chichester, UK. pp. 419, ISBN 0 471 90979 3

[18] K. Esbensen et.al. *Multivariate Data Analysis –in practice, 4th edition.* (2000) CAMO ASA, Oslo, Norway, p. 498, ISBN 82-993330-2-4.

[19] F. Lindgren, P. Geladi and S. Wold. *The Kernel Algorithm for PLS.* J. of Chemometrics, **7** (1993) 45-59.

[20] A. Höskuldson. *Prediction Methods in Science and Technology, vol. 1. Basic Theory*. Thor Publishing, DK, p. 405, ISBN 87-985941-0-9

[21] T. Yamazaki and D. Gingras. *Image classification Using Spectral and Spatial Information Based on MRF Models*. IEEE transactions on image processing **4** (1995) 1333-1339

[22] J. R. Carr. *Spectral and Textural Classification of Single and Multiple Band Digital Images*. Computers & Geosciences. **22** (1996), 849-865.

[23] N. Lamei, K.D. Hutchison, M.M. Crawford and N. Khazenie. *Cloud-Type discrimination via multispectral texture analysis.* Optical engineering **33** (1994): 1303-1313.

[24] R. Andrle. The Angle Measure Technique: A New Method for Characterizing the Complexity of Geomorphic Lines. Mathematical Geology, **26** (1994), 83-97.

[25] K. Esbensen, K. Kvaal and K.H. Hjelmen, *The AMT Approach in Chemometrics-First Forays.* J of Chemometrics, **10** (1996), 569-590.

[26] J. Huang and K. Esbensen. *Applications of AMT (Angle Measure Technique) in Image Analysis Part I: A new Methodology for in-situ Powder Characterization.* Chemometrics Intell. Lab. Syst. In press (2000).

27 Lindgren, F. 1994: Third Generation PLS. PhD thesis, Umeå University. ISBN 91-7174-911-X