# DEVELOPMENT OF VISION-BASED INFERENTIAL SENSORS FOR PROCESS MONITORING AND CONTROL

By

HONGLU YU, B. ENG., M. ENG.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

# DEVELOPMENT OF VISION-BASED INFERENTIAL SENSORS FOR PROCESS MONITORING AND CONTROL

Doctor of Philosophy (2003)
(Chemical Engineering)

McMaster University
Hamilton, Ontario

TITLE: Development of Vision-based Inferential Sensors for Process Monitoring and Control

AUTHOR: Honglu Yu, B. Eng. (Tsinghua University), M. Eng. (Tsinghua University)

SUPERVISOR: Professor John F. MacGregor

NUMBER OF PAGES: xv, 145

# Abstract

This thesis develops inferential sensors for on-line process monitoring and control based on multivariate image analysis. Several methodologies based on multivariate statistical methods, such as Principal Component Analysis (PCA) and Partial Least Squares (PLS), are developed to efficiently extract information in real-time from time varying images and to predict process and product properties. The inferential sensors developed based on these methodologies are shown to be sufficient for on-line monitoring and feedback control as illustrated through two industrial applications: snack food processes and flame monitoring. These methods can be easily extended and used for a wide variety of other on-line monitoring and control problems.

In the snack food applications features are extracted from RGB color images and used in predicting the average coating concentration on the product and the coating coverage distribution over the product pieces. Data collected using both on-line and off-line imaging from several different snack food product lines are used to develop and evaluate the approaches. Results of successful monitoring and control from the on-line applications to the industrial processes are shown. Several robustness issues such as detection of model inadequacy and on-line correction are also discussed. Preliminary results on the prediction of organileptic properties (taste and texture of snack foods) are shown as well.

In the flame application, an on-line digital imaging system is developed for monitoring a turbulent nonpremixed flame in an industrial boiler. By using PCA score plots stable information can be obtained for highly fluctuating flame images. A feature extraction approach is proposed to extract the information from the flame color images. The information extracted from the images is then used to successfully predict the

performance of the boiler system, such as the energy content of the fuel, and the concentration of $NO_x$ and $SO_2$ emissions in the off-gas using PLS regression. Results show that flame images contain a large amount of information that is useful for monitoring the performance of the boiler system. The approach is very general and can be applied to a wide range of combustion processes.

A general framework for building vision-based inferential sensors for monitoring and control of process and product properties using multivariate image analysis is presented. The applications to the snack food processes and the flame monitoring system are shown to fit into this general framework.

# Acknowledgements

I would like to thank my supervisor, Professor J. F. MacGregor, for his confidence, encouragement and support throughout my thesis work. My supervisory committee, including Professors T. E. Marlin and J. S. Chang, provided many useful suggestions which have enhanced the quality of this work.

Thanks to NSERC and the Department of Chemical Engineering for the financial support. Thanks to Fritolay, Inc. and Dupont Canada for providing data and cooperation.

For my colleagues and friends in the Penthouse: Manish, Seongkyu, Yongsong, Yaqiu, Weitao, Kevin, Niki, San, Koji, Tina, Fouad, Ben, Hatopan, Jun, Pierre, Mark-John, David, Haslinda, Mohammed, Shahab, Taufiq, Sal and Adam, thank you all for the friendship and help discussion. Thanks especially to Kevin for many suggestions to improve the thesis.

I want to especially thank Jesus, my best friend, for the unconditional support throughout my thesis.

And finally, I would like to thank my grandparents and my parents for the endless love that they have given me.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Strong competition in the marketplace is forcing industries to place much attention on improving product quality and reducing costs. Adequate control systems and efficient monitoring schemes are the key to meet such requirements. One of the most important elements in successful monitoring and control systems is fast, reliable and inexpensive on-line sensors. However, not all the important variables can be measured on-line. In this situation, a strategy often used is to estimate the unknown variable from other easily measured variables through a model, known as inferential control. In this thesis, the development of vision-based inferential sensors is studied. These kinds of sensors can greatly improve the quality control performance in at least two areas: i) industries that produce solid products such as food, polymers, pulp and paper, where product properties can only be measured periodically in a laboratory; and ii) processes whose performance is related to their visual appearance, such as flame in a combustion process and flotation froth in a mineral process.

The petrochemical industry made rapid advances in multivariable model predictive control largely because they had the availability and abundance of inexpensive and informative sensors such as thermocouples, pressure transducers, flow meters, pH and ion-specific meters and gas chromatographs. This is a direct result of the fact that the major streams in petrochemical processes consist of well mixed gases and liquids which made the use of such sensors very easy. On the other hand, the solids processing industry has had much less success at implementing advanced control precisely because of the lack of such sensors. In industries that produce solid products, the product properties are

generally measured periodically by manually collecting samples and then analyzing them in the laboratory. The analysis procedure may require that the samples be destroyed and the procedures are time consuming and manpower intensive. Therefore, it is usually impossible to obtain on-line quality measurements and even the simplest automatic control algorithm can not be implemented. In this situation, an inferential sensor based on a vision system is very attractive. By taking images of the products, the samples remain untouched and the sampling rate can be very high. The product properties can then be predicted in real time from image data.

Another area of large potential for the use of image analysis is where performance is related to process appearance. Two examples are floatation froth in mineral process and flame in combustion process. In many combustion processes, cameras have been installed for years. However, these cameras are only used to show live pictures of the flame so that operators can check if the flame is burning in some acceptable manner. Because the flame appearance is highly related to the performance of the combustion process, if a model can be developed to relate the flame images to the process properties, a better understanding of the process can be obtained and such a model can be further used for on-line monitoring and control.

With fast development of technology, the prices of cameras and computers have become cheaper while their performance and ability has improved. To set up a fairly good vision system now has similar costs to installing a thermocouple in a process. Moreover, people have accumulated over 30 years experience in building machine vision systems, so that setting up a good vision system is not a difficult task. Therefore, the main challenge is to develop the software and methodologies capable of building models that relate the image data to the process/product properties.

The scope of this thesis is to develop practical methodologies for building inferential sensors for on-line process monitoring and control based on multivariate image analysis.

Many applications of image analysis and machine vision have been used for industrial production, such as character recognition, food sorting, fault detection. However, their key objectives focus mainly in developing vision systems to replace human eyes. This is different from the objective of this thesis. The problem that is solved here is to extract information from images, which may not be easily seen with human eye, and use the information to *quantitatively* predict some process and/or product properties. It is a task that can not be performed by human vision.

The images considered in this thesis are multivariate images (e.g. a color image, which is a multivariate image with 3 variables), because compared to grayscale images they can provide much more information and can be easily obtained with today's imaging technology.

In this thesis, a two step procedure is adopted to build models: i) extracting features from the image data; and ii) regressing the features against the property data. The first step converts the image data to a feature vector related to the property data and is the key for building a successful model. The second step is straightforward and any regression method in data analysis could be used. Multivariate statistical methods, such as Principal Component Analysis (PCA) and Partial Least Squares (PLS), are used in both steps in this study because these methods have the ability to handle the high correlation existing in both multivariate image data and feature variables and they are easy to compute. A brief overview of PCA and PLS can be found in Appendix A.

Two industrial applications are studied in this thesis corresponding to the two areas mentioned before. The first application is to predict coating concentration and coating spatial distribution for snack food processes. In the second application, flame images inside an industrial boiler are analyzed and used to predict several process properties, such as the concentration of $NO_x$ and $SO_2$ emissions in the off-gas.

The organization of the thesis is as follow. Chapter 2 gives a literature review on multivariate image analysis. In this review, we categorize the problems in multivariate

image analysis that have appeared in different areas, including remote sensing, Magnetic Resonance Imaging (MRI) and multispectral microscopy imaging, in the past 30 years. Among the numerous multivariate image analysis approaches, one method using PCA and interactive masking techniques in the reduced PCA score space is highlighted. The concepts behind this approach are the key for feature extraction used in this thesis.

Chapter 3, Chapter 4 and Chapter 5 present the research on industrial snack food applications. In Chapter 3, six feature extraction methods, background/foreground separation methods and two coating concentration distribution estimation methods are developed. Chapter 4 presents some of the on-line results on two different types of snack food. Model predictions are compared with measurements from the laboratory. Process monitoring and fault detection as well as closed-loop feedback control results are also shown. In Chapter 5, some complementary studies are shown: the first part is on model robustness, including on-line model assessment and model correction while in the second part, a preliminary study on prediction of organileptic data (the properties related taste and texture of snack foods) is presented.

In Chapter 6, a feature extraction method is proposed to obtain information from highly fluctuating flame images. The feature variables extracted are further used to predict process properties using PLS models. The results of predicting heat of combustion of the liquid fuel (coming from waste streams of other processes) and the concentration of $NO_x$ and $SO_2$ emissions in the off-gas are presented in Chapter 7.

Chapter 8 summarizes the thesis and presents a general framework for prediction of process/product properties using multivariate image analysis.

# Chapter 2

# Review of Multivariate Image Analysis

The purpose of this chapter is to give a brief literature review of multivariate image analysis and machine vision, as well as to introduce different types of problems that can be addressed using multivariate image analysis. An overview of a multivariate image analysis approach, which is the basis of the thesis, is also provided.

## 2.1 Image Analysis and Machine Vision

After almost half a century of development, computers can now be programmed to manipulate images in a variety of ways. An image can be *processed* to produce modified images; it can be *analyzed* to generate various types of descriptive information about it; and conversely, descriptive information can be used to *synthesize* images (this is a major area of computer graphics). The conceptual relationships between image processing, image analysis and image synthesis are summarized in Figure 2.1 based on Rosenfeld [2001].

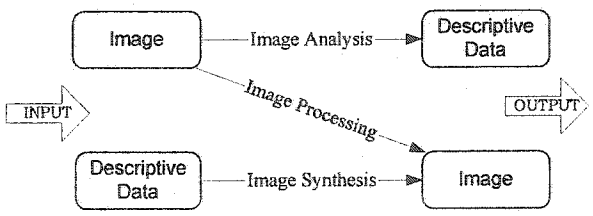



**Figure 2.1 Relationship between image processing, image analysis and image synthesis**

Image synthesis is beyond the scope of the thesis and will not be discussed here. Image processing and image analysis are closely related. Many techniques are used in both areas, such as Fourier transforms, segmentation and filtering, making it difficult to give a sharp division. Generally, image processing focuses on image enhancement and restoration (quality improvement by histogram equalization, sharpening, noise reduction, etc.); image compression, with application to image communication (television, internet) and reconstruction of 3-D images from 2-dimensional projections (e.g. stereo camera). Image processing is often used as a preprocessing step in image analysis.

In image analysis, the outputs are usually numerical descriptions rather than an image. The operations are used to extract information from an image by quantifying its elements through various mathematical descriptors of shape, size, color, texture, etc. These techniques are largely used in machine vision applications that require quantitative measurements.

There are many different types of machine vision applications, including character recognition, human motion analysis, automatic guidance, inspection of food production, object sorting, face recognition and defect detection. Even though the images that need to be analyzed in these widely varied application areas are very different, the types of analysis to be performed on these images have many things in common. Image analysis almost always involves a few basic processes (see Figure 2.2): distinguishing certain parts of the image (segmentation); measuring properties of these parts, or relations among the parts (feature extraction); and using the values of these properties or relations to classify the images or part of the images or to predict some information related to the images (classification or regression) [Rosenfeld, 2001].

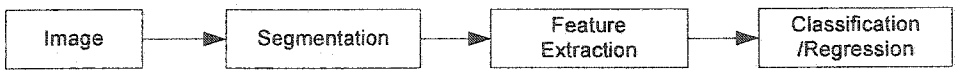



**Figure 2.2 Basic processes in image analysis**

The research work in this thesis focuses on using multivariate image analysis for process monitoring and control, however, as can be seen later, the problem solving concept still follows these basic process steps.

## 2.2 Multivariate Images

A multivariate image is a set of congruent images. The definition for congruence in imaging is given by Geladi and Grahn [1996]: two or more images are congruent if they can be stacked so that for each pixel in one image there is a corresponding pixel in the other image(s) that can be referred to the same position in the object or scene depicted.

Figure 2.3a shows a 4×512×512 satellite image of Mobile, Alabama [Geladi and Grahn, 1996] (the 4 bands are: green-yellow, orange-red, deep red and near-infrared). From the data storage point of view, a multivariate image is a three-way matrix (Figure 2.3b). Two of the ways in this three-way matrix are the geometrical image coordinates which describe the image scene plane and usually treated as a pair. The third way is the 'variable' way. Hence one could also view a multivariate image as a two-way array of pixel intensity vectors (Figure 2.3c). Multivariate image data often contain highly correlated information among the variables (see Table 2.1) because the individual images within one multivariate image are congruent.

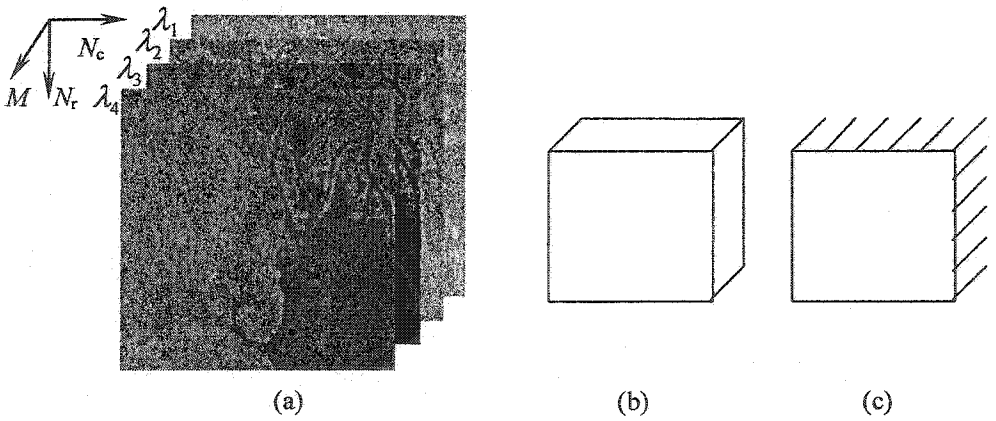



**Figure 2.3 (a) A 4×512×512 satellite image (b) three-way matrix (c) two-way matrix of pixel intensity vectors**

**Table 2.1 Correlation coefficients between variables (satellite example)**

| | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |
|---|---|---|---|---|
| $\lambda_1$ | 1 | 0.927 | 0.411 | 0.191 |
| $\lambda_2$ | | 1 | 0.554 | 0.340 |
| $\lambda_3$ | | | 1 | 0.957 |
| $\lambda_4$ | | | | 1 |

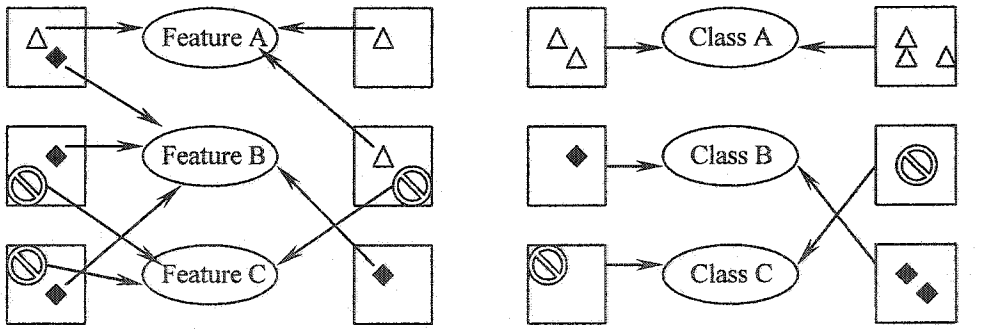
A multivariate image can be obtained from different sources. A common way is to stack the images of an object at different radiation energies or wavelengths. This is a multispectral image. Increasing amount of multispectral data have been obtained by the remote sensing satellites. Magnetic Resonance Imaging (MRI) is another imaging technique to obtain multivariate images. By varying the parameters (most common parameters are the spin-echo time and the relaxation delay time), a set of images can be obtained. This imaging technique can study biological processes in cells, isolated organs and living systems in a non-invasive way. A multivariate image can also be constructed by combining images obtained from different instruments. For example, in satellite imaging, images from different satellites over the same region can be stacked together. However, some warping methods may be necessary to make the images congruent. A multitemporal image is a type of multivariate image as well, which consists of the images of the same scene but taken at different times. A color image can be also considered as a 3-variable multivariate image because a color image consists of 3 color channels: red, green and blue. Another technique of creating multivariate images is to combine a univariate (grayscale) image with the copies of it that are derived from different spatial filtering operations. Such multivariate image can be used to study texture under the assumption that the filtered versions add spatial information on the original unfiltered image [Bharati and MacGregor, 2000].

## 2.3 Typical Problems in Multivariate Image Analysis (MIA)

Most of the image processing and analysis techniques discussed in the standard textbooks [Pratt, 1978; Rosenfeld and Kak, 1976; Gonzalez and Woods, 1992] are based on the study of grayscale images (univariate images) and do not take into account the spectral ('variable') information. However, with the fast development of imaging systems, more and more multivariate images can be obtained easily in different scientific research areas. A large body of literature exists on multivariate image processing and analysis, especially in the area of remote sensing and magnetic resonance imaging. Most of the literature focuses on compression techniques, visualization approaches and segmentation.

Depending on the final analysis objective, we divide the problems in multivariate image analysis (MIA) into two general types: classification and regression.

In classification, depending on the objects that need to be classified, there are two kinds of problems: one is to classify the features within images (image segmentation); the other is to classify the images themselves based on features in the images (see Figure 2.4).




Type I: Classify features within images  Type II: Classify images based on features

**Figure 2.4 Two types of classification problems**

Regression is another type of problem in multivariate image analysis. Based on different objectives and different types of data, we roughly divide the regression problems into 3 categories (see Table 2.2). Type I regression builds a model relating two

congruent images. Type II regression predicts non-image data (e.g. product properties) from image features. Type III regression uses the non-image data as well as image features to predict some non-image response.

**Table 2.2 Three types of regression problems**

| | X (predictor set) | Y (response set) |
|---|---|---|
| Type I | Image | Image |
| Type II | Image | Non-image |
| Type III | Non-image + Image | Non-image |

### 2.3.1 Classification ---- Type I: Classify features within images (segmentation)

The objective of this type of classification is to segment an image into several areas so that each area represents a certain feature. A large amount of the research work in the literature has been carried out on this topic.

Preprocessing: variable dimension reduction

A significant preprocessing procedure in dealing with multivariate images (especially of high variable dimension) is variable dimension reduction. The individual bands of a multivariate image are often highly correlated, so the proper reduction of the number of variables can not only remove or reduce this variable redundancy and increase the computation efficiency, but can also often result in a better signal to noise ratio and concentrate on the useful information.

For a multivariate image with a few variables, variable reduction can be obtained by selecting a subset of the variables based on some principle. However, for a high dimension multivariate image, a more efficient way to reduce dimension is by linear projection. In this type of approach, a low dimensional set of projection directions are defined by using certain criteria. The multivariate image data is then projected to this low dimensional subspace. In many cases, the projection directions are orthogonal. Many methods have been presented to find such a set of low dimensional basis vectors, including Principal Component Analysis (PCA, also known as the Karhunen-Loeve [KL]

transform) that finds the subspace that explains the largest percentage of the variation, Discriminant Analysis (DA, including Fisher's Discriminant Analysis, Canonical Component Transform and PLS Discriminant Analysis), that finds the subspace that has the largest discriminating power [Maxwell, 1976; Lied and Esbensen, 2001], Independent Component Analysis (ICA) that finds the subspace from which the projections of the data onto each of the basis vectors are independent [Kaarna, 2000], Projection Pursuit (PP) that finds the subspace that captures the non-Gaussian distributed information [Nason, 1995; Ifarraguerri, 2000] and Decision Boundary Feature Extraction (DBFE) that finds the subspace defined by the decision boundary [Lee and Landgrebe, 1993]. Among these projection methods, PCA is the simplest one in computation and the one that has been most widely used.

Other than linear projection methods, several non-linear transformation methods, such as Self-Organizing Map (SOM, [Kohonen, 1984]) and Sammon's mapping [Sammon, 1969] have also been proposed. However, the main drawback of these nonlinear mapping methods is that they often require long computation time [Bonnet, 1997].

Recently, with the development of hyperspectral imaging techniques, it is easy to acquire a multivariate image with hundreds of bands. For this type of high dimensional image, the simple linear projection may not be appropriate, because the low ratio of the amount of training data to the number of variable dimensions leads to an unreliable estimation of projection directions [Jia and Richards, 1999; Kumar, et al., 2001]. Several approaches have been presented to overcome this problem. The basic strategy is to divide the full dimension space into several groups and merge adjacent bands. Jia and Richards [1999] proposed a technique based on segmented principal component analysis, in which PCA is performed on each group of adjacent highly correlated bands and a subset of principal components from each group is selected based on their discrimination capacity. A similar idea but based on projection pursuit is used in Jimenez and Landgrebe [1999]. Other variable reduction approaches can be found in Serpico and Bruzzone [2001] and

Kumar et al. [2001]. In addition, proper preprocessing techniques (smoothing, noise reduction, taking derivatives [Tsai and Philpot, 2002], ratioing [Maxwell, 1976], etc.) may be used for all the variable reduction approaches discussed above.

Segmentation approaches

Basically, the segmentation approaches in the literature can be divided into two main categories. In the first category, only spectral (variable) information of each pixel in the image plane is considered. In another words, pixels are treated independently of one another. Therefore, a pixel can be treated as an observation in multivariate data analysis. The classical classification methods, such as K-means [Duda and Hart, 1973; Artyushkova and Fulghum, 2002], fuzzy C-means [Boudraa et al., 2000], neural networks [Reddick et al. 1997], maximum likelihood [Liang et al. 1994] and discriminant analysis, can be directly used. Under the same category, segmentation can also be obtained by thresholding the pixel intensity values based on histogram information, which has been extensively used in the segmentation of grayscale images. When used for multivariate images, one option is to find a proper thresholding value for each individual band (variable) image and combine the results [Raya, 1990; Vittorio and Emery, 2002]. However, in this approach, the nature of the multivariate image has been ignored. Another option is to find a thresholding range in the $M$-dimensional space for an $M$-band image. However, the computation of the $M$-dimensional histogram and choosing a proper range of the thresholding area will be very difficult for the case where there are more than two dimensions ($M>2$). A compromising solution is presented by Esbensen and Geladi [1989], in which the thresholding operation is performed on the 2-dimensional subspace obtained by PCA. The thresholding area is represented by a polygon mask in this subspace. A more detailed explanation of this approach will be provided in the following text.

In the second category, spatial information is used for classification as well as spectral (variable) information. This is also known as multivariate image texture analysis. There are three general approaches to segment an image based on textural information. In

the first approach, spatial and spectral information are considered simultaneously. The approaches presented are often extended from grayscale image texture analysis. Several segmentation methods based on Markov Random Fields (MRF) models and maximizing a posteriori distribution probability (MAP), which utilize both spectral and spatial information to model the local correlation structure of an image, have been presented [Kartikeyan et al., 2002; Yamazaki and Gingras, 1999]. Kovalev[2001] proposed a texture analysis method for multivariate image that is based on extended co-occurrence matrices. Extraction and classification of homogeneous objects (ECHO) [Kettig and Landgrebe, 1976; McCauley and Engel, 1995], is another texture segmentation method which is based on the spatial and spectral homogeneity of the regions. The listed methods above are only a few examples. However, methods considering both spatial and spectral information at the same time generally involve iterative computation and/or optimization that requires intensive computation time at both the training stage and testing stage. In the second approach, one or several representative images are generated by variable reduction or some other operation (e.g. using ratios). Then the texture methods for grayscale images can be applied on them individually. For example, in Haralick and Dinstein [1975] it has been shown that spatially homogeneous image areas can be found by using thresholding and smoothing of gradient images. In the third approach, the images of one or several channels in a multivariate image are filtered or transformed. The newly filtered or transformed images are expected to contain the spatial information. A new multivariate image is constructed by stacking the new images (and the original image) together. The segmentation methods in the first catalogue can then be used for classification. Haralick et al. [1973] reported a method that computes 28 textural features for each pixel and then segments the image using both spectral and texture features. Liu and MacGregor [2002] studied both of the latter two approaches by using wavelets to extract the spatial information.

Multivariate image segmentation plays a very important role in all of multivariate image analysis. It could be directly used for drawing final conclusions, such as in the area

of pattern recognition, defect detection or object sorting. Moreover, it is often used as the preprocessing procedure in solving other types of problems.

### 2.3.2 Classification --- Type II: Classify images based on features

In some cases it is necessary to compare or classify images of different content. Most of the approaches use a set of feature variables to describe the image and then use classification methods, such as PCA, discriminant analysis, neural networks, etc., to classify the feature variables. Two examples are given as follow. Vailaya et al. [1998] presented an approach to classify color images. In their approach, feature variables, such as a color histogram, color coherence vectors, moments of the DCT (Discrete Cosine Transform) coefficient, edge direction histogram, and edge direction coherence vector, are extracted from images. A weighted k-NN classifier is used for the classification. Bharati [2002b] studied the classification of paper based on multispectral Near Inferred Imaging technique. In that research PCA was performed on each NIR image. The loading vector of the first principal component was used as a feature vector. A PCA was then carried out on the feature matrix and 9 types of paper were shown to be clearly clustered in the score plot.

### 2.3.3 Regression --- Type I: Regression between two congruent images

In this situation, the predictor and response data are congruent multivariate images. To perform regression, we only need to unfold both image data as shown in Figure 2.5 and then perform the normal regression between the two matrices.

This type of problem is the traditional problem referred to as the Multivariate Image Regression (MIR) problem in Geladi and Grahn [1996]. The regression methods often used are PLS and PCR (Principal Component Regression). Lied [2000] shows that the Y response image could have 3 modes: $\mathbf{Y}_{discrim}$, $\mathbf{Y}_{grid}$ and $\mathbf{Y}_{total}$.

$\mathbf{Y}_{discrim}$ is the yes/no discriminator. In every location in the Y image, a pixel is either 1 if it belongs to a current class, otherwise 0. When the regression method is PLS,

this is actually discriminant PLS and can be used for image segmentation (problem in section 2.3.1). Though many good results obtained by this approach have been reported by several researchers [Lied, 2000; van den Broek et al. 1996], Bharati [2002b] discusses that the success of this approach requires meeting certain conditions. The classes must be "tight" and occupy a small and separate volume in the space defined by the variables of **X**. From the research results presented in the literature, there appears to be no too obvious advantage in using this discriminant PLS analysis over ordinary PCA together with a masking technique (this method will be explained in section 2.5) for image segmentation.

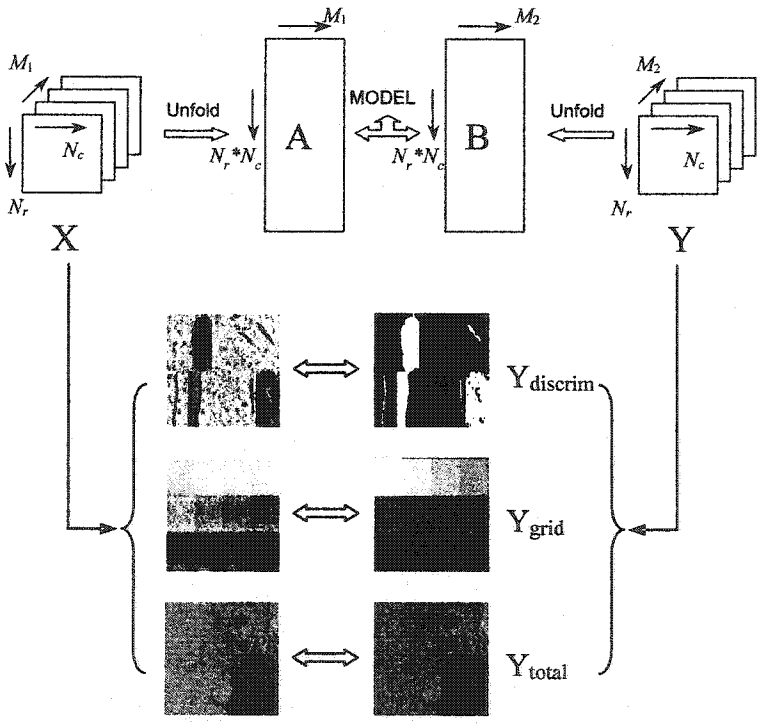



**Figure 2.5 Scheme of regression between two congruent images (some images shown are from [Lied, 2000])**

In the $Y_{grid}$ mode, the X image contains several sub-images and corresponding Y image will have a constant value for each sub-image. In Lied and Esbensen (2001), an application is studied to monitor the deteriorating process of a banana. Sixteen 200×200 images were used as sub-images to form an 800×800 X image. The Y image consisted of

16 blocks with each block having a certain gray level corresponding to the storage time. PLS was then performed between the **X** and **Y** images. The progress of the aging banana could be observed in a score plot and the prediction showed promising results. A similar study was reported in Wold and Kvaal [2000], where multispectral imaging of autofluorescence was used to map lipid oxidation in chicken meat. However, regression directly between the **X** image and the **Y** grid image tries to map the spectral information of each pixel with a certain **Y** value. For most of the pixels belonging to the same sub-image, the spectral responses are different yet are forced to correspond to the same **Y** values. Therefore, the PLS model must deal with large variations in **X** that are not related to **Y**. Similar comments can be found in [Wold and Kvaal, 2000]. Another disadvantage of this approach is that the output is an image and its use for inspection is not clear. The mean value of the predicted image is used in [Wold and Kvaal, 2000], but there is a large prediction error. To solve this type of prediction problem, a more efficient and accurate way will be to use Type II regression approaches addressed in the next section.

The $\mathbf{Y}_{total}$ mode is the situation where each pixel in the **X** image has a corresponding value in the **Y** image. An example of this regression problem is given in Shibayama and Akiyama [1991], who try to relate rice yield to multispectral images. Another example is in Kruse [1989] where an attempt is made to relate spectra of satellite images and mineral composition on the ground.

### 2.3.4 Regression --- Type II: Regression between images and non-image response

Compared to the extensive literature on the segmentation and classification problem, much less has been published on regression between multivariate images and non-image responses. Generally, to solve the type II regression problem, a feature vector is obtained from each image (see section 2.3.2), and then regression is performed between the feature data and the response data. Different feature extraction and regression methods can be used. An example is using image information to predict mineral concentrations in flotation froth [Hätönen, 1999]. 14 feature parameters were

extracted from each RGB color image of froth, including the mean, standard deviation, skewness, and kurtosis of each RGB channel and two parameters related to bubble collapse rate and spatial speed variance. Then a PLS model was built between the feature matrix and the mineral concentration. Lied [2001] presents several feature extraction approaches to predict the fraction of the mixed materials. One feature extraction method was to compute the histogram of the predicted **Y** image from MIR (see section 2.3.3). Another feature extraction method based on the Angle Measure Technique (AMT) was also presented to obtain textural information.

In this thesis, six models based on six feature extraction methods are proposed to predict the coating concentration on snack foods and will be discussed in Chapter 3. A model for prediction of organileptic properties of snack foods based on texture information will be presented Chapter 5. In Chapter 6 and Chapter 7, methods for predicting boiler behavior/response from flame images are presented.

### 2.3.5 Regression --- Type III: Regression between mixed data and non-image response

In some situations, other available process data can be combined with image data to build a model. There appears to be little literature that addresses this type of problem. However, by converting the image data into feature variables, the image data and process data can be combined and the regression model can be easily built. In Wang et al. [2002], several features extracted from a color image are combined with some data from a boiler process such as the excessive air coefficient and used as the input to a neural network to predict the $NO_x$ emissive concentration of a boiler. In the flame application in this thesis (Chapter 7), a PLS model is built to predict the heat of combustion of the liquid fuel (coming from the waste streams of other processes) by combining the information from the image and the flow rates of the fuels. The result shows that the model obtained can achieve better prediction than using only image data.

### 2.3.6 Summary of the problems in MIA

Table 2.3 gives a summary of the different types of the problems in MIA.

**Table 2.3 Typical problems in multivariate image analysis**

| Typical Problems | | Description | General Solving Scheme |
|---|---|---|---|
| Classification | Type I | Classify within image (segmentation) | Preprocessing: variable reduction<br>• variable selection<br>• projection methods<br>• nonlinear transformation<br>• merge adjacent bands ('variable')<br>Segment based on only spectral information<br>• Data analysis type of methods<br>• Thresholding<br>Segment based on spectral and spatial information<br>• Consider spectral and spatial simultaneously<br>• Use grayscale texture method on the representative images<br>• Form a new multivariate image by stacking the filtered image and then consider only band ('variable') information. |
| | Type II | Classify images | Use a set of feature variables to describe each image and then classify the images based on features |
| Regression | Type I | Regression between two congruent images | Directly unfold **X** and **Y** images and perform data regression |
| | Type II | Regression between image and non-image response | Use a set of feature variables to describe image and regression the features against response data |
| | Type III | Regression between mixed data and non-image response | Use a set of feature variables to describe image and regress the features and the process data against the response data |

## 2.4 Multivariate Image Analysis for Process Monitoring and Control

Many machine vision systems have been developed and successfully used in different kinds of industries. However the purpose of most of them is concentrated on replacing the eyes of the human operator for tasks such as character recognition, defect detection, object sorting. Most of the applications involve the problems of image segmentation and classification. Little attention has been paid to the prediction of non-

easily measured process and product quality properties using imaging techniques (Type II and Type III regression problems).

Hätönen [1999] used color image information to predict mineral concentrations in flotation froth. Bharati et al. [2002a, 2002b] developed a PLS model to predict four properties of pulp using a NIR imaging spectroscope and showed good results in at-line studies. Wang et al. [2002] reported a model obtained by a neural network to predict $NO_x$ emissive concentration for a boiler using color flame images. However, most of the studies reported in the literature are limited to off-line and experimental stages.

The focus of this thesis is on the regression problems (Type II and Type III). The key question is how to extract useful information from images. Figure 2.6 gives a summary of the hierarchy of problems in multivariate image analysis.

## 2.5 Basic Problem Solving Strategies used in the Thesis

As mentioned in Chapter 1, two industrial applications are studied in this thesis. Both applications involve prediction using color images. To solve the problems, features are first extracted from images and then regressed with the properties that need to be predicted, as described in section 2.3.3 and 2.3.4. In this thesis, we consider mainly two types of the feature extraction methods. In the first type of feature extraction method, the overall characteristic for each channel ('variable') is used as a feature variable. In the second type of the feature extraction method, the image is first segmented based on some principle and features are then computed from the segmented parts of the image.

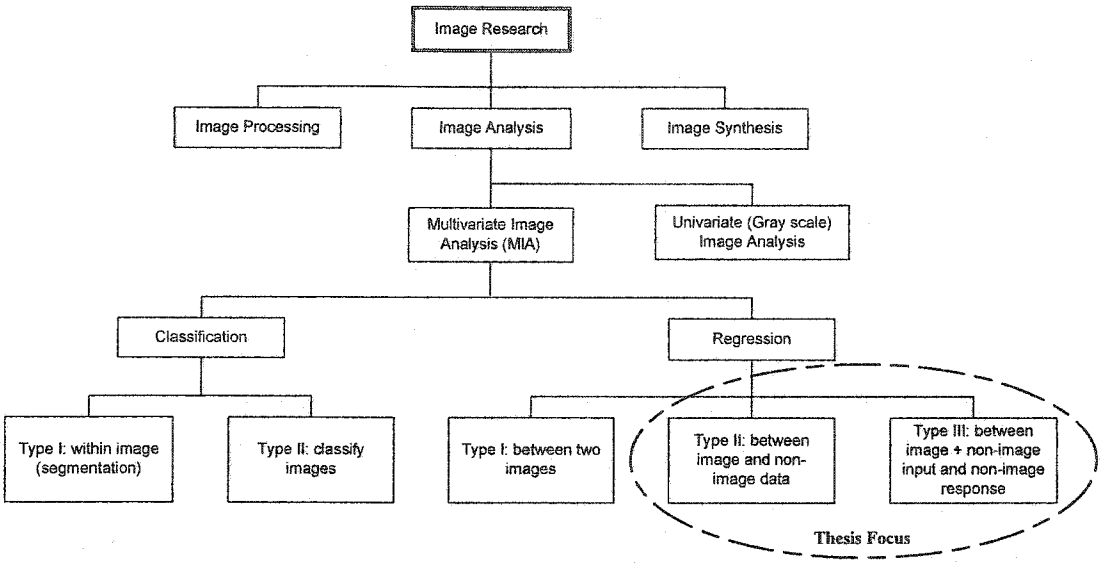



**Figure 2.6 Summary of the problems in image analysis**

The segmentation method used in this thesis is an approach based on PCA introduced by Esbensen and Geladi (1989). In this approach, PCA is performed as a preprocessing step to reduce the dimension of the data. Segmentation models, so-called masks in the score plots (can also be seen as the thresholding area), are manually determined by mapping between the score plot and the original image space. Only spectral information is used in this approach. However, as mentioned in section 2.3.1, spatial information can be incorporated by stacking filtered or transformed images.

When we perform PCA on an $M$-band multivariate image (often referred to as multi-way PCA), we unfold the image and consider it as a set of vectors in an $M$-dimensional space:

$$\underline{\mathbf{I}}_{N_r \times N_c \times M} \xrightarrow{unfold} \mathbf{I}_{N \times M} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_i \\ \vdots \\ \mathbf{c}_N \end{bmatrix}$$

where $\underline{\mathbf{I}}$ is a three-way multivariate image matrix with image size $N_r \times N_c$ and $M$ bands. $\mathbf{I}$ is the unfolded two-way image matrix. $N$ is the number of pixels in the image, $N = N_r \times N_c$. $\mathbf{c}_i$ ($i=1,\ldots,N$) is the $i$-th row vector of $\mathbf{I}$ with element length $M$, which represents the variable responses of pixel $i$. In the following text, we always use the two-way matrix $\mathbf{I}$ to represent a multivariate image.

After performing PCA, we have

$$\mathbf{I} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T + \mathbf{E}$$

where $A$ is the number of principal components, the $\mathbf{t}_a$'s are score vectors and the corresponding $\mathbf{p}_a$'s are loading vectors, $\mathbf{E}$ is the residual.

Since the row dimension of the $\mathbf{I}$ matrix is very large (equal to 262,144 for a 512×512 image space) and the column dimension is much smaller (equal to 4 for the satellite image shown in Figure 2.3a), a kernel algorithm [Geladi, et al. 1989] is used to compute the loading and score vectors. In this algorithm a kernel matrix ($\mathbf{I}^T\mathbf{I}$) is first formed (for a set of $K$ images, the kernel matrix is calculated as $\sum_{k=1}^{K} \mathbf{I}_k{}^T \mathbf{I}_k$), and then singular value decomposition (SVD) is performed on this low dimension matrix (4×4 for the satellite image) to obtain the loading vectors $\mathbf{p}_a$ ($a=1,\ldots A$). After obtaining the loading vectors, the corresponding score vectors $\mathbf{t}_a$ are then computed via $\mathbf{t}_a = \mathbf{I}\mathbf{p}_a$. $\mathbf{t}_a$ is a long vector with length $N$. After proper scaling and round off, the $\mathbf{t}_a$ vectors can be refolded into the original image size and displayed as an image.

$$s_{a,i} = \text{Round}\left( \frac{t_{a,i} - t_{a,\min}}{t_{a,\max} - t_{a,\min}} \times 255 \right), \; i=1,\ldots,N$$

$$(\mathbf{s}_a)_{N\times 1} \xrightarrow{refold} (\mathbf{T}_a)_{Nr\times Nc}$$

where $\mathbf{s}_a$ is the scaled and rounded score vector. $\mathbf{T}_a$ is the score image of component $a$. The values in $\mathbf{s}_a$ and $\mathbf{T}_a$ are integers from 0 to 255. It should be pointed out that when a set of images are studied, a common scaling range ($t_{a,\min}$ and $t_{a,\max}$) should be used for all the images.

Score images have been extensively used for visualization of remote sensing imagery, as well as multimodal medical imaging. Figure 2.7 and Figure 2.8 show the $T_1$ image and the composite image obtained by using scaled $T_1$, $T_2$ and $T_3$ images as red, green and blue channels respectively.

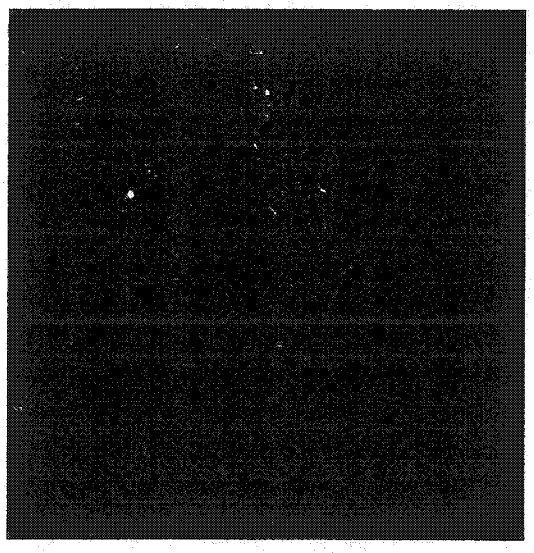


**Figure 2.7 $T_1$ score image**

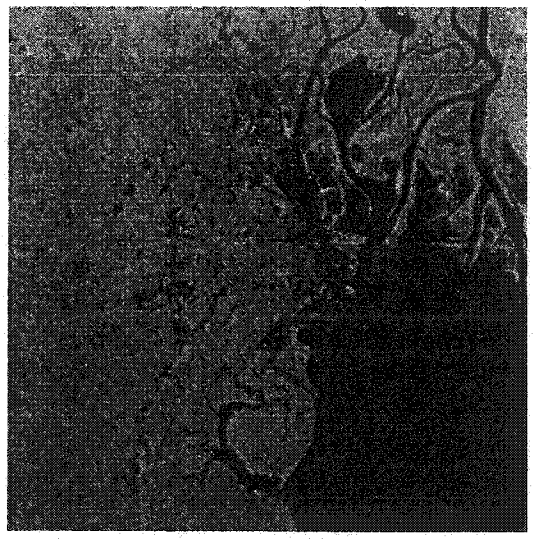


**Figure 2.8 Composite image**

**($T_1$, $T_2$ and $T_3$ score images are used as red, green and blue channel)**

The number of principal components ($A$) can be chosen by cross validation, or based on the variance percentage explained, or by visual inspection of the residual image [Geladi and Grahn, 1996]. Generally, the value of $A$ is smaller than the original variable number. In the satellite example, the first two components explained 97% of the variance. Therefore, instead of working in the original 4-dimensional space, working in the 2-dimensional orthogonal $t_1$-$t_2$ score space allows one to interpret the images much easier.

A $t_1$-$t_2$ scatter score plot (a plot of $t_1$ values vs. $t_2$ values) is a common tool in PCA to give an overview of the whole system and/or to detect clusters or outliers. However, when the studied objects are images, because of the large number of pixels, many of the pixels may have nearly identical $t_1$-$t_2$ values and plot on top of each other. Following the suggestion of Geladi et al. ([1989]), a 256×256 histogram, is used to describe the score plot space in this situation and depending upon the number of pixels in each bin a color scheme is used, ranging from dark colors (e.g. black) representing bins with a low number of pixels to light colors (e.g. white) representing bins having a high pixel density. This two-dimensional histogram, denoted as **TT**, can be obtained from scaled and rounded score vectors, $\mathbf{s}_1$ and $\mathbf{s}_2$. **TT** is a 256×256 matrix with elements computed as

$$TT(i,j) = \sum_l 1, \quad \forall l, s_{1,l} = i \,\&\, s_{2,l} = j$$

$$i, j = 0,\ldots 255$$

In the following text, the $t_1$-$t_2$ score plot is represented by the color coded 2-dimensional histogram **TT**. The $t_1$-$t_2$ score plot of the satellite image is shown in Figure 2.9.

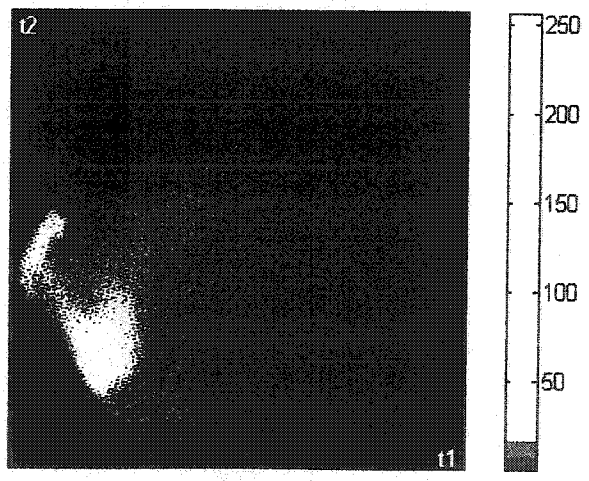



**Figure 2.9 $t_1$-$t_2$ score plot of the satellite image**

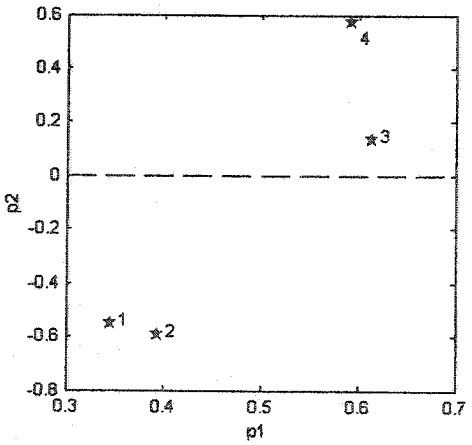



**Figure 2.10 $p_1$-$p_2$ loading plot of the satellite image**

As in normal PCA, the loadings from PCA contain information about the correlation structure of the original variables and the importance of each wavelength band (variable) in each component. Figure 2.10 shows the $p_1$-$p_2$ scatter loading plot for the satellite image. From the loading plot we can see that band 1 and 2 have high correlation and band 3 and 4 have high correlation. This conclusion is consistent with the information shown in Table 2.1. Furthermore, the loading plot reveals that the first principal component is dominated by high weights for all bands (i.e. $\mathbf{t}_1$ is some type of average spectral response among all bands), and the second principal component $\mathbf{t}_2$ is dominated by the difference between bands 1,2 and band 4.

The most important use of the score plot is to segment the image using the technique of interactive visual inspection of the score plots and the original image space. Because pixels having similar spectral features will have similar score values, these pixels will fall in a cluster in the score plot no matter where their spatial locations were in the original image space. Therefore a group of pixels can be selected in the score plot (it often can be done by drawing a polygon around the pixels on the screen) and when mapping these pixels back to the image space they represent a certain class of feature. The polygon area in the score plot is called a mask. One typical example of using this technique for image segmentation is shown in Figure 2.11. In Figure 2.11a, 5 masks are shown corresponding to 5 classes of pixels: deep water (dark blue), shallow water (light blue), road (red), golf course (yellow) and sand pits (green). Figure 2.11b shows the locations of the pixels falling in these masks in the original image space. The same colors as the masks are used to represent the features in the image space. The decision of the mask size and boundaries is made by trial-and-error selection of masks and interactively mapping the masked pixels into the original image space. When the number of principal components is higher than 2, multi-dimensional masks can be defined in different combination pairs of the score space. Bharati [2002b] studied the selection of multi-dimensional masks for an NIR lumber image (110 bands) which had 7 significant principal components.

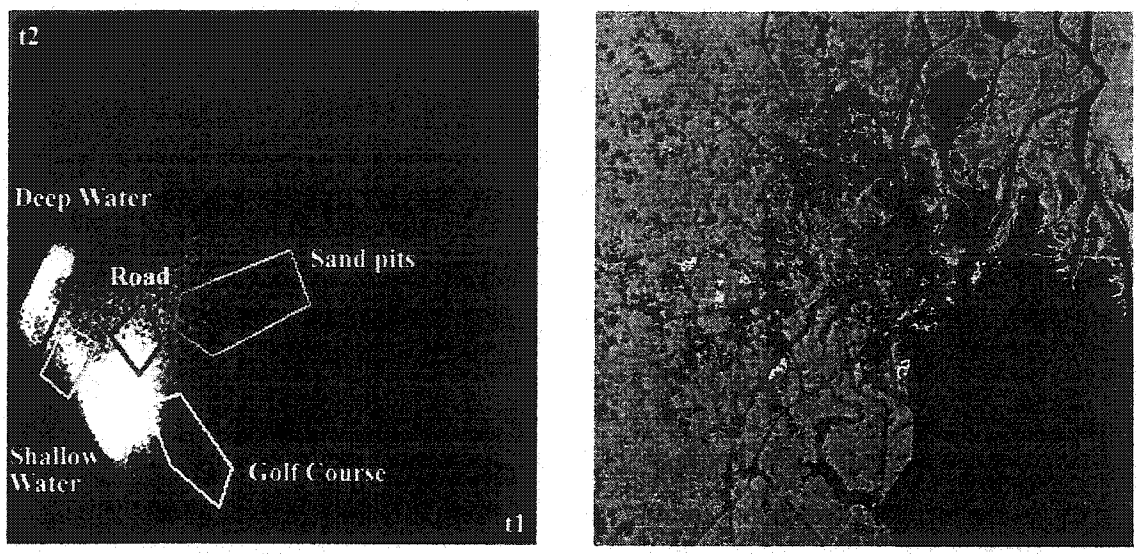



**Figure 2.11 (a) Five masks are indicated by colored polygons in the $t_1$-$t_2$ score plot for the satellite image. (b) The regions corresponding to the masks shown in (a) are indicated using the same color scheme in the original image plane.**

The use of PCA for compressing image data and for information extraction in multivariate image analysis was studied in the early 1970s in the remote sensing research area. The earliest paper available to the author is by Ready and Wintz [1973]. In this paper, PCA was performed on a 6 band airborne- and a 3 band satellite-gathered multispectral image. The score images were shown to be very useful for classification. The significant improvements in signal-to-noise ratio were also studied. In the work of Donker and Mulder [1977] the use of the false-color composites of score images and score plot classification was introduced from the study of a four-band Landsat image. In the book of Robert [1983, p136], a 2-dimensional histogram, so-called scattergram, is used to describe the bi-variable space. In the same book [Robert, 1983, p177], a classification algorithm, called *table look-up* developed by Eppler [1974], is explained. This table look-up technique is used to describe the class partitions and threshold boundaries. In two dimensions the decision boundaries are stored as four tables that describe the lower and upper decision boundaries for each class. This is very similar to

the masking technique described above. Moreover, an interactive processing method to choose the proper look-up table was also presented in the book. Though the example used was only in one dimension (limited by the hardware and software conditions at that time), the concept of interactive mapping with human judgment was similar to the masking between the score plot space and the image space. Since the 1980s research related to multivariate image analysis emerged in many areas other than remote sensing. Browning [1985] studied multispectral Auger microscope images, in which scatter plots of two individual band (variable) images was used. Kargacin and Kowalski [1986] use PCA and cross-validation to decide the number of components in a 12×64×64 ion image and a multivariate curve resolution procedure is then applied. In El Gomati et al. [1987], the bivariate scatter plot and a rectangular mask was used to separate different feature clusters, illustrated with scanning Auger microscopy. In the work presented by King et al. [1989], PCA was performed on a 20×64×64 multispectral X-ray photoelectron spectroscopy image and the scatter score plot was used for classification. In 1989, Esbensen and Geladi connected the research works from different areas and presented a general strategy for multivariate image analysis using PCA (the approach described above), illustrated by a seven-channel Landsat image with emphasis on interactive visual analysis. They also incorporated their multivariate data analysis experience from chemometrics. Their work drew renewed attention to using PCA and provided a simple but useful tool using improved computer graphical techniques. A similar interactive analysis between a 2-dimensional histogram (called concentration histogram) and the image space was presented by Bright and Newbury [1991].

By borrowing ideas from other multivariate statistical methods, additional approaches have been presented. Maxwell [1976] has shown the use of canonical transformation in multispectral images. Robert et al. [1992] also used canonical discriminant analysis for segmentation, illustrated using a multivariate near-infrared image. The idea of Multivariate Image Regression (MIR), introduced in 1991 by Geladi and Esbensen, was based mainly on Principal Components Regression (PCR) and Partial Least Squares (PLS). Many tools in normal multivariate data analysis have also been

applied in multivariate image analysis, such as cross validation [Lied, 2000], SIMCA classification [Wold, et al. [2001], etc. Classification using methods other than manually choosing masks has also been studied. Artyushkova and Fulghum [2002] presented the research on classification of the feature space using maximum likelihood, K-means clustering and neural networks.

Compared to other segmentation methods reported in the literature (see section 2.3.1), the model built by selecting masks in the score plot is not based on numerical computation, but on a manual trial-and-error procedure. The main criticism is that the resulting model is not optimal and/or unique because the trial-and-error process would lead to slightly different shapes and sizes of masks for different persons or for the same person at different time. However, a fact that exists in dealing with *real* multivariate image analysis is that the criterion for judging the results of different methods is often by visual inspection rather than some quantitative measurements. In the manual trial-and-error mask-finding process, the model is updated based on visual inspection information. Therefore, the obtained model, although it may not be optimal in some mathematical sense, can produce good segmentation results. On the other hand, models computed from more complicated methods may not lead to good visual results even though they are obtained from some kind of optimal criterion. Moreover, the optimality is only based on the training set. In the research of Artyushkova and Fulghum (2002), they showed that slight deviations from the selected training set can sometimes cause poor classification. Furthermore, since the model is dependent on the training set and/or other parameters (e.g. hidden element number in neural networks, parameters in the maximum likelihood method, etc.), if the model does not satisfy the visual inspection criteria, the readjustment of the training set or parameters to achieve a better result is not as explicit nor as simple as redrawing a mask in the score plot.

The concept behind the method described above (PCA together with mapping between the score plot space and the image space) is the basis of feature extraction methods developed in the thesis. However, we are not limited to the original approach. In

the thesis, we select masks using different methods depending on the different problems. For example, we use the traditional manual way to choose the mask for the flame luminous area (see section 6.5) in the flame application; a semi-automatic way is developed to choose the mask for separating the background and foreground image (see section 3.7) in the snack food application; an automatic way to choose masks based on covariance properties is presented in section 3.4 for prediction of coating concentration on the snack food.

# Chapter 3

# Snack Food Application (Part I) — Methodologies

Chapter 3 to Chapter 5 present a series of studies involving feature extraction and prediction on an industrial snack food production process. The main objective is to develop an inferential sensor for on-line prediction of coating concentration on the snack food and coating coverage distribution over the product pieces using an RGB color vision system. In Chapter 3, methodologies are developed and evaluated using datasets collected both off-line and on-line. In Chapter 4, some on-line results are presented, while in Chapter 5 complementary studies on on-line model assessment and correction, as well as prediction of organileptic properties using images are presented.

## 3.1 Introduction

The snack food industry is constantly striving to obtain a competitive advantage by improving their product quality. The quality of these products is mainly influenced by features such as texture, shape, color, flavor, and nutritional content. Most snack foods typically are produced by coating the "product". Coating plays an important role in both the flavor and the appearance of a snack, and greatly influences its acceptability.

Topical coatings are applied to snacks primarily by one of three methods: i) dusting or dry coating, particularly for fried snacks, ii) spraying of a flavoring blend suspended in vegetable oil onto the snack surface, and iii) oil spraying of baked products followed by dusting. There are two important factors that influence the consistency of the

coating in the finished product: the consistency of the flow of the coating to the applicator, and the degree of adhesion to the product. Adhesion depends on the temperature and the concentration of the surface oil, the coating particle size and shape, and even the condition of the frying oil.

The usual quality control procedure of periodically taking samples of the product and analyzing the sample concentration in a laboratory is often inadequate. There is a long time delay and the sample analysis procedure is slow and destructive. Moreover, coating concentration is often not obtained directly, but by measuring the salt concentration with the assumption that the ratio between the coating and salt concentration remains constant, which is not necessarily true. Under such circumstances a method based on image processing is obviously attractive. By using a camera, the surface remains untouched and the method can be used on-line for almost instantaneous monitoring and feedback control.

Traditional image analysis methods are used in a variety of applications in the food industry, such as fruit and vegetable sorting and grading, automatic portioning, inspection for foreign objects, and general package-line applications [Pla et al. 1999; Wulfsohn et al. 1993; Recce et al. 1998; Frazier et al. 2000]. However, applications on food systems containing particulate coatings are very few. A method was presented [Shan et al. 1997] to estimate surface concentration of adhered particles. However, the method is based on manually selecting a coated region and counting its area with the assumptions that the surface is flat, that the adhered particles have a distinct color and that there is no overlapping of the product, none of which are generally valid in real production.

The objective of this chapter is to develop a practical procedure to predict the coating concentration and coating coverage distribution of snack foods from the RGB images, which can be further used for feedback control and process monitoring. In this procedure, each image is represented in terms of feature variables, which are then regressed against coating concentration using Partial Least Square (PLS) method. The goal is to choose features most related to coating concentration and least influenced by

lighting variations caused by non-flat surface and overlapping among products and/or other factors. Section 3.2 gives the problem definition. The industrial data set of three types of snack food products used to illustrate and compare the performance of the methods is presented in section 3.3. Two categories of feature extraction methods are introduced in section 3.4: those based on overall color features and those based on distribution color features. Results and discussion are illustrated in section 3.5 and section 3.6, respectively. In order that the model can be industrially implemented, the problem of separating background and product pixels is addressed in section 3.7. Section 3.8 presents two new methods to effectively monitor the coating coverage. In Chapter 4, implementation issues will be addressed and on-line monitoring and feedback control results will be presented. Robustness issues for on-line long term running are illustrated in Chapter 5.

## 3.2 Problem Definition and Basic Model Building Strategy

Consider a data set of $K$ color images $\mathbf{I}_k$ and their corresponding average coating concentration $\bar{y}_k$ ($k$=1,...,$K$) (measured from lab analysis). As mentioned before (section 2.3.3), a model to predict coating concentration can be obtained by regressing the features extracted from image against the average coating concentrations (see Figure 3.1). Feature extraction converts the image into a feature vector that contains information most related to coating concentration, and is therefore a critical step to achieve good prediction. In the regression step, different regression methods may be used. Here PLS regression is employed because it can well handle the high correlation among the feature variables.

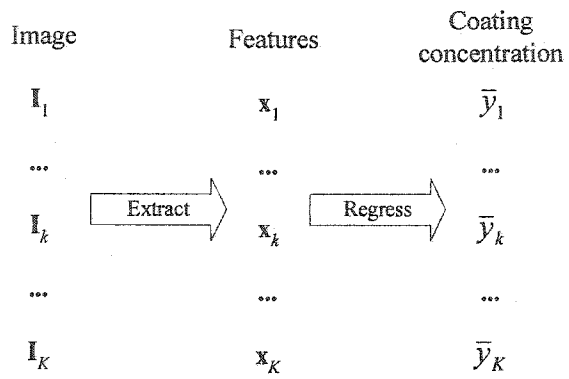


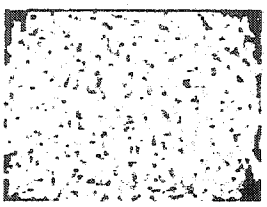
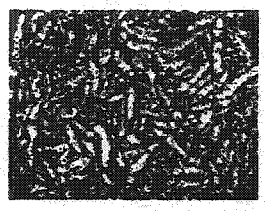
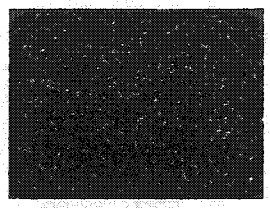
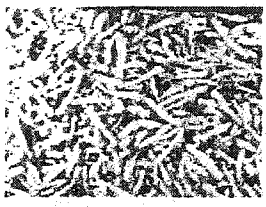
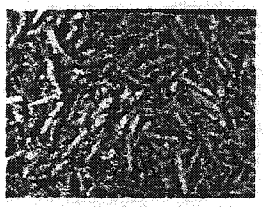
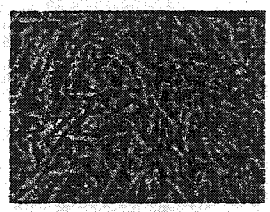
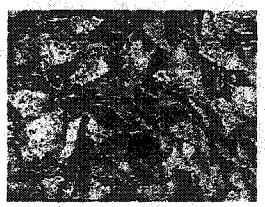

**Figure 3.1 Scheme of the model development procedure**

## 3.3 Industrial Datasets

The datasets include both uncoated and coated product samples (coating levels are varied). For each coated product image, snack products corresponding to the image are collected to measure the average coating concentration in a laboratory. A sampling error exists because this manual sample collecting procedure can not guarantee that the sample products sent to laboratory are the same as the ones shown in the image.

A number of samples for 3 types of products were gathered. Samples of product A were collected off-line in the lab, while samples of product B and product C were collected on-line. All images are 480×640 RGB color images, with 256 intensity levels in each channel, corresponding to a image scene about 25×30cm$^2$. Half of the samples are used as training sets and the other half as test sets. Some sample images and additional information can be found in Table 3.1.

**Table 3.4 Sample images for three types of snack food products**

| Product | Sample images | | | Comments |
|---|---|---|---|---|
| | NonUncoated | Low-coated | High-coated | |
| A |  |  |  | Off-line; 83 samples; 40 for training and 43 for test |
| B |  |  |  | On-line; 180 samples; 90 for training and 90 for test |
| C |  |  |  | On-line; 110 samples; 55 for training and 55 for test |

## 3.4 Feature Extraction Methods

### 3.4.1 Factors influencing the color for a pixel

From the sample images shown in Table 3.1, we notice that as the average coating concentration increases (across the rows of Table 3.1), the overall visual feeling of the colors in the images becomes redder. This human vision experience indicates that a certain relationship exists between the color and the coating concentration. However, such a relationship is not a one-to-one mapping, because the pixels containing the same coating concentration may not appear with the same colors. For example, pixels from an uncoated image (first column of Table 3.1) have different colors while their corresponding coating concentrations are all zero.

Consider a piece of pixel-size coated product. The color of this pixel would depend on the color of the uncoated part, the color of the coating particles, the coating concentration and the lighting condition of this piece of product, as shown in Figure 3.2. It is assumed that there is no variation in the global condition of the vision system because the images were collected within a short period of time. The influence of the variations in the vision system will be addressed in Chapter 5. If factors other than coating concentration are lumped together into one factor $\phi$, which will be referred to as the imaging condition, the color of one pixel $\mathbf{c}_i$ can be expressed as function of the local imaging condition $\phi_i$ and local coating concentration $y_i$.

$$\mathbf{c}_i = f(y_i, \phi_i) \tag{3-1}$$

Equation (3-1) reveals the relationship between the coating concentration and the color value. It will be used as the foundation of the feature extraction methods.

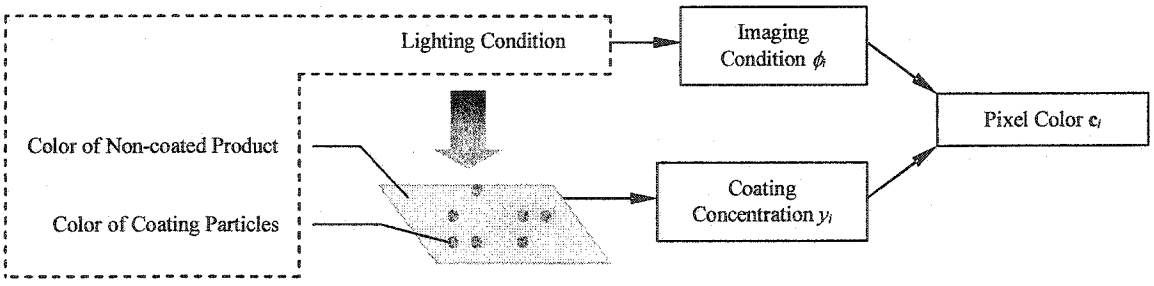



**Figure 3.2 Factors that influence the color of a piece of pixel-size coated product**

### 3.4.2 Feature extraction in RGB space

Six feature extraction methods are proposed in this chapter (see Table 3.2). These methods can be further classified into two categories: overall feature methods and distribution feature methods. In the following explanation, product C is used as the example to illustrate all six feature extraction methods.

**Table 3.2 Six feature extraction methods**

| | Method | Feature variables |
|---|---|---|
| Overall Feature Methods | 1 | Average color |
| | 2 | Loading vector of the 1st principal component |
| Distribution Feature Methods | 3 | Two-dimensional histogram in $t_1$-$t_2$ score space |
| | 4 | Histogram based on linear projection in $t_1$-$t_2$ space |
| | 5 | Cumulative histogram based on linear projection in $t_1$-$t_2$ space |
| | 6 | Cumulative histogram based on covariance property segmentation |

### 3.4.3 Overall feature methods

In these methods information from all pixels is used to obtain some overall color feature. Two methods are presented: method 1 (average color) and method 2 (loading vector of the 1st PCA principal component). The dimension of the feature vector is three for both methods, one element for each of the three color channels. Since color is a function of coating concentration (equation(3-1)), the overall color features will be correlated with the average coating concentration and models can then be built based on this correlation relationship.

Method 1: Average color

Average color is a straightforward way to extract features. In this method, the feature variables are the average values for each channel taken over the whole image.

Method 2: Loading vector of the first principal component

When performing PCA (without mean-centering) on an image, the first loading vector represents the major direction of the pixel intensities in the RGB color space. Therefore, it is also a good overall descriptor of the image color and can be used as a feature variable.

Figure 3.3 shows the relation between these two overall features (average color and first principal component loading vector) and the coating concentration. Models are

then developed based on regressing these sets of features against the average concentration data for all the training images.

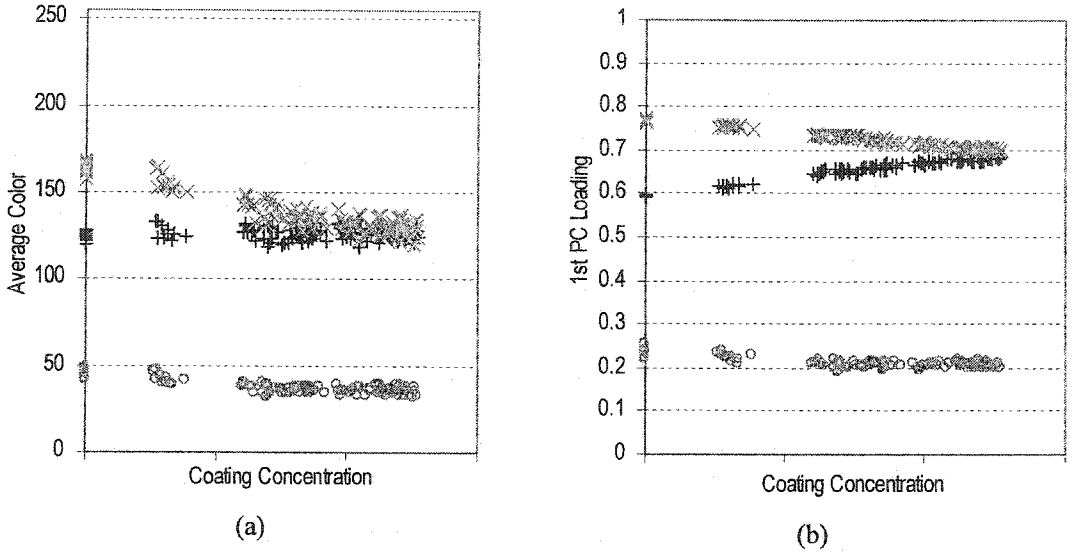



**Figure 3.3 Relation between overall feature and average coating concentration. (a) Average color, (b) Loading vector of the first principal component.**
**(+: Red Channeal,×: Green Channel;○: Blue Channel)**

### 3.4.4 Distribution feature methods

In this section, another type of methods, called distribution feature methods, is derived in order to obtain more detailed information. As shown in equation (3-1), the color of a pixel is a function of both the local coating concentration and the local imaging conditions. It is reasonable to assume that pixels having the same color contain the same amount of coating. For an image, the number of pixels, whose color is [r g b] (r, g and b are integer between 0 and 255), can be counted as $n_{[r,g,b]}$ . If the coating concentration for each of these pixels is denoted as $y_{[r,g,b]}$, then the average coating concentration for the whole image can be calculated as:

$$\bar{y} = \sum_{r}\sum_{g}\sum_{b} y_{[r,g,b]} \frac{n_{[r,g,b]}}{N} \quad r,g,b=0,1,\ldots,255 \tag{3-2}$$

Notice that equation (3-2) has a linear model structure. For each image, $n_{[r,g,b]}/N$, which is a 256×256×256 three dimensional relative histogram, can be used as feature variables. Estimation of $y_{[r,g,b]}$ can be obtained through a linear regression between the observed values of $n_{[r,g,b]}/N$ computed from the images and the average coating concentration from the lab analysis. However, this model is neither robust nor practical because to obtain such a 3-D histogram for each image is time consuming and even if such a three dimensional histogram is computed, the large number ($256^3$=16,777,216) of feature variables are extremely highly correlated and generally have a very low signal-to-noise ratio. This leads to a very ill-conditioned and poor model for predicting the average coating level.

To reduce the number of feature variables, we can divide the color space into $L$ regions, such that pixels falling in a region contain a similar amount of coating. In this form, each region becomes a new histogram bin and the average coating concentration for an image can be obtained by:

$$\bar{y} \approx \sum_{j=1}^{L} y_j \frac{n_j}{N}$$

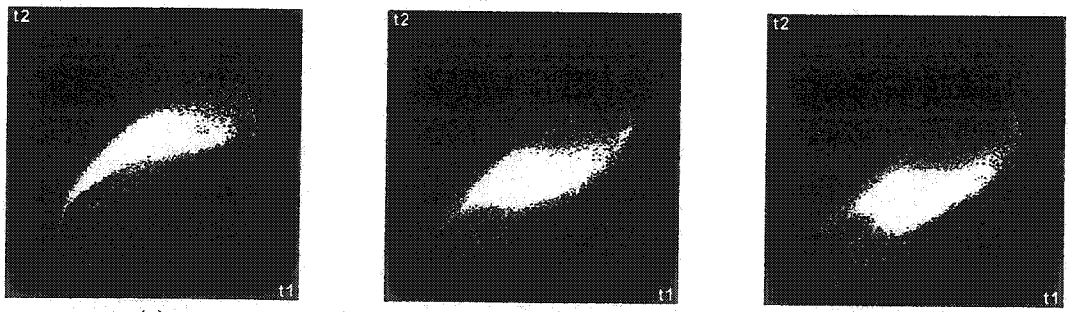
where $n_j$ and $y_j$ is the number of the pixels and the average coating concentration belonging to region $j$, respectively. The model structure remains linear as long as the assumption that pixels in the same region represent a similar amount of coating is not strongly violated.

Models based on distribution features have an advantage over the overall feature models, in that information from every region is taken into account rather than using only a global description of the image.

Three different methods are now presented to divide the color space into regions. In all the methods, PCA is first performed to reduce the dimension and all the operations are then carried out in the $t_1$-$t_2$ score space. Method 3 uses a simple 32×32 two-dimensional histogram in $t_1$-$t_2$ score space. Method 4 and 5 are based on a one-dimensional histogram and a cumulative histogram, respectively, obtained by a further linear projection in $t_1$-$t_2$ score space. Method 6 begins with a fine 256×256 two-dimensional histogram in $t_1$-$t_2$ space, then combines the elements having similar coating based on covariance properties and a new one-dimensional cumulative histogram is eventually used as feature vector.

Method 3: two-dimensional histogram in $t_1$-$t_2$ score space

One effective way to reduce the number of bins is to perform PCA on training images and find a $t_1$-$t_2$ plane that contains most of the information (see section 2.5). The images are corrected by subtracting the average color of uncoated product images in order to obtain a plane that captures most of the difference between coated and uncoated product.




(a) (b) (c)

**Figure 3.4 Score plots for three sample images of product C. (a) uncoated product image, (b) Low-coated product image, (c) High-coated product image**

Three scatter scaled $t_1$-$t_2$ score plots (see section 2.5) of three sample images of product C (Un-, low- and high- coated product as shown in Table 3.1) are illustrated in Figure 3.4. From these plots we can see that the distribution of the pixels in the score space strongly correlates with coating concentration.

In the $t_1$-$t_2$ space, if we divide each score variable into only 32 bins, the number of final bins would be reduced to $32^2$=1,024. This is illustrated in Figure 3.5. The white pixels indicate the location of pixels for all the training images. The number of bins can be further reduced by simply dividing the score variables into fewer bins and enlarging the bin size; however, at some point, the precision may be degraded. This 32×32 two-dimensional histogram formed by summing the number of pixels falling in each bin can be unfolded into a row vector and used as feature vector for each image.

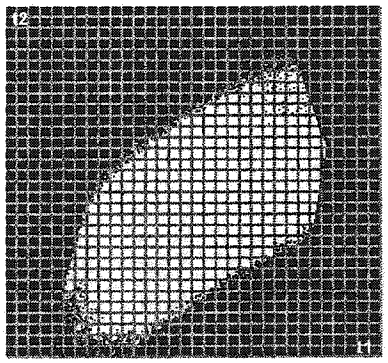


**Figure 3.5 Illustration of method 3: score plot space divided into 32×32 bins**

Method 4,5: One-dimensional histogram/cumulative histogram based on a linear projection in $t_1$-$t_2$ plane

The (32×32) two-dimensional histograms still have many bins which may lead to an ill-conditioned regression problem. Furthermore, not all variations in the histograms seen in the $t_1$-$t_2$ space will be related to spice concentration changes.

If one can find a projection direction in the $t_1$-$t_2$ score plot, along which the variation of color is unrelated to coating concentration changes, then the dimension can be reduced to one. One approach to find such a direction is to perform another PCA (mean-centered) on the $t_1$-$t_2$ score data of only the uncoated product images. Since all color variations in uncoated products are unrelated to coating concentration, the first component direction in this space ('1-1' line in Figure 3.6) will indicate the direction of the largest variance which is due mainly to changes in imaging conditions such as lighting and shading, or to changes in base product color. To use this approach, the pixels from each image are first projected onto the $t_1$-$t_2$ plane using MPCA model for the training images. Then the scores are projected along the first component direction obtained from the PCA performed on the uncoated image score plots. In this way the original three-dimensional RGB space is simplified to a one dimensional set of parallel bins in the $t_1$-$t_2$ score space (see Figure 3.7). The one dimensional histogram (method 4) and cumulative histogram (method 5) can be computed by counting the number of pixels falling into each parallel bin. These histograms are shown in Figure 3.8 where 32 bins are used. The 32 histogram and cumulative histogram points obtained for each image are then used as features to regress against the average coating content for these images.

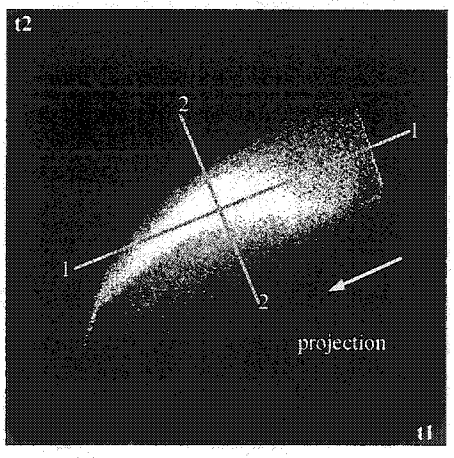



**Figure 3.6 Performing PCA on score data of uncoated images**

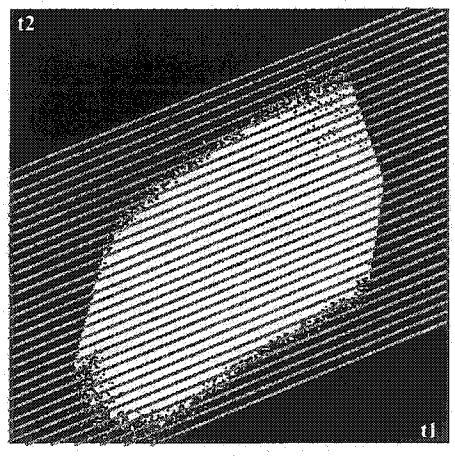



**Figure 3.7 Illustration of method 4 & 5: score plot divided into parallel histogram bins**

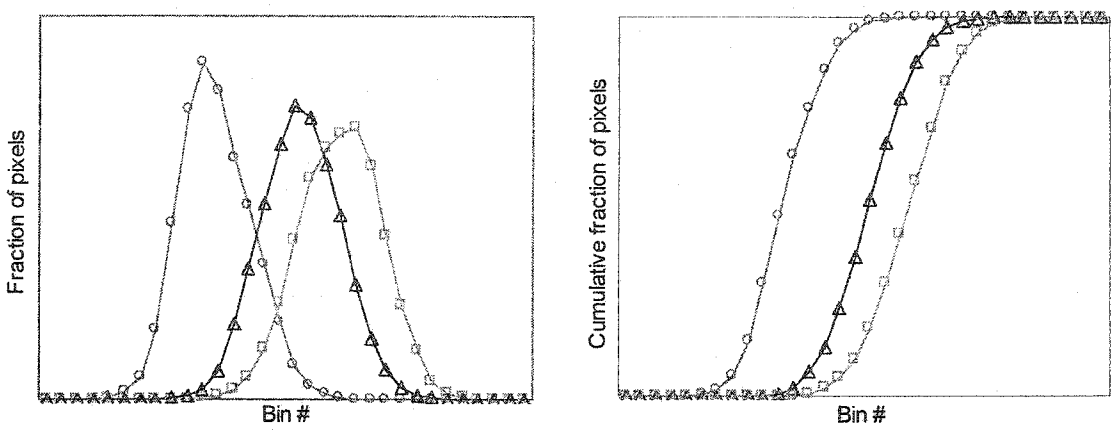



**Figure 3.8 Feature variables of method 4 & 5. (a) Histogram, (b) Cumulative histogram. (○: Uncoated product image; △: Low-coated product image; □: High-coated product image)**

Method 6: Cumulative histogram based on correlation property segmentation

In methods 4 and 5, a linear projection is used to convert the two-dimensional histogram into one-dimensional histograms. However, a linear projection in the score space may not achieve the best result because imaging conditions may exhibit a nonlinear behavior in this space.

In method 6, we start with the 256×256 histogram in $t_1$-$t_2$ space as we did in methods 4 and 5, and then lump the histogram elements that are expected to contain similar coating concentrations into bins based on the covariance properties between the histogram elements and two variables, $\mathbf{z}_1$ and $\mathbf{z}_2$, related to the average coating concentration.

For an image **I**, we can express its relative histogram as:

$$\mathbf{P}_\mathbf{I} = [P_\mathbf{I}(B_1) \quad P_\mathbf{I}(B_2) \quad \cdots \quad P_\mathbf{I}(B_M)]$$

where $M$ is the total number of histogram elements. For a 256×256 histogram in the $t_1$-$t_2$ space $M$ equals to $256^2$=65,536. $P_{\mathbf{I}}(B_i)$ is the pixel count for element $B_i$ divided by the total number of image pixels, which is an estimation of the probability of pixels falling into the $i$-th element.

A matrix $\Gamma$ can be constructed by stacking relative histograms for all the training images together:

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{P}_{\mathbf{I}_1} \\ \mathbf{P}_{\mathbf{I}_2} \\ \vdots \\ \mathbf{P}_{\mathbf{I}_K} \end{bmatrix} = \begin{bmatrix} P_{\mathbf{I}_1}(B_1) & P_{\mathbf{I}_1}(B_2) & \cdots & P_{\mathbf{I}_1}(B_M) \\ P_{\mathbf{I}_2}(B_1) & P_{\mathbf{I}_2}(B_2) & \cdots & P_{\mathbf{I}_2}(B_M) \\ \vdots & \vdots & \ddots & \vdots \\ P_{\mathbf{I}_K}(B_1) & P_{\mathbf{I}_K}(B_2) & \cdots & P_{\mathbf{I}_K}(B_M) \end{bmatrix} = \begin{bmatrix} \mathbf{P}(B_1) & \mathbf{P}(B_2) & \cdots & \mathbf{P}(B_M) \end{bmatrix}$$

$\mathbf{P}(B_i)$ is the $i$-th column vector of matrix $\Gamma$ for each element $B_i$. It will be shown that the vectors $\mathbf{P}(B_j)$ and $\mathbf{P}(B_k)$ would have the same direction if elements $B_j$ and $B_k$ represent the same coating concentration. This property will be used for lumping histogram elements with similar coating content together.

The 256×256 histogram in the $t_1$-$t_2$ score space is a fine histogram, which means each histogram element represents pixels having similar color and similar coating content. From equation (3-1), we know color is a function of coating level and imaging conditions and these two factors are independent. Considering two histogram elements having the same coating content $y$ (denoted as $B_j$ and $B_k$) for image $\mathbf{I}$, we will have

$$P_{\mathbf{I}}(B_j) = P_{\mathbf{I}}(y)P_{\mathbf{I}}(\phi_j) \ , \ P_{\mathbf{I}}(B_k) = P_{\mathbf{I}}(y)P_{\mathbf{I}}(\phi_k)$$

where $\phi_j$ and $\phi_k$ are the local average imaging conditions. For all the training images collected over a short period of time, often the only factor changing from image to image is coating concentration distribution while the overall imaging conditions (such as lighting distribution) remain the same. Therefore

$$P_{I_1}(\phi_j) = P_{I_2}(\phi_j) = \cdots = P_{I_K}(\phi_j) = s_j,$$

$$P_{I_1}(\phi_k) = P_{I_2}(\phi_k) = \cdots = P_{I_K}(\phi_k) = s_k$$

in which $s_j$ and $s_k$ are two scalars. So

$$\mathbf{P}(B_j) = \mathbf{P}(y) \cdot s_j,\ \mathbf{P}(B_k) = \mathbf{P}(y) \cdot s_k$$

Therefore, for two histogram elements $B_j$ and $B_k$, which correspond to the same amount of local coating concentration $y$, $\mathbf{P}(B_j)$ and $\mathbf{P}(B_k)$ will have the same relationship with $y$ but different magnitudes.

We can compute the covariance between $\mathbf{P}(B_j)$ and any other $K$-element vectors $\mathbf{z}_1$, $\mathbf{z}_2$:

$$\mathrm{cov}_1(B_j) = \mathrm{cov}\left(\mathbf{P}(B_j), \mathbf{z}_1\right) = \mathrm{cov}\left(\mathbf{P}(y), \mathbf{z}_1\right) \cdot s_j,$$

$$\mathrm{cov}_2(B_j) = \mathrm{cov}\left(\mathbf{P}(B_j), \mathbf{z}_2\right) = \mathrm{cov}\left(\mathbf{P}(y), \mathbf{z}_2\right) \cdot s_j$$

The scalar $s_j$ can be canceled by computing the phase angle of the observed point in the space of $\mathrm{cov}_1$ vs. $\mathrm{cov}_2$.

$$\theta(B_j) = \arg\left\{\mathrm{cov}_1(B_j), \mathrm{cov}_2(B_j)\right\} = \theta(y)$$

If $\theta(y)$ and $y$ has a one-to-one mapping relationship, we can divide $\theta$ into a finite number of bins and each bin should represent a different level of coating concentration.

The choice for $\mathbf{z}_1$ and $\mathbf{z}_2$ is not unique. Obviously, setting $\mathbf{z}_1$ as the average coating level is a natural choice. $\mathbf{z}_2$ is chosen as a function of average coating level, but in such a way that $\theta(y)$ and $y$ can achieve a one-to-one mapping relationship. In this study $\mathbf{z}_1$ and $\mathbf{z}_2$ are chosen as:

$$\mathbf{z}_1 = \begin{bmatrix} \bar{y}_{I_1} \\ \bar{y}_{I_2} \\ \vdots \\ \bar{y}_{I_K} \end{bmatrix}, \quad \mathbf{z}_2 = \left|\mathbf{z}_1 - y^*\right| = \begin{bmatrix} \left|\bar{y}_{I_1} - y^*\right| \\ \left|\bar{y}_{I_2} - y^*\right| \\ \vdots \\ \left|\bar{y}_{I_K} - y^*\right| \end{bmatrix}, \quad y^* = \frac{\max(\bar{y}) + \min(\bar{y})}{2}$$

We can apply the covariance computation for each column of the matrix $\Gamma$ and two covariance vectors (one for $\mathbf{z}_1$ and one for $\mathbf{z}_2$) can be obtained:

$$\mathbf{cov}_1 = [\mathrm{cov}_1(B_1), \mathrm{cov}_1(B_2), \cdots, \mathrm{cov}_1(B_M)],$$

$$\mathbf{cov}_2 = [\mathrm{cov}_2(B_1), \mathrm{cov}_2(B_2), \cdots, \mathrm{cov}_2(B_M)]$$

A phase angle is then computed for each histogram element and an angle vector can be obtained:

$$\boldsymbol{\theta} = [\theta(B_1), \theta(B_2), \cdots, \theta(B_M)]$$

The histogram elements that have similar angle values can be lumped into one bin. Notice that the two covariance vectors and the angle vector all have the same dimension as the $t_1$-$t_2$ histogram; therefore, they can be further shown as 256×256 color coded image as can be seen in the examples.

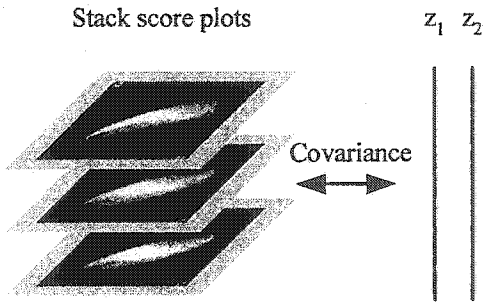
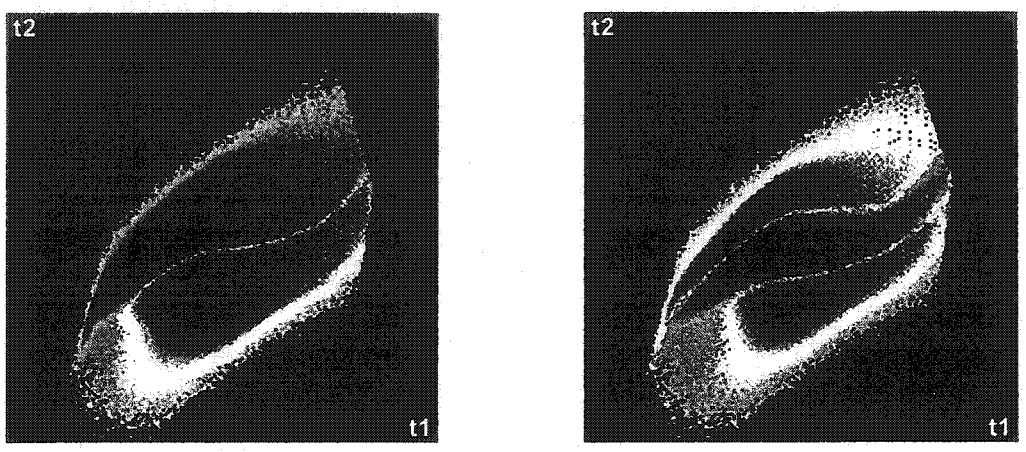



**Figure 3.9 Illustration of stacking score plots and computing covariance plots**




(a) (b)

**Figure 3.10 Covariance plots. (a) Covariance with $z_1$, (b) Covariance with $z_2$**

Method 6 is illustrated with Figure 3.9, 3.10 and 3.11. The two covariance plots shown in Figure 3.10 are obtained by stacking up the 256×256 $t_1$-$t_2$ score plots of the training set and computing the covariance of the number of pixels in each $t_1$-$t_2$ elements with the two variables related to lab coating concentration as shown in Figure 3.9. In the covariance plots, warm colors (red) indicate positive covariance values and cold colors (blue) indicate negative covariance values. Darker shades indicate large absolute covariance values. From the two covariance plots, an angle plot can be calculated as explained above. The $t_1$-$t_2$ elements having similar angles should correspond to pixels in the image having the same coating concentration. Figure 3.11 is the color-coded angle plot divided into 32 bins based on angle values. These 32 bins are called a segmentation mask. Note the similarity of the one dimensional bins obtained by this method and the linear projection approach of methods 4 and 5 shown in Figure 3.7. Now instead of parallel linear bins running from the lower left to upper right in the score space, method 6 gives curved, unequally spaced bins running in essentially the same direction. To use this approach the score plot for any new image can be superimposed on top of this plot and the number of pixels falling into each of these 32 angle bins recorded to form a one-dimensional histogram, as illustrated in Figure 3.12. The resulting histogram and the

cumulative histogram for the three sample images of Figure 3.4 are shown in Figure 3.13. Only the cumulative histogram will be used as the feature vector in the subsequent studies.

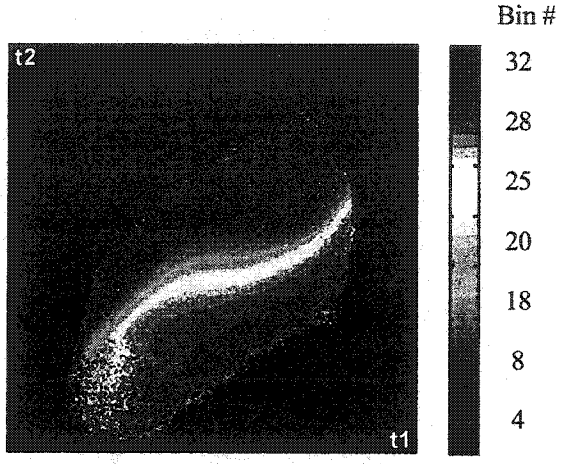
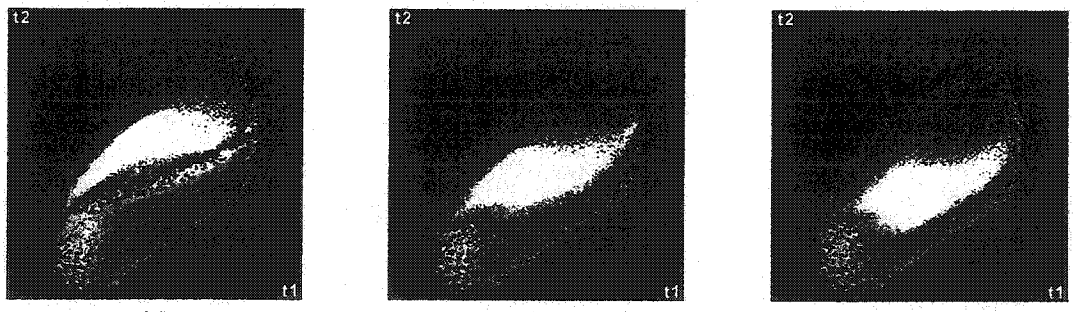



**Figure 3.11 Color coded angle plot in $t_1$-$t_2$ space, with 32 bins based on angle values**




(a) (b) (c)

**Figure 3.12 Superposition of the score plots of the three sample images in Figure 3.4 on top of the angle plot**

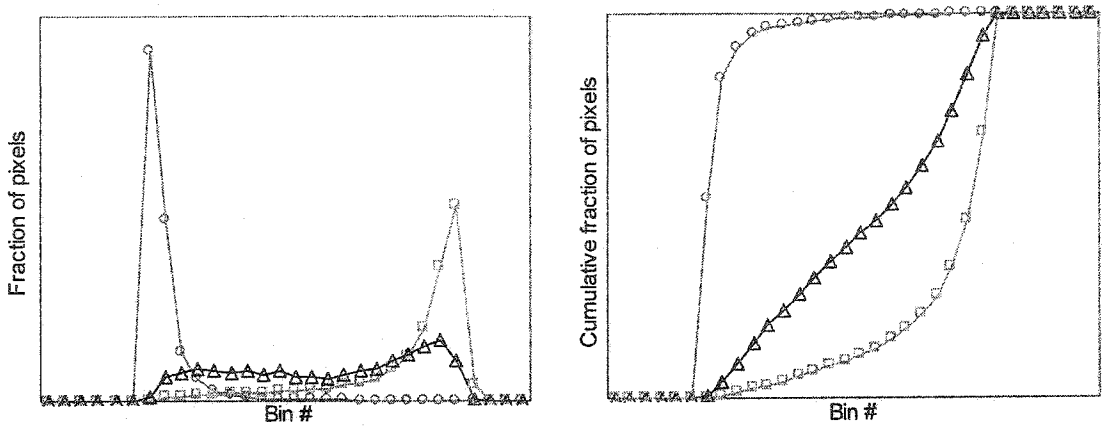



**Figure 3.13 Feature variables of method 6 for the three sample images.**
**(a) Histogram, (b) Cumulative histogram. (○: Uncoated product image; △: Low-coated product image; □: High-coated product image)**

## 3.5 Prediction of Coating Concentration

### 3.5.1 Overall coating concentration

Once feature variables have been obtained by any of the six methods we can build inferential models by regressing these feature variables against the laboratory coating concentration for the training set. PLS regression is used for this. The feature variables are mean-centered and scaled to unit-variance for all PLS models. Transformations are used for the overall features (method 1 and 2) to correct the nonlinearity. For the average color features a logarithmic transformation is used:

$$x_l' = \log(x_l), l = R, G, B$$

Since the loading values have the range between 0 and 1, a logistic transformation is used for the 1st loading vector features (method 2):

$$x_l' = \log(\frac{x_l}{1 - x_l}), l = R, G, B$$

Results using all 6 feature extraction methods for products A, B and C for both the training and testing image sets are shown in Tables 3.3-a, b and c. In these tables the number of feature variables, the number of components used in the PLS regressions, the sum of square prediction errors (SSE) and R-square statistic (the ratio of sum of squares explained by the regression to the total sum of squares) are given. From the results, it is clear that all methods work well and have almost the same performance in all the cases. Moreover, it seems that the simple overall feature variable methods perform as well as the more complex distribution feature models. However, the study in next section does show a large difference in the performance of the six models when predicting local coating concentration of small window images. The models built based on the distribution feature will be seen to be much more robust than those based on the overall features, and among the distribution feature models, method 6 will be seen be the most robust. Figure 3.14 plots the predicted average coating level vs. the lab analysis data for product C using method 6. One can see that both the fit of the training set and the prediction of the test set are excellent.

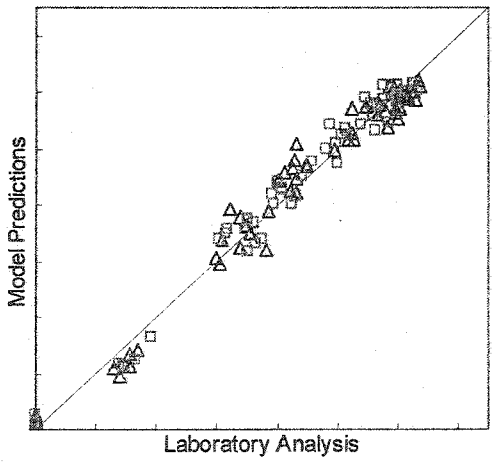



**Figure 3.14 Predicted vs. observed coating content for product C using method 6 (△: Training set; □: Test set)**

**Table 3.3 Model prediction results**

(a) Product A; $SS_T$=377.44(Training set); $SS_T$=385.13(Test set)

| | | Feature variable # | Latent variable # | Variance Analysis | |
|---|---|---|---|---|---|
| | | | | $SS_E$ | $R^2$=1-$SS_E$/$SS_T$ |
| Model 1 | Training set | 3 | 2 | 1.13 | 0.997 |
| | Test set | | | 1.60 | 0.996 |
| Model 2 | Training set | 3 | 2 | 2.25 | 0.994 |
| | Test set | | | 1.59 | 0.996 |
| Model 3 | Training set | 424 | 3 | 1.59 | 0.996 |
| | Test set | | | 3.60 | 0.991 |
| Model 4 | Training set | 32 | 3 | 0.95 | 0.997 |
| | Test set | | | 1.83 | 0.995 |
| Model 5 | Training set | 25 | 3 | 1.34 | 0.996 |
| | Test set | | | 1.18 | 0.997 |
| Model 6 | Training set | 19 | 2 | 0.74 | 0.998 |
| | Test set | | | 0.89 | 0.998 |

(b) Product B; $SS_T$=869.83 (Training set); $SS_T$=845.41(Test set)

| | | Feature variable # | Latent variable # | Variance Analysis | |
|---|---|---|---|---|---|
| | | | | $SS_E$ | $R^2$=1-$SS_E$/$SS_T$ |
| Model 1 | Training set | 3 | 2 | 7.96 | 0.991 |
| | Test set | | | 8.64 | 0.990 |
| Model 2 | Training set | 3 | 2 | 4.58 | 0.995 |
| | Test set | | | 5.42 | 0.994 |
| Model 3 | Training set | 329 | 5 | 2.21 | 0.997 |
| | Test set | | | 5.13 | 0.994 |
| Model 4 | Training set | 32 | 3 | 3.58 | 0.996 |
| | Test set | | | 7.14 | 0.992 |
| Model 5 | Training set | 24 | 2 | 5.11 | 0.994 |
| | Test set | | | 8.72 | 0.990 |
| Model 6 | Training set | 18 | 2 | 5.82 | 0.993 |
| | Test set | | | 5.83 | 0.993 |

(c) Product C; $SS_T$=817.19 (Training set); $SS_T$=751.63(Test set)

| | | Feature variable # | Latent variable # | Variance Analysis | |
|---|---|---|---|---|---|
| | | | | $SS_E$ | $R^2$=1-$SS_E$/$SS_T$ |
| Model 1 | Training set | 3 | 3 | 19.81 | 0.976 |
| | Test set | | | 13.12 | 0.983 |
| Model 2 | Training set | 3 | 2 | 20.09 | 0.975 |
| | Test set | | | 14.50 | 0.981 |
| Model 3 | Training set | 427 | 3 | 12.67 | 0.984 |
| | Test set | | | 18.56 | 0.975 |
| Model 4 | Training set | 32 | 3 | 17.75 | 0.978 |
| | Test set | | | 17.49 | 0.977 |
| Model 5 | Training set | 15 | 3 | 16.29 | 0.980 |
| | Test set | | | 13.66 | 0.982 |
| Model 6 | Training set | 19 | 2 | 20.56 | 0.975 |
| | Test set | | | 13.87 | 0.982 |

### 3.5.2 Local coating concentration of small window images

In this section, predictions of small window images are studied. This study is important because it is the basis for a small window strategy for estimating the coating coverage distribution, which will be explained in section 3.8. It is also important in on-line applications because in many on-line cases the conveyer belt is only partially covered with product and so only part of the image is filled with product and is used for prediction. Therefore, to perform either of these functions the methodology must be robust to the image size.

In this study, an image is divided into $J \times J$ windows (see Figure 3.24), and the models developed in section 3.5.1 for the whole images are then applied for each window. Totally $J^2$ local coating concentrations can then be estimated. The Root Mean Square Error of Prediction (RMSEP) is used as measurement of prediction error. Uncoated product images are chosen for this study. For any window image, the perfect coating concentration prediction should be 0.

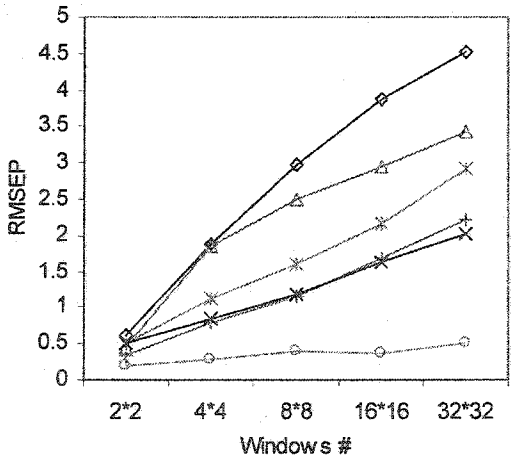



**Figure 3.15 RMSEP against image window size.**
**(◊: Model 1; △: Model 2; *: Model 3; +: Model 4; ×: Model 5; ○: Model 6)**

Figure 3.15 shows the RMSEP at different window sizes for product C using the six feature extraction models. For product A and B, similar results are obtained. From the figure we can see that, as the number of windows increases, the RMSEP increases in all the cases. However, the six methods appear to have quite different performances. Model 6 always has the smallest RMSEP and, even with 32×32 windows, the prediction error is still within an acceptable range. Image size has a greater effect on the overall feature models (method 1 and method 2) than on the distribution feature models. Furthermore, among the overall feature methods, method 1 has a larger prediction error than method 2. The performance of method 3 which uses the two-dimensional histogram in the score space is consistently worse than method 4 and method 5, which robustify the features by projecting out the effect of uncoating related variations. Method 4 and method 5 have similar performance; this is understandable, since the only difference between these two is the use of the histogram and the cumulative histogram.

## 3.6 Discussion

### 3.6.1 Overall feature models

There is a strong relationship between the average color and the first loading vector of PCA as used in overall methods 1 and 2. In method 2, a non-mean-centered PCA is performed on each image. In this situation, the direction of average color $\bar{\mathbf{c}}$ is approximately equal to the direction of first loading vector $\mathbf{p}_1$.

$$\bar{\mathbf{c}} \cdot \mathbf{p}_1 \approx \|\bar{\mathbf{c}}\| \Rightarrow \mathbf{p}_1^T \approx \frac{\bar{\mathbf{c}}}{\|\bar{\mathbf{c}}\|}$$

Therefore, the first loading vector is approximately equal to the normalized average color. Instead of first loading vector of PCA, the normalized average color could have been used as the feature variables and should give similar performance as method 2.

Lighting variations will have an influence on the overall feature methods. Lighting variation comes from a non-uniformly distributed light source, a non-flat orientation of the product, or overlapping among the pieces of product. Suppose that the lighting conditions and color has the following linear relationship for each pixel location:

$$\mathbf{c}_i = L_i \cdot \mathbf{r}_i,\ i = 1, \cdots, N$$

in which, $\mathbf{c}_i$, $L_i$, $\mathbf{r}_i$ are the measured color, the local lighting intensity and the color under an ideal reference lighting condition for pixel $i$.

Assume that the light source and the image scene are independent. The average color of an image is equal to

$$\bar{\mathbf{c}} = \bar{L} \cdot \bar{\mathbf{r}}$$

Any variation in lighting will directly influence the value of average color.

Since the first loading vector is approximately the normalized color, then

$$\mathbf{p}_1^T \approx \frac{\bar{\mathbf{c}}}{\|\bar{\mathbf{c}}\|} = \frac{\bar{L} \cdot \bar{\mathbf{r}}}{\|\bar{L} \cdot \bar{\mathbf{r}})\|} = \frac{\bar{\mathbf{r}}}{\|\bar{\mathbf{r}}\|}$$

Therefore, the lighting effect is canceled when using the first loading vector or normalized average color. In section 3.5.1, test set images have similar overall lighting conditions as the training set images, therefore both method 1 and method 2 give good prediction results. But when using small image windows as test images as in section 3.5.2, the lighting conditions vary from window to window. Since method 2 is less sensitive to lighting variation, it always has a smaller prediction error than method 1. The above analysis is based on a linear effect of lighting and the assumption that lighting intensity has the same effect on three channels, none of which are true in reality, and so method 2 is still influenced by lighting condition variations as illustrated by the increasing prediction error with the window number in Figure 3.15.

### 3.6.2 Distribution feature models

Distribution feature models can always be rearranged into the form:

$$y = \hat{y}_{Bin_1} x_1 + \hat{y}_{Bin_2} x_2 + \cdots + \hat{y}_{Bin_B} x_B$$

where $x_i$ is the relative histogram counts for bin $i$ and $\hat{y}_{Bin_i}$ is the coefficient for $x_i$. This means for a pixel with color value $\mathbf{c}$, if we know that this color value falls into $i$-th bin, the coating concentration for this pixel can be estimated as $\hat{y}_{Bin_i}$. Therefore, distribution feature models can be seen as pixel-level models because technically we can estimate the coating concentration for each pixel. To determine whether any of these distributional models are capable of predicting a reasonable coating concentration for local pixels, it is necessary to examine the estimated local coating concentration for each histogram bin for each of them (see Figure 3.16).

In Figure 3.16, we can see that the estimated coating concentration for method 6 is more reasonable than for method 3, 4 and 5, because it is impossible that the coating concentrations have negative values, and the resulting shape of the distribution is much closer to what is expected.

When predicting the images that have similar size as training images, errors in the estimated coating concentration for each bin tend to be averaged out because of the large number of pixels. However, as the image size becomes smaller, prediction errors increase for models having poorer estimation of coating concentrations at the pixel level.

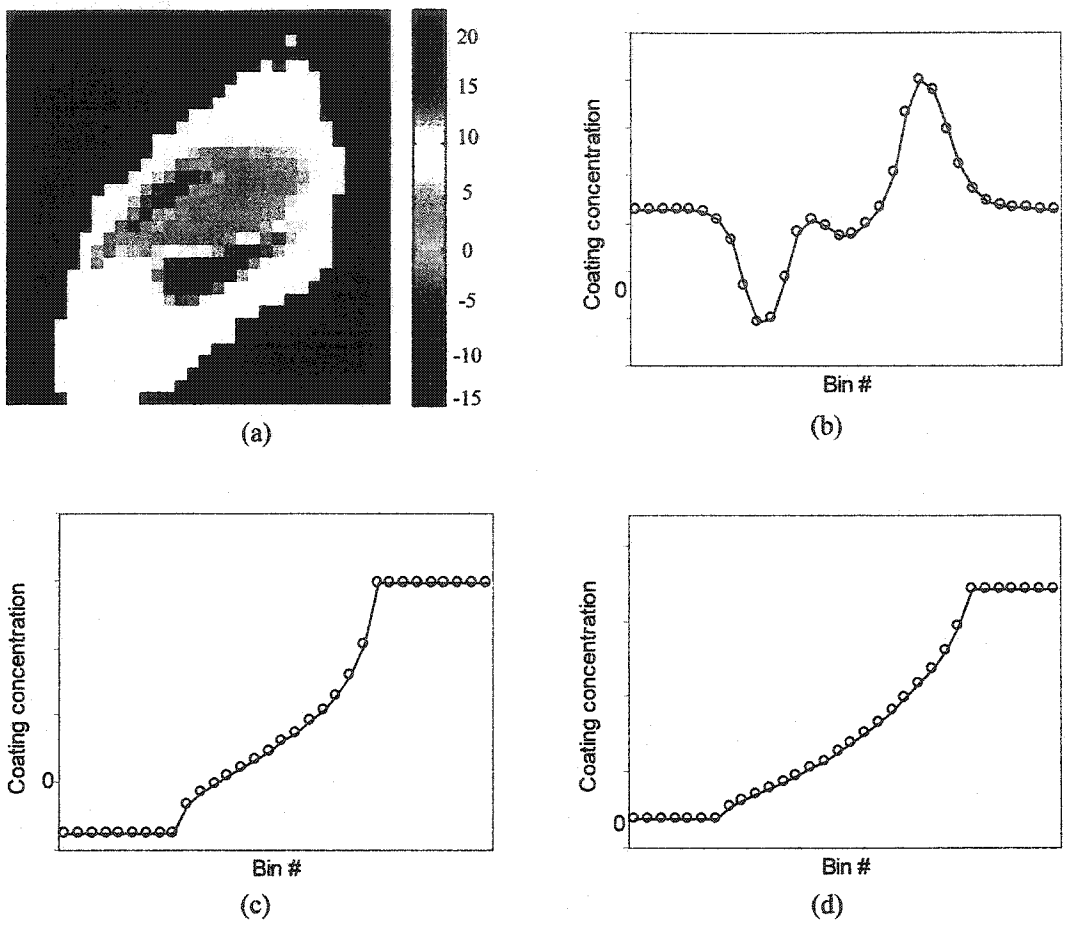



**Figure 3.16 Estimated coating concentration for each histogram bin obtained from the distributional models**

## 3.7 Foreground/Background Segmentation

In this section, we discuss an important step that must be performed prior to feature extraction: background removal. Background mentioned here arises from exposed areas of the table or pan on which the product is placed (off-line, product A case) or from exposed conveyor belt (on-line, product B and C cases) (see Figure 3.17).

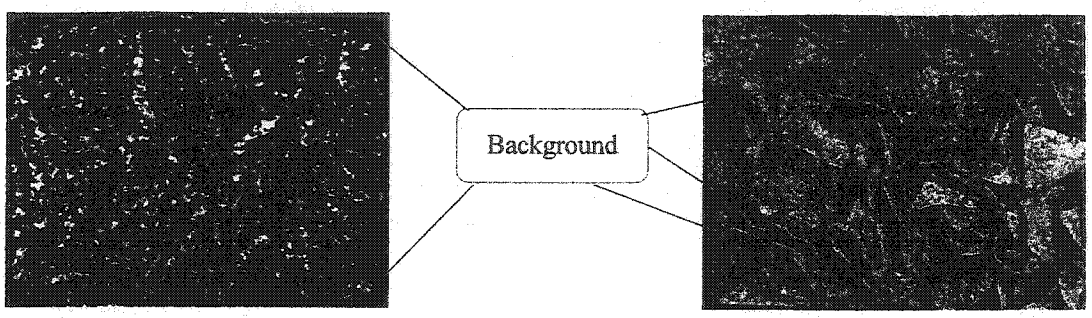
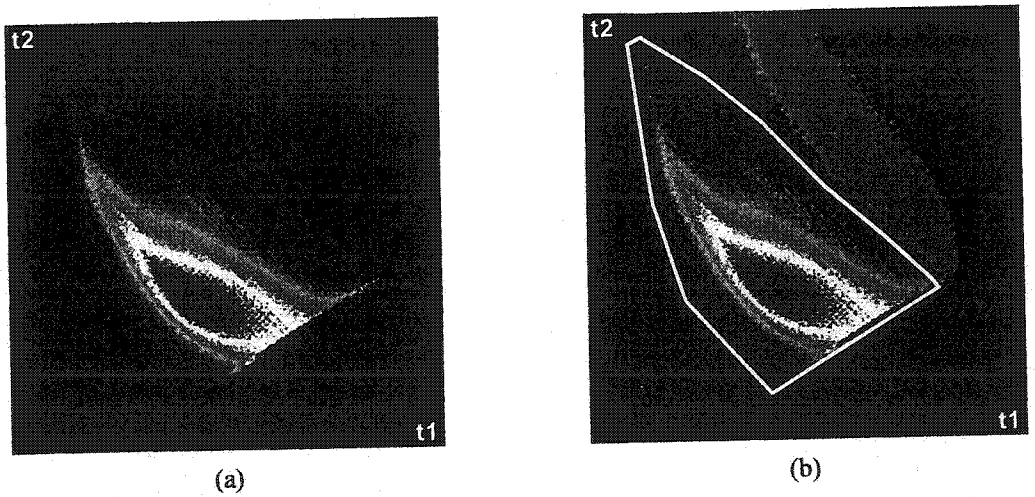



**Figure 3.17 Background in images**

An approach based on spectral feature (color) differences between the snack product and the background is presented in this section, using product C as the example. A PCA (without mean-centering) is first performed to reduce the dimension in which PCA loading vectors are computed using all the training images and a *pure belt image*. As in traditional MIA techniques (section 2.5), masks are chosen in the $t_1$-$t_2$ score space to separate different features. However, in this case masks are not easily chosen by a trial-and-error process, as has been traditionally done in MIA, because: 1) in some images, belt pixels can easily be overlooked by human vision because of the dark light conditions and misclassified as snack product pixels; 2) there is not only one image to be studied but a set of images and manual trial-and-error approaches on each image is too time consuming.

To solve these two problems, a new procedure is presented. After a $t_1$-$t_2$ plane is obtained by PCA, $t_1$-$t_2$ score plots are stacked together for all training images; then two covariance plots (with $\mathbf{z}_1$ and $\mathbf{z}_2$) can be computed as in method 6 (section 3.5.1). Since the background is independent of the coating concentration, histogram counts of a background color should have low values in both of these covariance plots. Therefore, we can use the summation of absolute values of these two covariance plots to locate the pixels that have low total covariance values. Such a summation of the covariance plots is color-coded and shown in Figure 3.18a (cold colors mean small covariance values and warm colors mean large values). Moreover, the projection of the pure belt image onto this

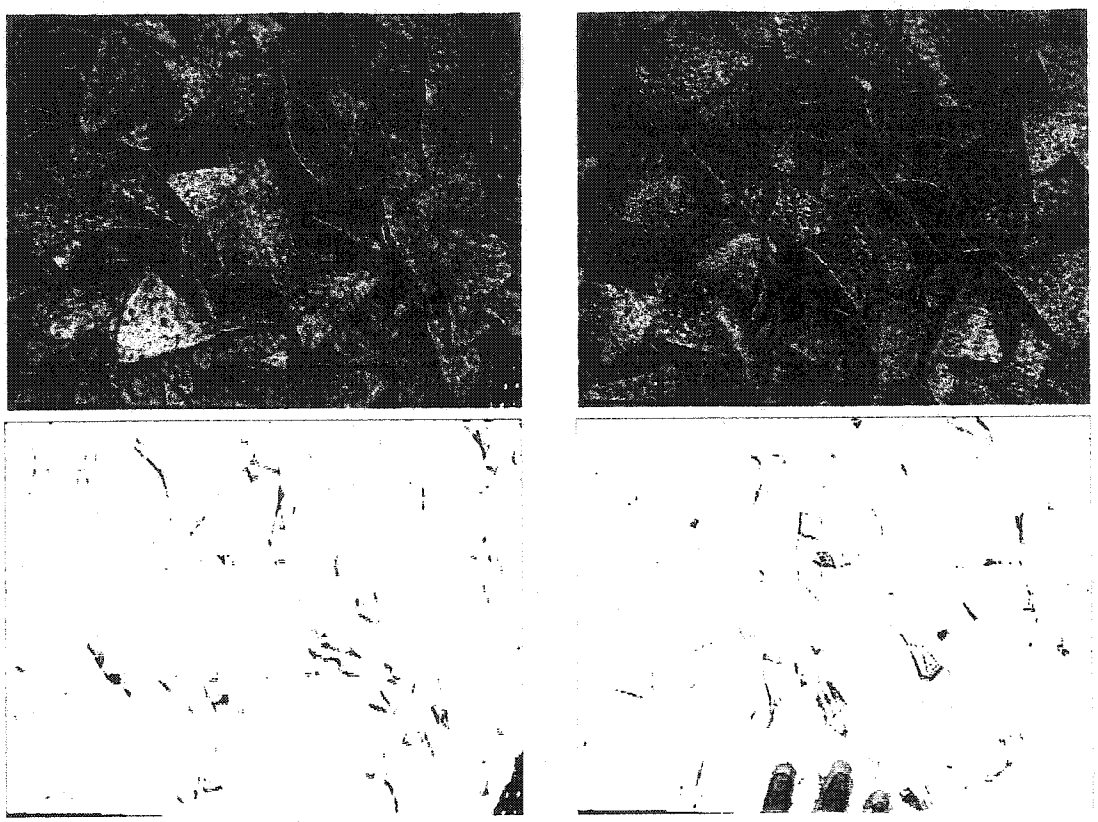
$t_1$-$t_2$ space can also help to locate belt pixels (shown as purple pixels in Figure 3.18-b). A product mask (the area within the white polygon in Figure 3.18-b) can be then decided based on the covariance information and location of pure belt image pixels. For the new images, score pixels falling outside the product mask are considered belong to background. The advantages of this method is when calculating covariance plots we have taken into account the information from all the training images and the approach is not as subjective nor is it influenced by the dark lighting conditions that could influence human visual judgment.




(a) (b)

**Figure 3.18 Selection of product mask (a) Color-coded summation of absolute covariance plots (cold colors represent small values, warm colors represent large values) (b) Purple pixels show the location of belt image; white polygon is the product mask. Pixels falling outside the mask are background**

The performance of the proposed background removal technique is illustrated with two examples showed in Figure 3.19. It is interesting to see that this technique is able to remove not only the expected background (belt in Figure 3.19a) but also the unexpected background (fingers in Figure 3.19b).



(a) (b)

**Figure 3.19 Background segmentation example. The first row shows original two images and the second row shows the background that separated. (a) Remove belt pixels; (b) Remove fingers pixels**

## 3.8 Coating Coverage

### 3.8.1 Estimation of coating distribution by directly using method 6

Since method 6 can predict the coating concentration for each pixel, an estimation of the coating concentration distribution can be obtained directly. In order to avoid pixel-by-pixel computation and to obtain a smoother result, the coating distribution is computed by combining the cumulative histogram from method 6 (Figure 3.13b) and the estimated coating concentration for each bin of that histogram (Figure 3.16d). The resulting 32 bin cumulative coating concentration distribution is then equally re-sampled and smoothed by a central moving-average smoother as shown in Figure 3.20a for the 3

sample images of product C in Table 3.1. The coating distribution can be obtained from this as shown in Figure 3.20b.

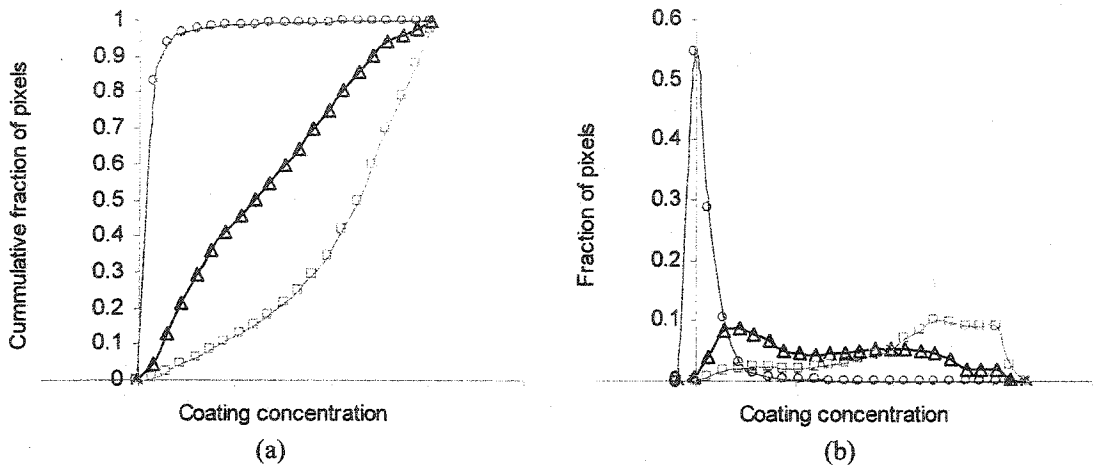



**Figure 3.20 Coating distribution obtained directly by method 6 for three sample images of product C (a) Re-sampled and smoothed cumulative distribution (b) Coating distribution plot**

A color-coded image can also be generated based on the coating concentration of each pixel. Figure 3.21 shows the color-coded sample images for the same three samples. These color-coded sample images are very convenient for visually monitoring of the processes.

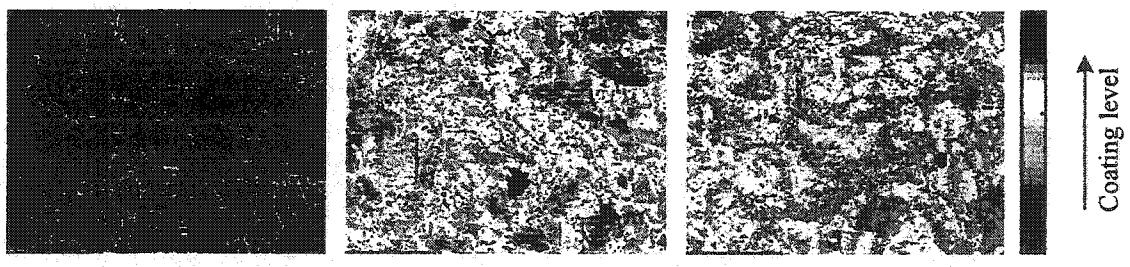



**Figure 3.21 Color-coded coating concentration images for visually monitoring**

### 3.8.2 Small window strategy

From Figure 3.20b, one would think that both the high-coated product image and the low-coated product image have coating distributions that are very rough and broad.

However, visual inspection of the color coded images in Figure 3.22 shows that the high-coated product looks more uniform than low-coated product. The reason for this conflict is that the sense of uniformity or non-uniformity from a visual or taste viewpoint is not at the pixel level but at the level of a product piece. Therefore, another reasonable estimate of the coating coverage distribution is obtained by using a small window strategy, which was mentioned in section 3.5.2. In the small window strategy each image is divided into many small windows and the coating distribution can be obtained by calculating average coating concentration for each window. Method 6 is used for this small window image prediction because only method 6 is image size insensitive among all the methods presented in section 3.4.

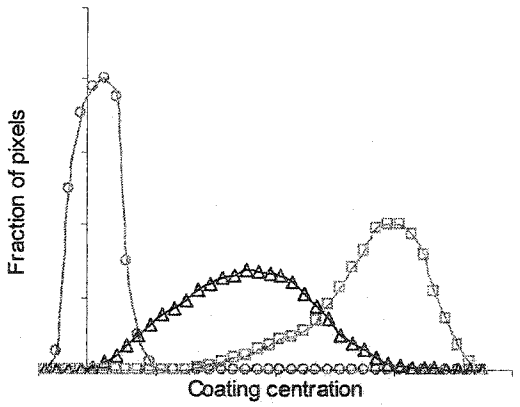



**Figure 3.22 Coating distributions obtained by small window strategy for three sample images of product C**

Figure 3.22 shows the resulting coating distribution obtained for the 3 sample images of product C in Table 3.1 using 20×20 windows. We can compare this figure with Figure 3.20b. From Figure 3.22, we see that the small window approach gives much smoother distributions that are centered more tightly about their mean coating values. It was felt that these distributions were more consistent with what was believed to be the truth based on visual experience with the product. The variance of the coating distribution

is a measurement of the uniformity of the coating coverage and can then be used for on-line monitoring and control purposes.

In the small window strategy, it is important to choose a proper window size. If the window size is too large, spatial distribution information may be lost; on the other hand, if the window size is too small, computation time would increase and the variance of the estimates would increase (as in the pixel by pixel approach in the previous section).

An image with mixed products (uncoated products together with high-coated products) is used to study the effect of different image window sizes. The image in Figure 3.23 is divided into 10×10, 20×20 and 32×32 windows. The estimated local coating concentration is shown by color-coded images (the same color map is used as in Figure 3.21). The resulting coating distribution estimates are shown in Figure 3.24. When using 10×10 windows, the two maxima are not as clear as using 20×20 and 32×32 windows. There is not much difference between the 20×20 and 32×32 cases. Since 20×20 can already capture enough of the coating distribution features, it can be chosen as a practical window size for this application.

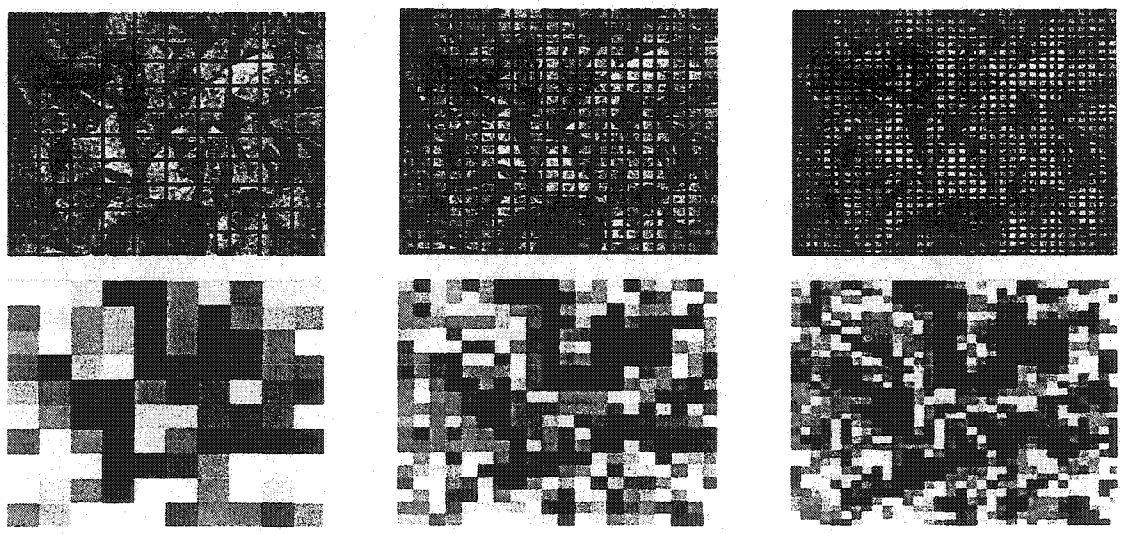


**Figure 3.23 Window size study. In the first row, windows are shown overlapped with original image. In the second row, color code images are shown to indicate estimated local coating concentration**

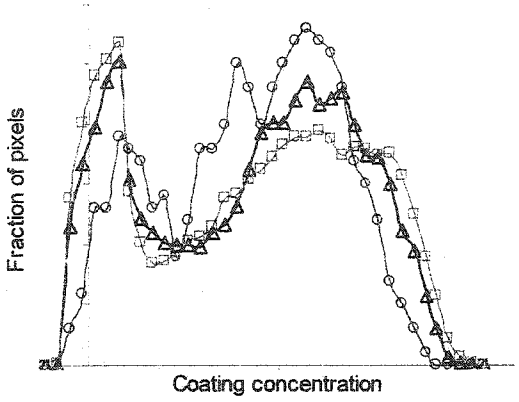



**Figure 3.24 Coating distribution predicted different window sizes.
(○: 10×10; △: 20×20; □: 32×32)**

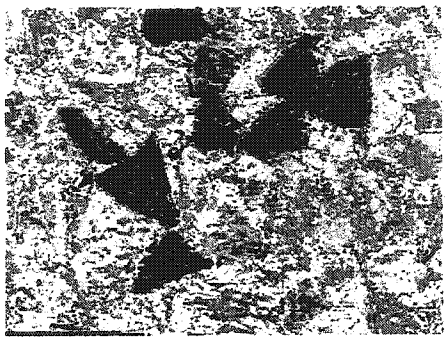


**Figure 3.25 Color-coded concentration predictions for a mixed product image using the pixel level method of section 3.8.1**

Finally, it is interesting to show the color-coded image (Figure 3.25) using the method of section 3.8.1. Compared to the color-coded images in Figure 3.23, Figure 3.25 indicates more clearly the location and shape of uncoated product. However, the coating distribution estimated by that approach appears to be less representative.

## 3.9 Conclusions

In this chapter practical methodologies for building inferential sensors to measure coating concentration and coating coverage distribution on snack food products are successfully developed. Six approaches are proposed to predict the average coating level for snack foods based on six feature extraction methods (2 overall features and 4 distribution features). Samples of 3 types of products (product A is collected off-line, while product B and C are collected on-line) are used to build models and to verify the model predictability. Prediction results show that all six methods have excellent performance. However, the distribution feature methods, and in particular method 6 is shown to be much more robust in predicting small size images. This latter method is

currently being used for on-line monitoring and feedback control in several production lines.

Background/foreground separation is also studied. A method is presented for choosing a product mask in the score plot space whereby the background can be easily and efficiently detected and separated from foreground pixels.

Method 6 can be directly used for estimating the distribution of coating on the product at the pixel level and to provide good visual presentation of the distribution. However, estimates of the coating distribution that are more consistent with human experience were obtained at the level of the product piece size using a small window strategy. The standard deviation computed from this latter approach is currently being used on-line.

These approaches have been evaluated on all three types of snack food products and have been implemented for monitoring and for feedback control. Results of these on-line studies are presented in Chapter 4.

# Chapter 4

# Snack Food Application (Part II) — On-line Results

Some on-line results of using the methodologies developed in Chapter 3 are presented in this chapter. Preliminary trial results are first shown to illustrate the power of the approach for monitoring and fault detection, and for tracking coating concentration over a series of manipulated variable changes. On-line results from the commissioned system are then presented to show its ability to predict coating concentration and distribution under a series of process changes, and to show its direct use in the closed-loop control of coating concentration.

## 4.1 Process Set-up

A schematic of the processes and the imaging systems is shown in Figure 4.1. Coating powders or slurries and uncoated products are mixed in a tumbler and the coated products are then conveyed via a moving belt to the subsequent operations. An RGB camera and lighting system is mounted above the moving belt and images are sent to a computer through a frame grabber. The choice of sampling interval depends upon the objectives for each process. The time required for analyzing one image is less than a second.

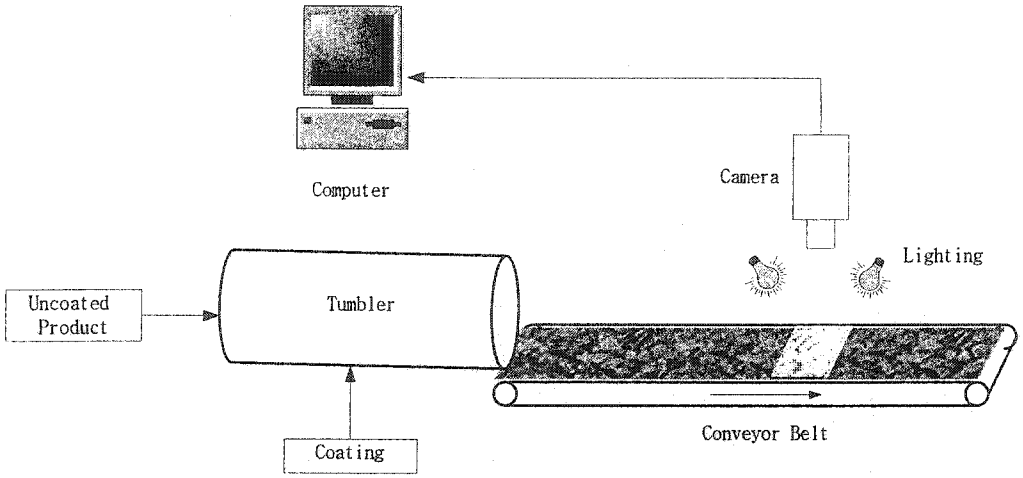



**Figure 4.1 A schematic of the processes and imaging systems**

## 4.2 Implementation of the Model

Six methods were proposed for feature extraction in Chapter 3. However, only method 6 is insensitive to image size and therefore is the one implemented.

To obtain a prediction for a new image we first preprocess the image to remove any background pixels. This is done by projecting the color value of the pixels onto the $t_1$-$t_2$ plane obtained by the first PCA model (performed on the training images and a pure belt image, section 3.7), and then remove the pixels falling outside the product mask (see Figure 3.18). After removing background, the remaining product pixels are projected onto the $t_1$-$t_2$ plane obtained by the second PCA model (section 3.4.2). By counting the number of pixels falling into each of the bins (Figure 3.11), we obtain the 32 bin cumulative one-dimensional histogram to be used as predictor variables in the PLS model (section 3.5). Average coating level is then predicted by the PLS model. As explained in section 3.8.2, a small window strategy is used to estimate the coating concentration distribution whose variance can be used for on-line monitoring and control. Figure 4.2 shows a summarized flow chart.

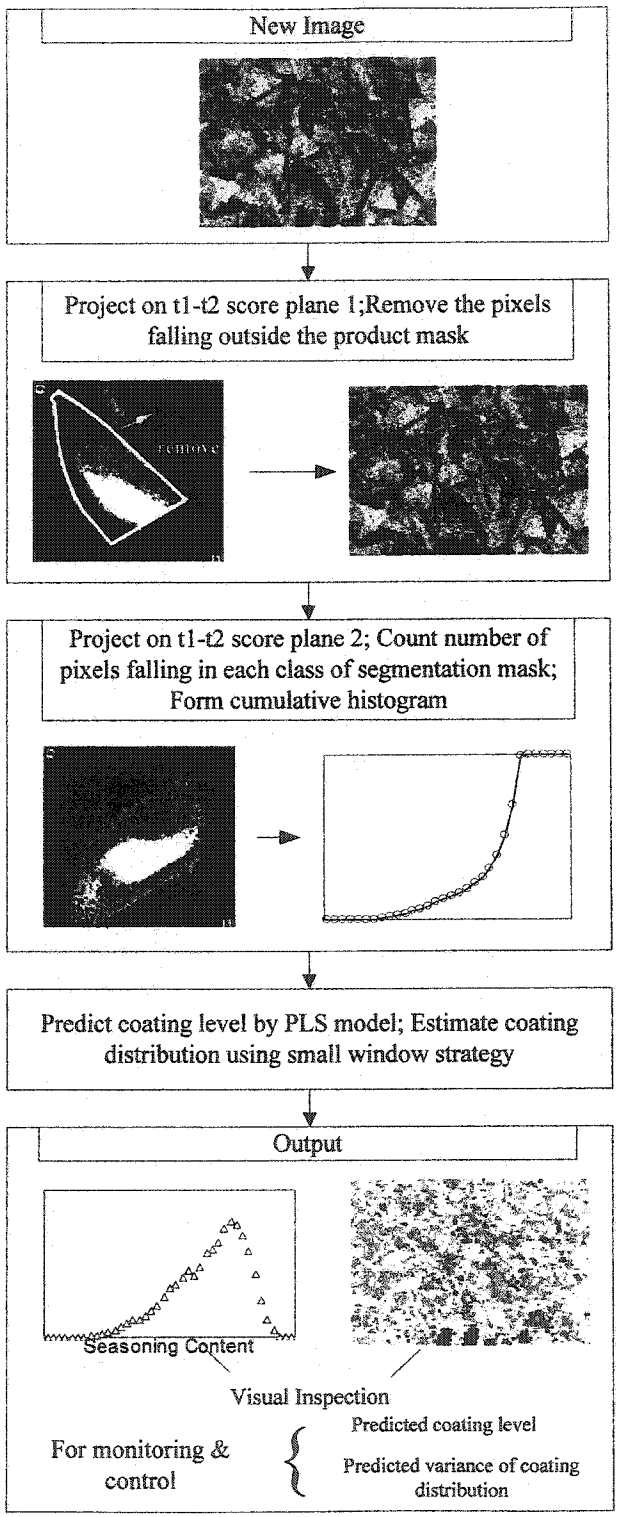



**Figure 4.2 Flow chart for predicting a new image**

## 4.3 On-line Results

Results are now presented to illustrate the abilities of this vision-based inferential sensor. The first two examples provide an assessment of the system's potential to track the coating concentration on-line under different conditions. They represent results obtained on the first day of implementation of the image processing algorithm. The subsequent studies illustrate typical results from using the system for routine monitoring and closed-loop control.

### 4.3.1 Prediction of coating level and detection of hopper effect on coating level (Product C)

Some of the first data collected from the imaging system are shown in the upper plot of Figure 4.3. The raw coating predictions are shown by the light gray line. Frequent grab samples were taken every 5 minutes and analyzed later in the lab during this evaluation study. These are shown as circles in the Figure 4.3. A Y-error bar and an X-error bar are also shown for each lab measurement. The Y-error bar indicates the sampling and measure error (1 standard deviation) estimated from replicated sampling and analysis data (see Appendix B). The X-error bar indicates the possible sample time mismatch (±1 min) between taking the digital images and manually grabbing the product sample.

One can see that the predicted coating concentrations from the images are in good agreement with the lab analysis. However, the image predictions reveal a clear saw-tooth behavior in the concentration that is not very evident only from the lab data, even during this fast sampling program (every 5 minutes; in normal operation the sampling time is about every half an hour). This unexpected result was explained by the coating hopper refilling operations in the process. The lower plot in Figure 4.3 shows the signal of the motor of the hopper refilling system. As the level of the coating powder in the feed hopper falls to a certain low level, the motor is activated to refill the hopper. The level of coating inside hopper then increases rapidly. Clearly Figure 4.3 shows that the discharge

of rate of coating from the hopper to the coating operation (tumbler) is a strong function of the coating level in the hopper.

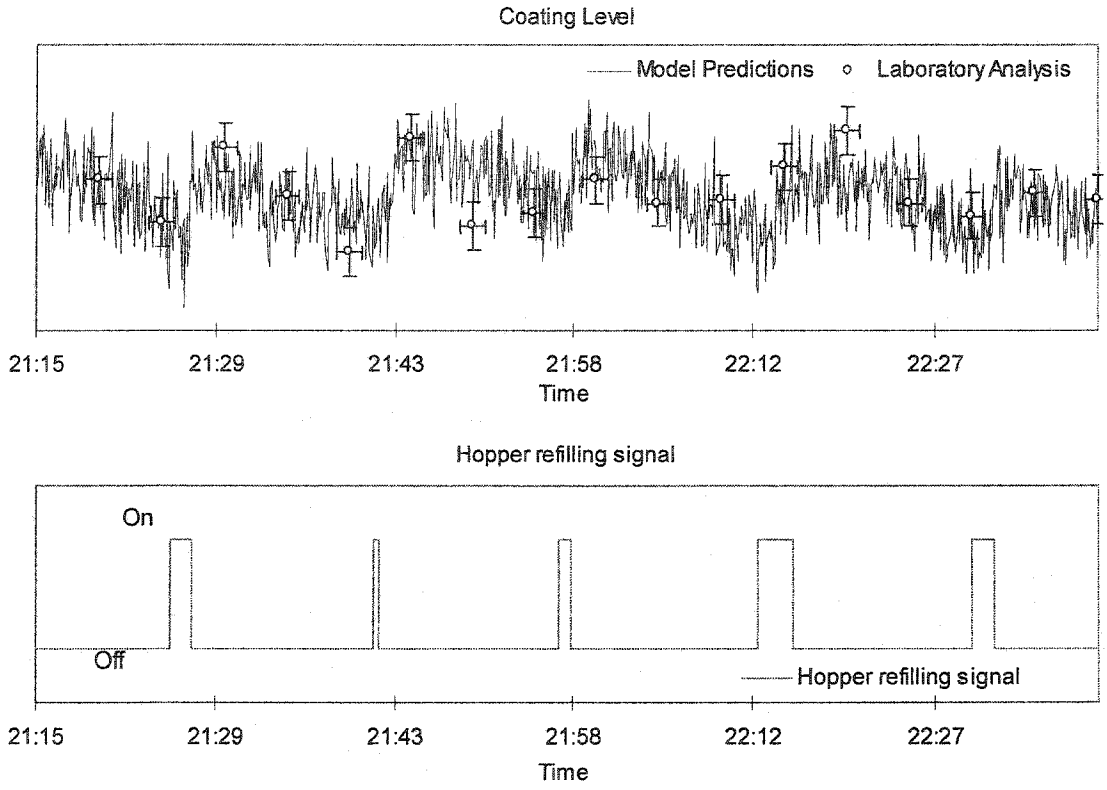



**Figure 4.3 Detection of hopper effect results**

### 4.3.2 Open-loop experiment (product C)

In Figure 4.4, the open-loop response of coating level caused by changing the coating level bias (manipulated variable) is shown. The prediction and lab analysis coating levels are shown in the upper plot, and in the lower plot the coating level bias signal is shown. Again we see that the predictions from the image analysis and the lab measurements are consistent. The predicted data shows a clear fast response to each coating level bias change.

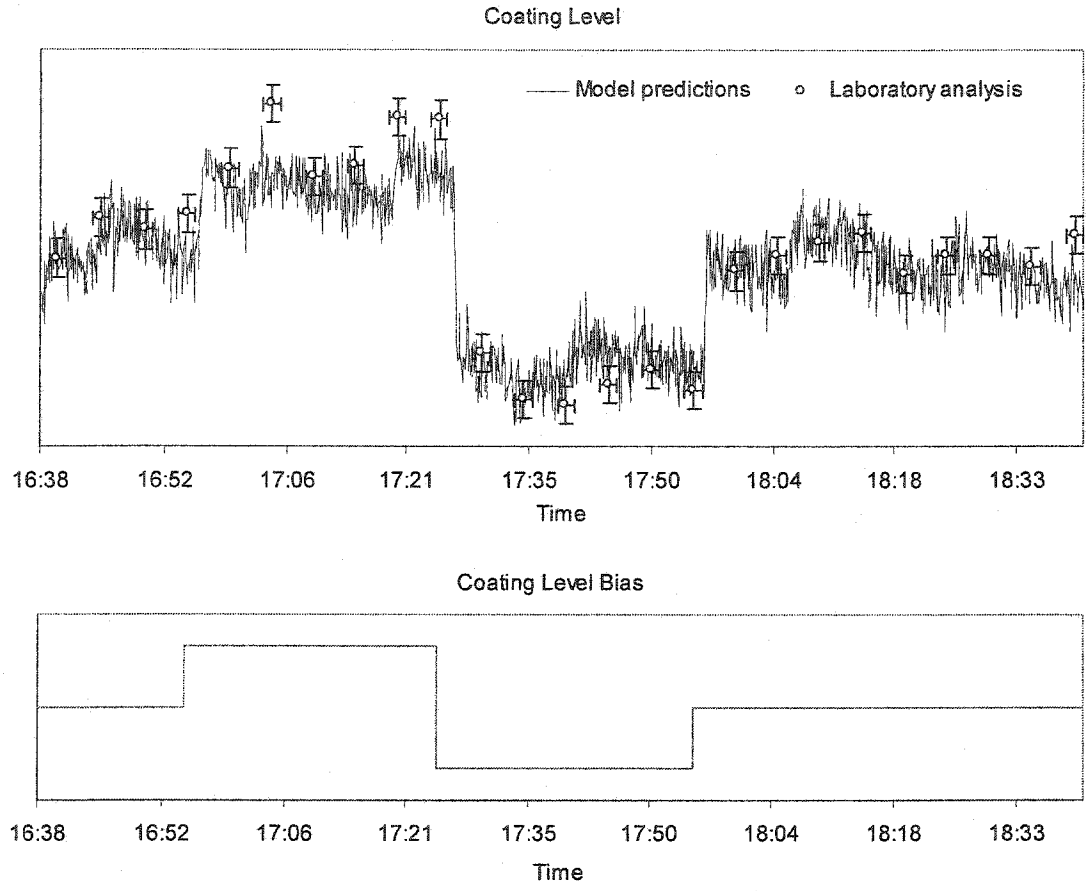



**Figure 4.4 Open-loop experiment results**

### 4.3.3 On-line monitoring examples (Product C and Product B)

Figure 4.5 shows one 4 hour on-line monitoring period for product C. Shown are the predicted coating level, the uncoated product weight, the coating feeder speed and the signal of the dump gate. During this period, at about time 19:35, the feed rate of the uncoated product to the tumbler suddenly increased and, as the result of feed forward ratio control, the coating feeder speed also increased. However, the coating feeder speed was limited by its maximum capacity and could not feed coating fast enough to keep the desired ratio to the uncoated product. Therefore, the coating level on the product decreased from its desired value. A second problem started at about time 20:40 where the

coating level suddenly started to continuously decrease. This occurred because the coating hopper was not set to automatic refilling mode and was being depleted of coating. The result was that eventually no coating was being fed to the tumbler the dump gate had to be opened to remove the uncoated products from the belt. It should be pointed out that by only looking at the process data (uncoated product weight and coating feeder speed) this fault was not detectable.

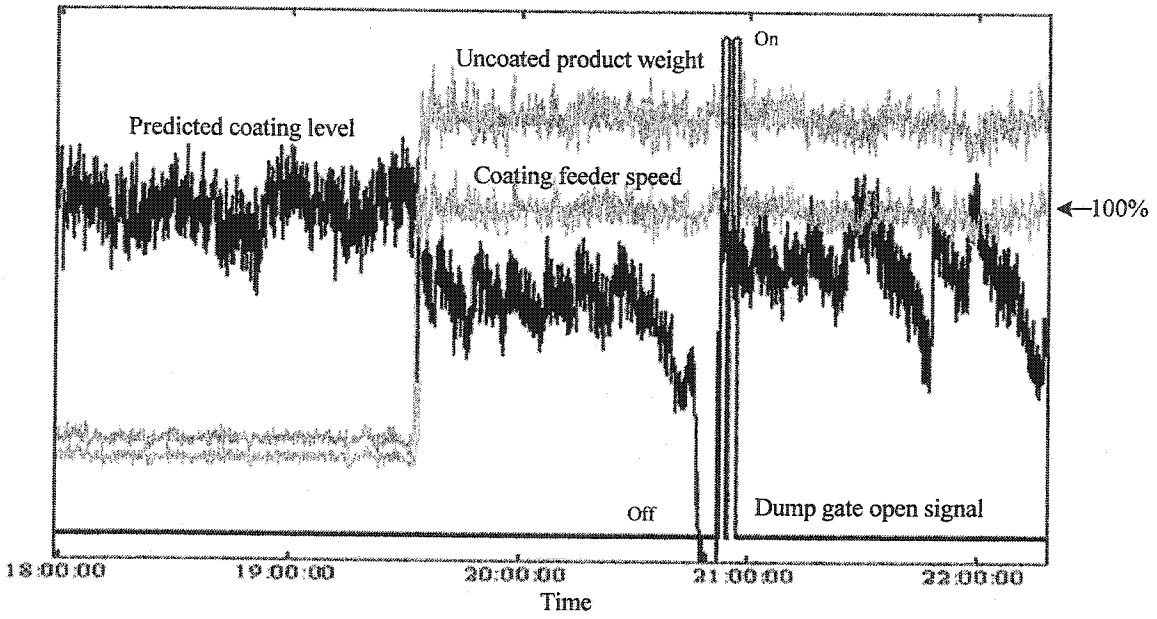



**Figure 4.5 On-line monitoring for product C**

Figure 4.6 shows one example of on-line monitoring both the coating concentration and the variance of the coating distribution for product B. Shown in this figure are the predicted coating level, the uncoated product weight, the coating slurry feed rate and the predicted coating distribution variance. The variance of the coating distribution was calculated from the histogram of the coating concentrations obtained by applying the small window strategy. During this approximate 1 hour period of time, there was no large change in the feed rate of uncoated product. However we see that between time 18:19 to time 18:23, the coating slurry feed rate suddenly dropped to zero. The effect of this disturbance on the coated product appeared after about 8 minutes of time

delay. At about time 18:27, both the predicted coating level and the coating distribution variance began to show large variation. To further understand what happened in the process during that time, Figure 4.7 shows eight images, corresponding to the eight numbers marked in Figure 4.6. The predicted coating concentration drops off rapidly up to image #2 and then recovers in an irregular manner. The variance of the coating distribution decreases at image #2 since by this time most of the products are uniformly uncoated. During the subsequent irregular recovery, shown in image #3 through #8 of Figure 4.7, the predicted variance of the coating distribution (Figure 4.6) is seen to vary widely as slugs of uncoated and coated product appear intermingled on the belt. A bimodal coating distribution would therefore be expected for these images. Figure 4.8 shows such a bimodal distribution for image #5. Both the coating level and the coating variance appear to have recovered from the upset by image #8.

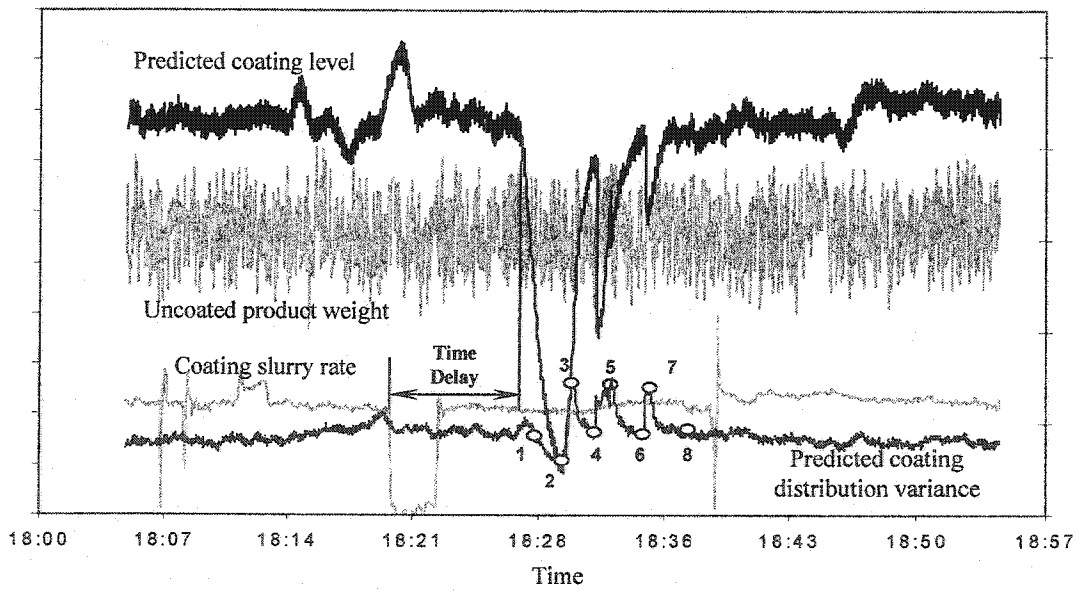



**Figure 4.6 On-line monitoring of product B**

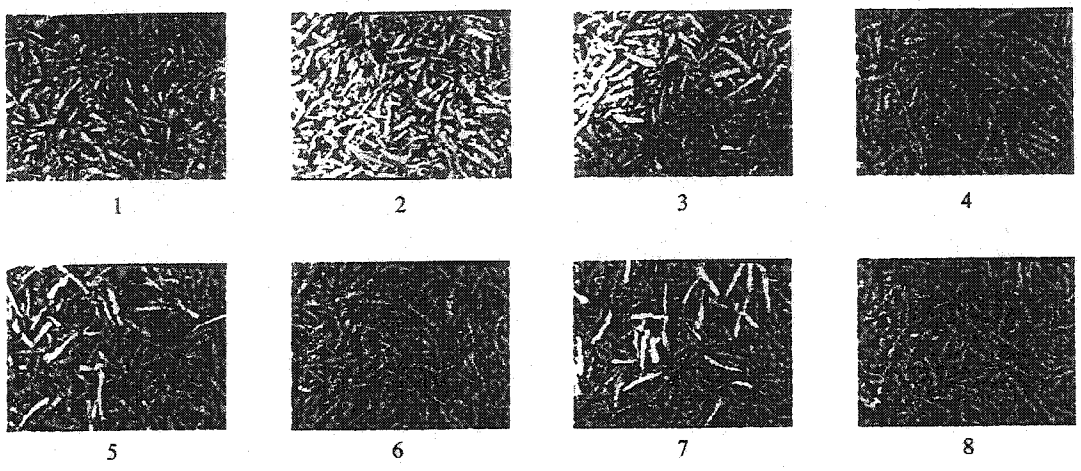



**Figure 4.7 Images corresponding to the numbers marked in Figure 4.6**

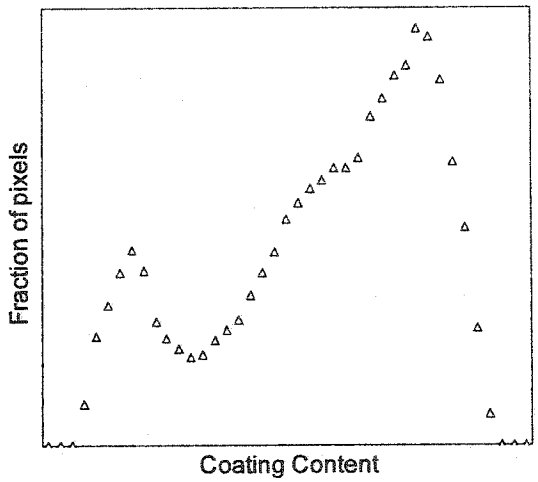



**Figure 4.8 Coating distribution for image #5 shown in Figure 4.7 estimated by using the small window strategy (20×20 windows)**

### 4.3.4 Closed-loop control (Product C)

The inferential sensor developed using method 6 of Chapter 3 has been used for closed-loop feedback control. The block diagram of the control system is shown in Figure 4.9. In this control system, both feedforward and feedback controls are used. The

feedforward control uses a ratio controller to compensate for the large variation of the uncoated product weight, which is the main disturbance source. In the feedback loop, coating level bias (its open-loop response has been tested in section 4.3.2) is computed from the feedback controller and used to adjust the coating feeder speed (manipulated variable). In traditional operation, the feedback control is performed manually by operators based on the sample measurements of samples analyzed in laboratory. By using vision-based inferential sensor, an automatic controller (PI controller in this case) can be used for feedback control.

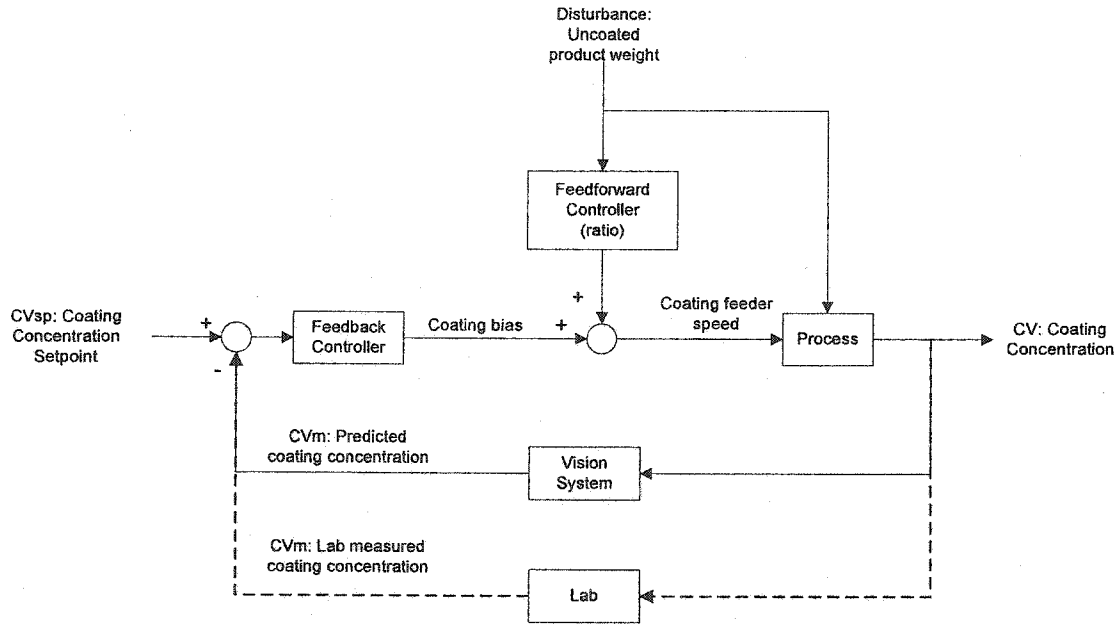



**Figure 4.9 Block diagram of the control system for snack food process**

Variations in the coating level for three cases are compared in Figure 4.10 for Product C. These cases are: 1) without automatic feedback control; 2) with automatic feedback control, set point remaining constant; and 3) with automatic feedback control, set point changing. Each case covers a period of 20 hours. The predicted coating concentration (black lines) and the set point (gray lines) are shown in the first row. For case 1 where no automatic feedback control is performed, the gray line indicates the average value over this period of time. The production line often stops once in a while, so

the plots in the second row indicate the time when production line was stopped. Only the data when the line was in production is used for computing the mean absolute error (MAE) shown in Table 4.1. The changes of the feedback controller output, 'coating level bias', are shown in the third row of Figure 4.10. The variation in the uncoated product weight, the process disturbance, is shown in the fourth row of Figure 4.10.

It is evident that by using automatic feedback control, the variation in the coating level has been greatly reduced and the coating level is successfully tracked to the set point while the disturbance is rejected. The values of the mean absolute error (MAE), listed in Table 4.1, are consistent with this conclusion.

**Table 4.1 Mean Absolute Error for the three cases shown in Figure 4.10**

| | Without Automatic Feedback Control | With Automatic Feedback Control (constant set point) | With Automatic Feedback Control (set point tracking) |
|---|---|---|---|
| MAE | 0.8523 | 0.4481 | 0.4769 |

## 4.4 Conclusions

The methodologies developed in Chapter 3 have been implemented on several production lines. The first two examples provide an assessment of this vision-based inferential sensor. Results show that the predicted data is consistent with the measurements from the laboratory. A saw-tooth behavior in the coating concentration, caused by the coating hopper refilling operation, can be seen clearly in the model predictions because of the high sampling rate of the vision system. The predicted data shows a fast response to the changes in the manipulated variable during the open loop experiment.

Two on-line monitoring examples are presented for two different products. The results show the ability of the vision system to predict the coating concentration and

coating coverage distribution under a series of process changes. The faults in the process can be easily detected by using such a system.

Automatic feedback control based on this inferential sensor was implemented and comparison of the performance, with and without automatic feedback control, is reported. The results show that the variation of the coating concentration can be greatly reduced by using automatic feedback control.

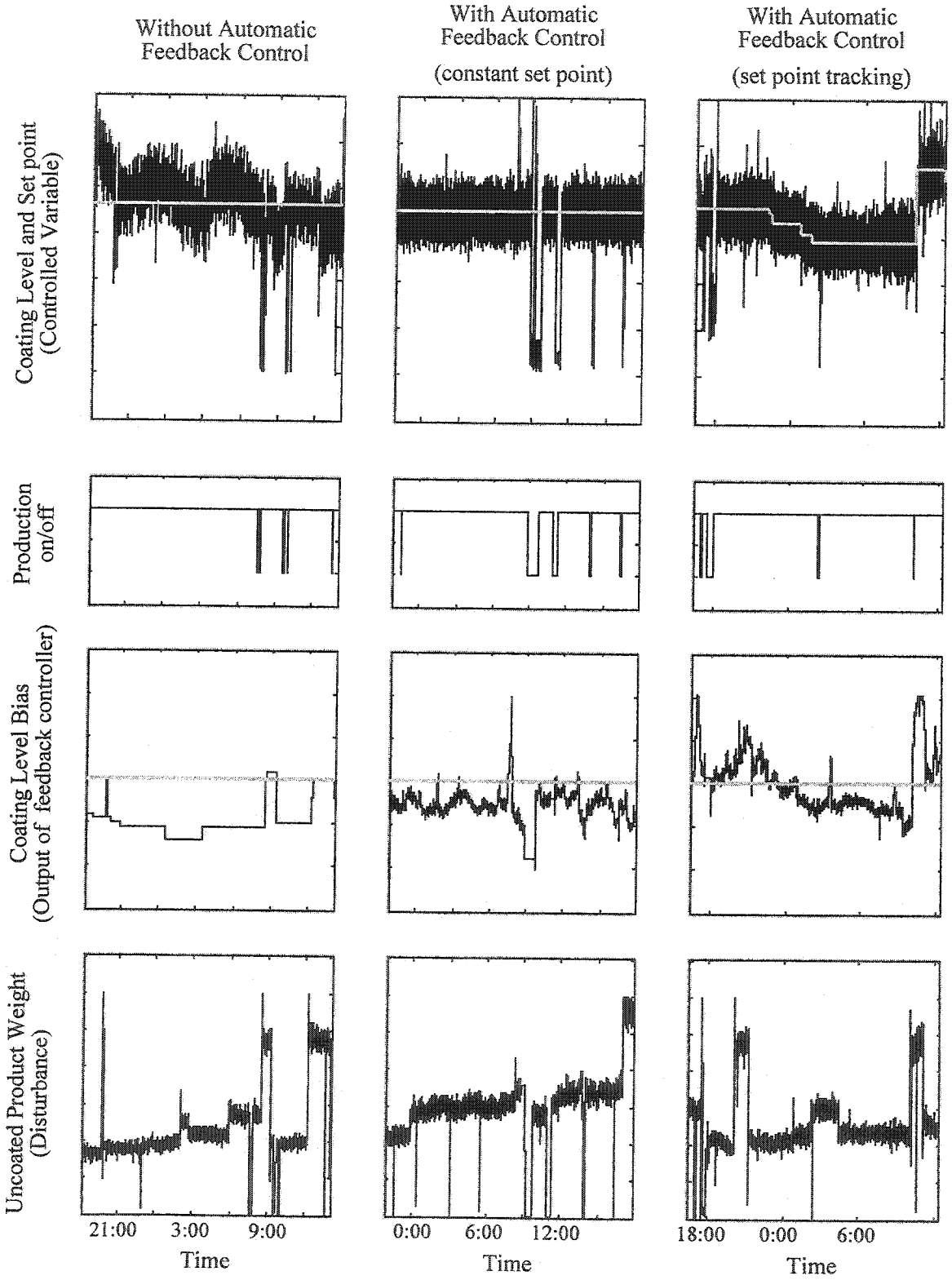



**Figure 4.10 Control performance comparison for three cases**

# Chapter 5

# Snack Food Application (Part III)

## — Miscellaneous Studies: 1) On-line Model Assessment and Correction; 2) Prediction of Organileptic Properties

The purpose of this chapter is twofold: i) to address robustness of the model to variations in the vision system, such as lighting condition changes, including on-line model validity assessment and model correction; and ii) to show some preliminary results for the prediction of organileptic properties (data collected by taste, feel, and sight) of the snack food products.

## 5.1 Robustness Issues

### 5.1.1 Introduction

As shown in the previous chapters on snack food applications, the model built to predict coating concentration is based on color information. Like other color-image based machine vision systems, the model is influenced by variation in RGB color values caused by changes in the vision system components. The RGB errors come from various sources: changes in the intensity and spectral distribution of illumination, changes in the dark current due to thermal fatigue of the sensors, or changes in the spectral transmittance of color filters [Chang and Reid, 1996]. Without compensating for, or robustifying the

model for such changes in the vision system components, the model prediction may contain large errors and therefore mislead the feed back control system or give the operator the wrong judgment about the process performance.

RGB calibration for color image analysis has been addressed in literature. It has been pointed out that it is very difficult to exactly model and classify all error sources and their effects on the RGB color values [Chang and Reid, 1996]. Most of the existing color calibration methods use the image of a color chart, consisting of many color patches with known true color values, to obtain the information of color value changes. Transformation functions are then developed to relate measured color values of the color patches to their true color value.

The choice of transformation functions plays a major role in the color-correction accuracy. Huang [1988] used polynomials as transformation functions. Kang and Anderson [1992] used neural networks to model the transformation functions for color correction. A method proposed by Kanamori et al. [1992] used a grid of 9×9×9 uniformly spaced colors to approximate the transformation functions. In Chang and Reid [1996] and Daul et al. [2000], the transformation functions used are based on information about the physical characteristics of the input and the output. Jackowski et al. [1997] propose the use of elastic transformation functions based on rational Gaussian hypersurfaces.

Two problems are addressed here that need to be overcome in order to achieve a robust model for the snack food application. The first problem is to detect, in real-time, when the conditions of the vision system have changed enough to make the model invalid. The second problem is to correct the model once it has been detected to be invalid. A dataset generated by simulating changes in the lighting source intensity on the training images of product C is used to illustrate the ideas. The coating concentration prediction model used for this study is the one developed in Chapter 3 using method 6.

### 5.1.2 Dataset

A model for the camera output

The output color value of a camera for a pixel $(x,y)$ can be modeled as [Daul et al., 2000; Chang and Reid, 1996]:

$$c_{x,y,i} = \alpha_i \left[ \int_{\lambda_{\min}}^{\lambda_{\max}} L(\lambda,x,y)R(\lambda,x,y)S(\lambda)\eta_i(\lambda)d\lambda \right]^{\gamma_i} + \beta_i \qquad (5\text{-}1)$$

where the subscript $i$ refers to red, green, or blue color channel, $(x,y)$ is the location of the pixel, $\lambda$ is the wavelength, $L(\lambda,x,y)$ is the spectral power distribution of the light source, $R(\lambda,x,y)$ is the reflectance of the object, $\eta_i(\lambda)$ is the transmittance of the $i$-th color filter, $S(\lambda)$ is the spectral sensitivity of the image sensors, $\alpha_i$ is the slope related to amplification and quantization in the camera electronics and frame grabber, $\beta_i$ is the intercept related to the dark current, and $\gamma_i$ is the gamma correction of the camera electronics. This model will be used for generating simulation data.

Dataset generation

The dataset is generated by simulating a change of the lighting source intensity using equation (5-1). The superscript $(tr)$ is used to indicate the lighting condition when collecting the training data set for building the model in Chapter 3, and $(new)$ is used to indicate the on-line lighting conditions of a new image. Assuming that the relative spectral distribution of the lighting source does not change, the spectral power distribution of the new lighting source for a pixel location can be modeled as

$$L(\lambda,x,y)^{(new)} = \xi \cdot L(\lambda,x,y)^{(tr)} \qquad (5\text{-}2)$$

where $\xi$ is the lighting change factor; if $\xi>1$, the lighting condition has become brighter and if $\xi<1$, it has become darker. For the same imaging object and the same pixel

location, the color value under new lighting condition can be derived from equation (5-1) and (5-2).

$$\begin{aligned} c_{x,y,i}^{(new)} &= \alpha_i \left[ \int_{\lambda_{\min}}^{\lambda_{\max}} L(\lambda,x,y)^{(new)} R(\lambda,x,y) S(\lambda) \eta_i(\lambda) d\lambda \right]^{\gamma_i} + \beta_i \\ &= \xi^{\gamma_i} \cdot \alpha_i \left[ \int_{\lambda_{\min}}^{\lambda_{\max}} L(\lambda,x,y)^{(tr)} R(\lambda,x,y) S(\lambda) \eta_i(\lambda) d\lambda \right]^{\gamma_i} + \beta_i \\ &= \xi^{\gamma_i} \left( c_{x,y,i}^{(tr)} - \beta_i \right) + \beta_i \end{aligned} \tag{5-3}$$

Parameters $\beta_i$ and $\gamma_i$ are the characteristics of the vision system and are not available in this case study. We will choose the $\beta$'s to be equal to zero because cameras often have auto black current correction. $\gamma$'s are chosen as 1.1, 0.7 and 0.5 for red, green and blue channels respectively.

Equation (5-3) is used for data generation. 120 images of product C are selected from the training set described in Chapter 3. The relation between the lighting change factor ($\xi$) and the image index number is shown in Figure 5.1. The objective of choosing this relationship is to simulate a linear decay of the lighting source and replacement with a new lighting source (notice there is a sudden value increasing at Image 41). The true function of lighting intensity decay may not be linear, however, the purpose of the dataset generated here is only to illustrate the general ideas concerning the robustness of the model. Three sample images corresponding to different lighting conditions are shown in Figure 5.2.

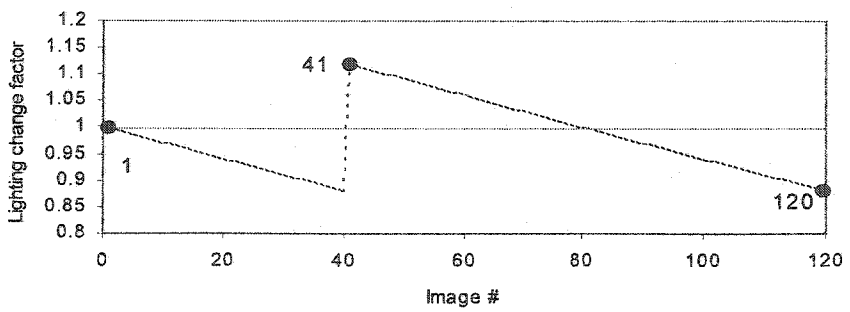
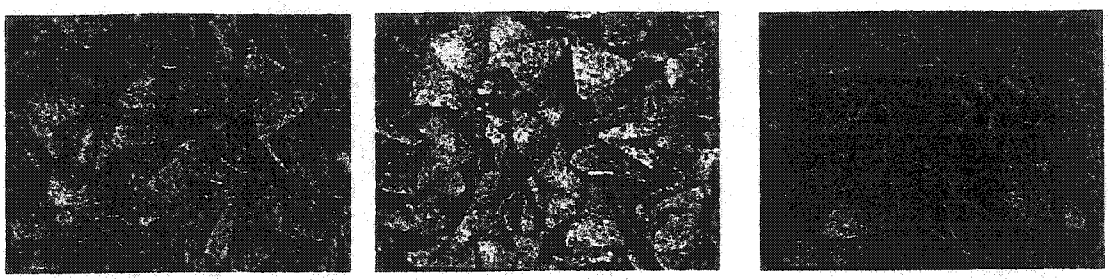



**Figure 5.1 The change of lighting change factor**



1 (Training lighting condition) 41 (Brighter lighting condition) 120 (Darker lighting condition)

**Figure 5.2 Three sample images under different lighting conditions**

### 5.1.3 On-line assessment of model validity

Three methods are proposed in this section to check model validity. The first method is to use the coating concentration prediction error of the model; the second method is to include a color chart in the image scene; and the third method is to use some statistics obtained from the PCA model using the residual images.

Prediction error

Use of the coating concentration prediction error is a simple and direct way to check the validity of a model. Here, the prediction error is defined as

$$e_j = \hat{y}(\mathbf{I}_j) - y_{lab,j} \tag{5-4}$$

where the subscription $j$ refers to the $j$-th image, $e$ is the prediction error, $\mathbf{I}$ is the image matrix, $\hat{y}$ and $y_{lab}$ are the predicted and lab measured coating concentrations.

Figure 5.3 shows the prediction error of the 120 altered images obtained by varying the lighting conditions as discussed above, when compared to the prediction error of the original images (under training lighting condition). The dashed lines indicate the 2 standard deviation prediction error limits estimated from the images under training lighting conditions. We can clearly see that either when the lighting condition becomes too dark (Image 20~ Image 40, Image 100~Image 120) or too bright (Image 41~ Image

60), the predicted values have large errors. Although using the prediction error is easy and effective, its usefulness is limited by the sampling time and laboratory analysis frequency. Since the lab sampling time can not be too frequent, usually not more frequent than once every half an hour, the ability to detect lighting changes in a timely manner is limited. Furthermore, if the frequency of the variation of the vision system component (e.g. fluctuation of lighting caused by temperature variations) is less than lab sampling time, the resulting model invalidity can not be detected by using the prediction error.

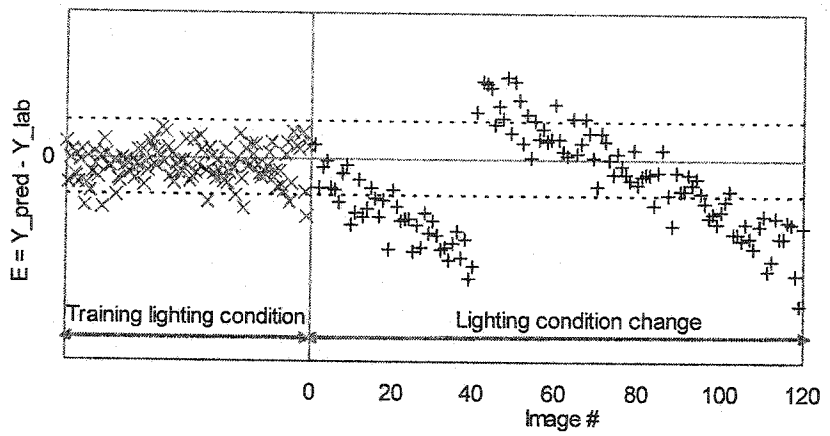



**Figure 5.3 Prediction error for images taken under varying lighting conditions**

Color Chart

If we can somehow fix a color chart beside the conveyor belt and the color chart is included in the imaging scene, any changes in the detected color values of the color chart can be used to indicate variations in the vision system. Furthermore, the information on the detected color changes in the chart can also be used to adjust the model. This is a common color calibration method in machine vision [Chang and Reid, 1996].

A hypothetical example image of using a color chart is shown in Figure 5.4. We can detect vision system component changes by monitoring observed color changes of the color chart. As an example, Figure 5.5 shows the changes of the average color values

of the color chart for the three color channels. These changes clearly indicate the variation of the vision system.

If a color chart can be implemented, it is a very easy and effective way to detect the changes in the vision system. However, using a color chart to calibrate the vision system is commonly used in an off-line mode and is not easily implemented in an on-line mode. In order to use such a color chart calibration approach in the snack food processes, the material of the color chart has to be able to withstand the industrial environment, the colors in the color chart must remain constant, and the surface of the color chart must be kept clean from fouling by the coating powders.

**Figure 5.4 A hypothetical example to include a color chart in the image scene**

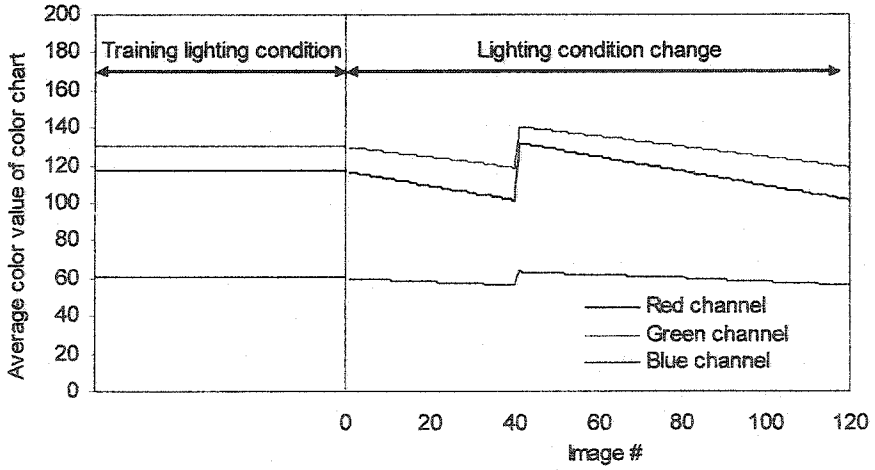


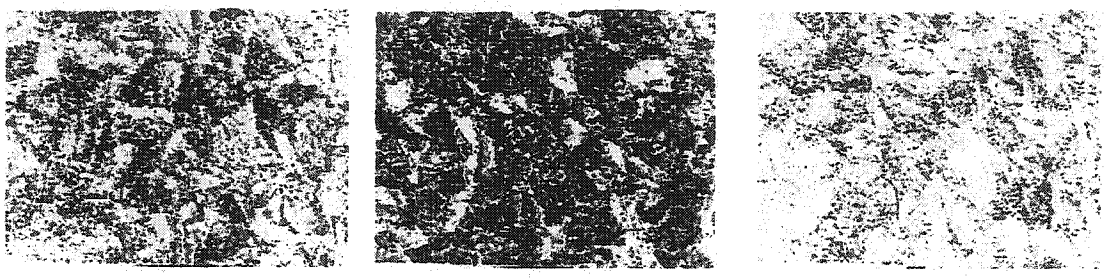

**Figure 5.5 The average color value changes of color chart for the images taken under varying lighting conditions**

Cumulative Histogram of the $T_3$ Residual image

Here an indicator is proposed to test the adequacy of the model for cases when a color chart is not available. This indicator is developed based on the statistical characteristics of the $T_3$ score image (refolded and scaled $t_3$ values, see Chapter 2). This image is simply the residual image after the first two components have been used for the model because a color image is a three-variable multivariate image.

As explained in Chapter 3, model 6 for predicting coating concentration is a distribution model developed in the $t_1$-$t_2$ PCA plane. The information in the $t_3$ direction is not used because, in the training data, it contains little information related to the coating concentration. When the condition of the vision system changes, the color values of the pixels will change, and then the projections $t_1$, $t_2$ and $t_3$ of each pixel in the image will also change. However, because the variations in $t_1$ and $t_2$ also can be caused by the variation of coating concentration, it is hard to distinguish between these two variation sources (a vision system condition change and a coating concentration change).

Therefore, we can use any changes in the $t_3$ dimension to monitor the variation of the vision system. This idea is similar to the use of the distance to model (DModx) in normal PCA analysis (in this three variable case, the magnitude of $t_3$ at each pixel location is its model distance). For visual assessment, the $T_3$ score image is a good way to see the nature of the variation of the residuals throughout the image space. Figure 5.6 shows the three $T_3$ score images, corresponding to the images shown in Figure 5.2. These residual images show the differences when the lighting conditions become brighter or darker.



1 (Training lighting condition) 41 (Brighter lighting condition) 120 (Darker lighting condition)

**Figure 5.6 $T_3$ score images for the three sample images shown in Figure 5.2**

However, direct visual inspection of the $T_3$ score image only provides a qualitative assessment. More quantitative measures need to be extracted for automatic monitoring. One way is to use the cumulative histogram from the $T_3$ score image. In Figure 5.7, the cumulative histogram (256 levels) of the 120 images under the training lighting condition and the two images under brighter or darker lighting conditions (Image 41 and Image 120) are plotted. It is evident that as the lighting condition changes, the cumulative histogram of the $T_3$ image also changes.

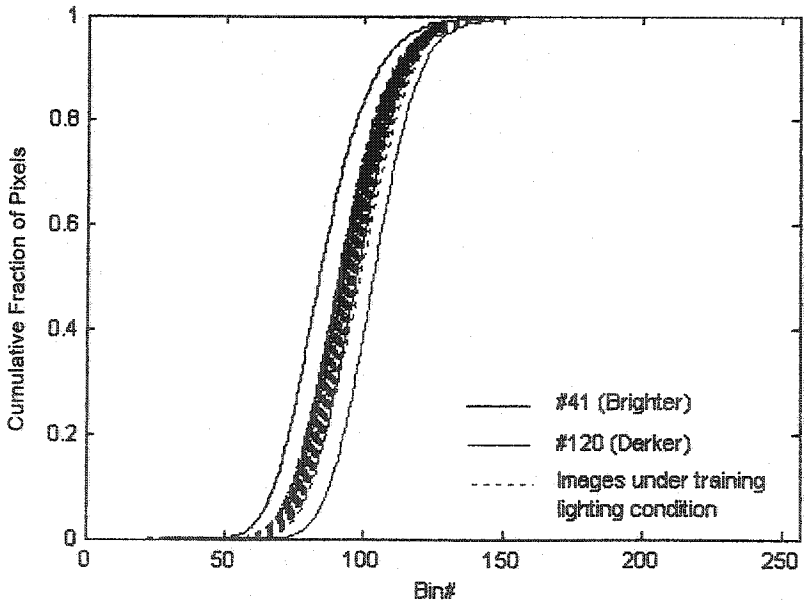



**Figure 5.7 Cumulative histograms of the $T_3$ score images**

One approach to quantify these changes is to apply PCA on the cumulative histograms and use Hotelling's $T^2$ to monitor the validity of the model. The training set includes the cumulative histograms of all the images under training lighting conditions. Three principal components were found to be adequate.

Hotelling's $T^2$, denoted as $H$, for observation $j$ (the cumulative histogram of the $T_3$ image for image $j$), based on $A$ components is defined as:

$$H_j = \sum_{a=1}^{A} \frac{\tau_{ja}^2}{s_{\tau_a}^2}$$

where $\tau_{ja}$ is the score value of the $a$-th component for observation $j$ and $s_{\tau_a}^2$ is the variance of $\tau_{ja}$, estimated from the training set.

Assuming that the score variables of the training data are normally distributed (based on the central limit theorem), $H_j \cdot \frac{N(N-A)}{A(N^2-1)}$ is $F$ distributed with $A$ and $N$-$A$ degrees of freedom, in which $N$ is the number of the observations in the training set and $A$ is the number of principal components. Therefore if $H_j > \frac{A(N^2-1)}{N(N-A)} \cdot F_{A,N-A}(p=0.05)$, then the observation $j$ is outside the 95% confidence region of the PCA model.

Figure 5.8 shows the Hotelling's $T^2$ value for the images under training lighting condition and for the images under varied lighting conditions. The dashed line indicates the 95% confidence region. This statistic clearly indicates that something is changing in the process. Compared to Figure 5.3, we can see that when the Hotelling's $T^2$ value is outside the 95% confidence region (e.g. from Image 20 to Image 60 and from Image 100 to Image 120), in most of the cases, the model prediction errors also fall outside the 2 standard deviation region.

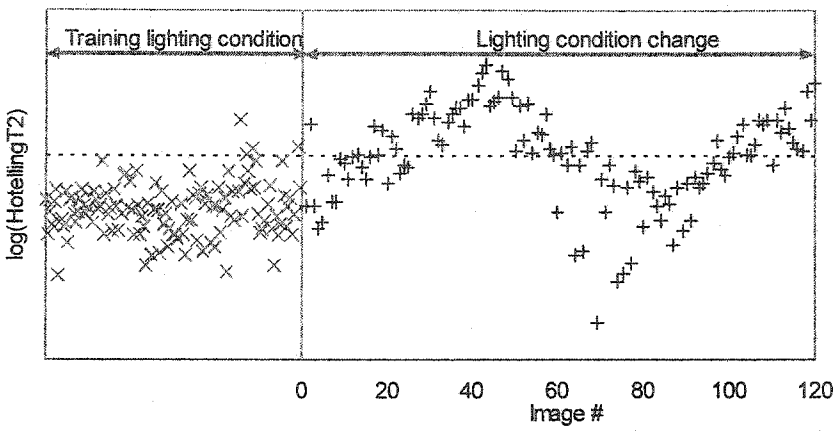



**Figure 5.8 Hotelling $T^2$ for images taken under varying lighting conditions**

This method can be applied at the same sampling rate as the imaging system, and extra equipment is not needed. Moreover, the computation is very simple to perform.

### 5.1.4 Model correction

Two methods for on-line correction of the models are presented in this section. The first method uses the information from a color chart to correct the colors appearing in the images, and the second method uses the information from the $T_3$ score image to estimate the model prediction error and then correct the model output.

Using information from the color chart

Denote $\mathbf{c}^{(tr)}$ as the vector of color values for the patches in the color chart under the training vision system conditions, and $\mathbf{c}^{(new)}$ as the vector of color values under the new on-line conditions. If we can use a function to describe the relationship between these two sets of color values:

$$\mathbf{c}^{(tr)} = f\left(\mathbf{c}^{(new)}\right) \tag{5-5}$$

then equation (5-5) can be used in an on-line manner to correct the colors of the new images.

A simple linear model is recommended for on-line correction because the deviations in color values may be approximated by a linear model when changes in the vision system are small or moderate. Also if $f$ is chosen to have a complicated nonlinear structure, optimization is often needed to obtain the parameters of the function and correction may take a long period of time. Consider the dataset generated in section 5.1.2. Since the equation for generating the dataset (equation (5-3)) had no error term and was of a linear form, then the use of a linear transformation model to correct the new images under varying lighting conditions will give corrected model predictions identical to the prediction of the original images (under training lighting conditions).

Using cumulative histogram of $T_3$ image

A method was presented in section 5.1.3 which uses the cumulative histogram of the $T_3$ image to monitor changes in the vision system's condition. In the following text, we show that this information can be correlated with the model prediction error and used to update the model prediction (obtained using model 6 developed in Chapter 3).

A PLS model is built by regressing the cumulative histograms of the $T_3$ image against the model prediction error. Of the 120 images taken under varying lighting conditions (generated by equation (5-3)), 63 images are used as a training set and the other 57 images as a test set. For the $j$-th image, the model is

$$\hat{e}_j = \mathbf{h}_j \cdot \hat{\mathbf{b}},\ j=1,2,... \tag{5-6}$$

where $e_j$ is the model prediction error defined in equation (5-4), $\mathbf{h}_j$ is the cumulative histogram of the $T_3$ score image, and $\hat{\mathbf{b}}$ is the coefficient vector of the PLS model.

For a new on-line image $\mathbf{I}_j$, the cumulative histogram vector of its $T_3$ image $\mathbf{h}_j$ is calculated and the prediction error $\hat{e}_j$ can be computed from equation (5-6). Then this estimated error value can be used to correct the prediction computed by the model built in Chapter 3. The corrected coating concentration will be

$$y_{cor,j} = \hat{y}(\mathbf{I}_j) - \hat{e}_j$$

Three latent variables are used by cross validation. The prediction error of the corrected prediction for the images taken under varied lighting conditions is shown in Figure 5.9. We can see that after correction most of the points are within the 2 standard deviation region defined by the prediction errors of the images collected under the training lighting conditions (dashed lines). By using this method, the model can be automatically corrected for variations in the lighting system. Moreover, the PLS model developed here can be continually updated every time new lab measurement data become available.

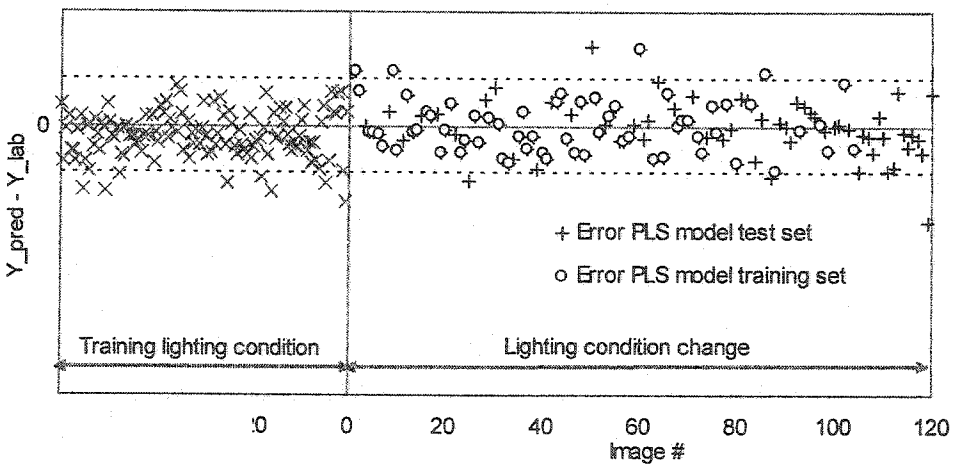



**Figure 5.9 Prediction error for images taken under varying lighting conditions after correction**

### 5.1.5 Conclusions

In this section, the problems of detecting and correcting for variations in the lighting and camera system are addressed. These problems need to be addressed to achieve robust model predictions under variations of the vision system. Three on-line methods are proposed to assess the validity of the model. Prediction error is a straightforward indicator of model invalidity, but limited by the long period between laboratory analyses. By attaching a color chart, changes in the vision system can be detected in a direct manner. However, the on-line implementation using color chart may be not easy. Therefore, we present a novel approach that uses information from the residual $T_3$ score image. In this method, PCA is performed on the cumulative histograms of the $T_3$ score images for a set of images under training lighting conditions. Hotelling's $T^2$ is then used to judge whether the color characteristics of a new image is similar to the images under the training lighting conditions. From the example, we can see that the proposed method can effectively detect when the model becomes invalid.

Two methods are also presented for on-line model correction: the first uses a color chart while the second uses the cumulative histogram of the $T_3$ score image. The first method corrects the model by adjusting the measured color through transformation functions which are identified by the change of the color in a color chart. The second method corrects the model by estimating the prediction error using a cumulative histogram of the $T_3$ score image through a PLS model. The simulation results show that the method can correct the model quite well. This method, combined with the method for on-line model assessment, provides a powerful tool to detect imaging condition changes in real-time and to correct its model prediction in an on-line manner when it needs to be corrected. Therefore a robust prediction of coating concentration can be obtained for monitoring or control.

## 5.2 Prediction of Organileptic Data

In the previous work (Chapter 3, Chapter 4 and section 5.1), we have discussed how to build a soft sensor to predict the coating concentration during snack food production. This section presents a preliminary study for the prediction of organileptic properties of snack food.

### 5.2.1 Organileptic data

Consumers generally make decisions on the quality of a snack through the senses of taste, texture, and visual presentation. Control of the coating concentration and coating coverage distribution, discussed in Chapter 3, 4 and section 5.1, is only part of the quality control problem in snack food production.

Basically, two types of analysis methods are used to measure the so-called organileptic properties, which are properties based on texture, taste and sight. One method is to use the human sensory response and the other one is to use a mechanical device. However, neither of these two methods can be used for on-line feedback control in high speed production. Therefore, this section presents some preliminary approaches at developing an image based soft sensor system for some of these properties.

In this study, four types of organileptic properties are considered: blister level, toast points, taste and peak break force. The first three properties are measured by human sensory response (sight and taste) and the last one is measured by a mechanical equipment.

Blister level is a measure of the degree of the blistering on the surface of the snack food (see Figure 5.10). Chips are graded from 0 to 6 by comparing the chips with reference drawings. Level 0 represents no blisters found on the chips, and level 6 represents chips that contain very large number of blisters. Level 3 is considered to be the optimal amount of blisters.

Toast points are small black 'grill' marks left by the oven belt on the surface of the chips (see Figure 5.10). The toast point level is graded from 0 to 4 by comparing chips with reference drawings. Level 0 represents no toast points on the chip and level 4 indicates a burnt chip. Level 2 is considered to be optimal.

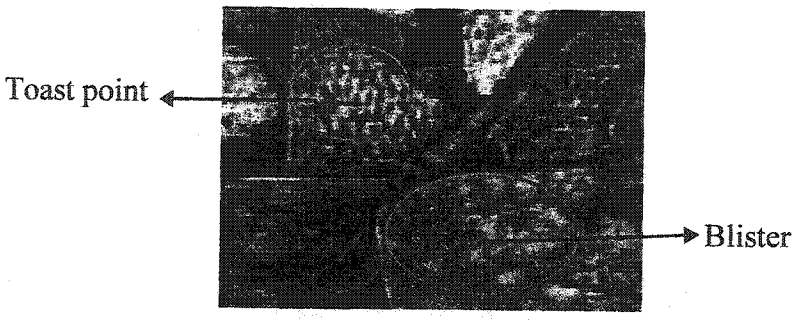



**Figure 5.10 Illustration of blister and toast points**

A taste property (unrelated to coating level) is also measured by comparing it with reference chips. The judges eat a reference chip and then eat the sampled chip. The taste of each chip is graded from 0 to 5. Level 0 indicates a dense chip and level 5 is a light, blistery chip. The aimed level of this taste property is level 3.

Peak break force is a mechanical measure of the firmness of the chips. The value of peak break force is related to the break strength and hence the force required to bite the chip.

Thirteen cells of data, shown in Table 5.1, are used for this study. Each cell corresponds to a certain operation condition. In each cell, 15 RGB color images are collected. These will be used as predictors in the model (4 sample images can be found in Table 5.2). 30 chips, randomly selected from each cell, are used to measure the blister level and the toast point level. Another, randomly selected, 10 and 15 chips from each cell are used to measure the taste property and the peal force, respectively. Average values of blister level, toaster points, taste and peal force for each cell are used as the response matrix Y (see Table 5.1). From Table 5.1, we can see that the Y variables are

highly correlated. The chips having high blister and toast points level, such as cell 6A generally have also high taste level and a larger peak break force.

**Table 5.1 Average organileptic properties**

| Cell Name | Coating Level (1-Coated; 0-Uncoated) | Y | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Blister Level | | Toast points | | Taste | | Peal Force (g) | |
| | | Mean | Std | Mean | Std | Mean | Std | Mean | Std |
| 1A | 0 | 0.87 | 0.57 | 0.97 | 0.52 | 1.5 | 0.71 | - | - |
| 1B | 0 | 1.3 | 0.53 | 0.98 | 0.69 | 2.5 | 0.97 | - | - |
| 2A | 1 | 1.07 | 0.65 | 0.45 | 0.51 | 2.1 | 0.57 | - | - |
| 2B | 1 | 1.27 | 0.43 | 1.42 | 0.63 | 2.4 | 0.52 | 480.9 | 112.6 |
| 2C | 0 | - | - | - | - | - | - | 451.3 | 103.2 |
| 3A | 0 | 1.63 | 0.85 | 1.38 | 0.64 | 2.1 | 0.74 | 530.9 | 105.8 |
| 3B | 0 | 1.52 | 0.52 | 1.65 | 0.44 | 1.9 | 0.74 | 606.7 | 142.7 |
| 4A | 1 | 1.62 | 0.54 | 1.68 | 0.70 | 2.4 | 0.46 | 542.7 | 121.8 |
| 4B | 1 | 2.15 | 0.87 | 1.72 | 0.67 | 2.55 | 0.69 | 655.4 | 176.1 |
| 5A | 0 | 4.73 | 1.44 | 1.8 | 0.34 | 3.1 | 1.10 | 786.1 | 150.6 |
| 5B | 0 | 3.64 | 1.40 | 1.8 | 0.85 | 2.7 | 1.27 | 709.3 | 98.5 |
| 6A | 1 | 4.22 | 1.58 | 2.5 | 0.63 | 3.3 | 1.16 | 720.8 | 166.6 |
| 6B | 1 | 4.43 | 1.69 | 2.17 | 0.82 | 3.7 | 1.06 | 668.2 | 158.7 |

### 5.2.2 Image analysis

Cancellation of the coating effect

Unlike the objective in Chapter 3, here we are not interested in the coating level of the snack food. Instead, the texture information of the image other than spectral information is considered. However, the color variation caused by coating level may influence the texture information. Therefore, a procedure based on PCA is proposed to cancel the effect of the coating level.

A PCA (mean-centered) is performed on several uncoated product images (background pixels have been removed). The first two components ($t_1$ and $t_2$) explained 87.0% and 11.3% of the variation respectively. Therefore, this $t_1$-$t_2$ plane captures most of the variation caused by changes in the imaging condition and is not influenced by the variation caused by the coating level. By projecting the pixels of the coated images onto this $t_1$-$t_2$ plane, the coating information should be largely cancelled. Table 5.3 shows the

reconstructed images (from the first two principal components) for the 4 sample images shown in Table 5.2.

**Table 5.2 Four sample images in the organileptic dataset**

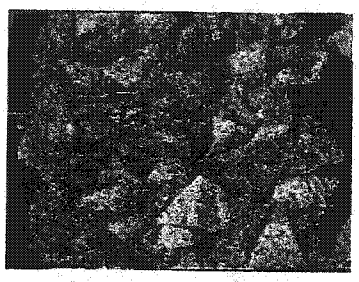
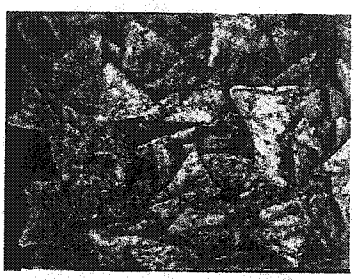
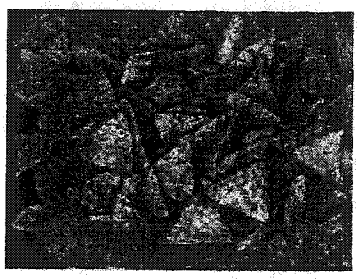
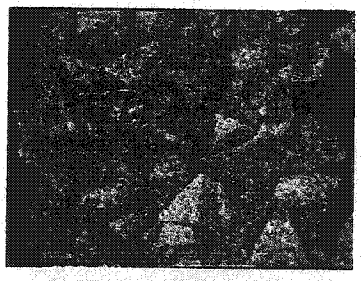
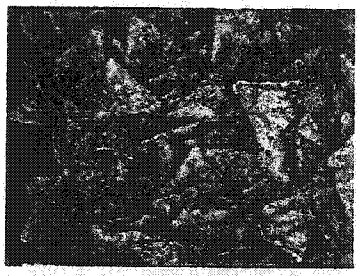
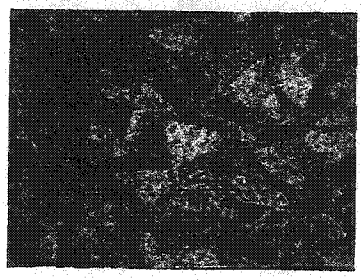
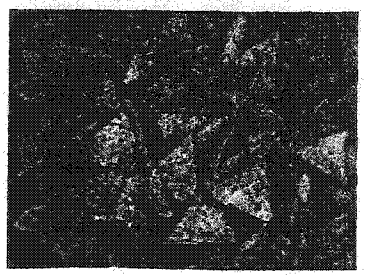
| | Low values of the Y's | High values of the Y's |
|---|---|---|
| Uncoated Product |  |  |
| Coated Product |  |  |

**Table 5.3 Four Images reconstructed from the first two principal components (coating information has been removed)**

| | Low values of the Y's | High values of the Y's |
|---|---|---|
| Uncoated Product |  |  |
| Coated Product |  |  |

Texture information extraction

A series of 9 Laplacian of Gaussian (LoG) spatial filters with different window sizes are applied to the first two score images to extract texture features. Laplacian filters are derivative filters used to find areas of rapid change (edges) in images. Since derivative filters are very sensitive to noise, it is common to smooth the image (e.g., using a Gaussian filter) before applying the Laplacian. This two-step process is called the Laplacian of Gaussian (LoG) operation. The 2D LoG function centered on zero with Gaussian standard deviation $\sigma$ has the form:

$$\mathrm{LoG}(x,y) = -\frac{1}{\pi\sigma^4}\left[1-\frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The Gaussian standard deviation is chosen as 0.5. The window sizes of the filters are chosen from 4 to 20 (even numbers). These parameters were chosen by trial-and-error. For each filtered image, a simple average value is computed. Therefore, totally 18 texture feature variables for each RGB image are generated, including the 9 average filtered values for the first score image and the 9 average filtered values for the second score image.

Prediction of organileptic data

Four PLS models are built to predict the organileptic data (one for every Y property). An average texture feature vector is computed for each cell. This averaged texture feature vector along with a dummy variable, which indicates whether the products in the image are coated (this could be replaced with the predicted coating concentration in future studies), are used as predictors.

Among the 13 observations (corresponding to the 13 cells), 10 observations are used for training the models. Figure 5.11 (a-d) shows the prediction vs. observation plots for each one of the Y's.

In Table 5.4, the root mean square errors for prediction (RMSEP) of the test set is listed for the four Y's, as well as the estimated standard deviation of measurement error for one piece of chip. This measurement error is estimated (see Appendix B) by assuming that the global texture property for each cell is constant and all the samples in each cell are treated as replicate points. Relative errors for blister level are used because the standard deviation value for each cell increases as the average values of the blister level increases (see Table 5.1). The number of latent variables used is also shown in Table 5.4.

We can see from Table 5.4 that the measurement error is larger than model prediction error for all 4 Y's if only one piece of chip is measured. The measurement error can be easily reduced by taking more samples within short time and using the average measurements of these samples. However, since organileptic properties of a piece of chip are measured manually (people have to grade the product by looking at it, eating it or taking it to the mechanical machine), the number of samples used for measuring the properties is proportional to the time and manpower required. Therefore, to set up a monitoring system for organileptic properties at a moderate sampling time requires a large manpower. On the other hand, by using an imaging system, once a good model has been trained, the predictions can be obtained very fast with almost no extra cost, and the sampling rate can be increased as desired.

### 5.2.3 Conclusions

In this section, a methodology has been developed for using color images to predict organileptic properties (properties related to taste, texture, etc.) of a snack food product. This is the first known attempt to predict such properties using images. By projecting the pixels from images of the coated products onto the two dimensional space obtained by performing a PCA on uncoated product images, the variation caused by coating concentration is largely separated from the texture information. A series of 9 Laplacian of Gaussian spatial filters are applied on the first two score images of the coated images and the average value for each filtered image is used as texture feature variable. Totally 18 texture feature variables and 1 dummy variable to indicate the

coating level are regressed with four Y property variables using four PLS models. The model prediction results appear to be reasonably good for all four properties. It is expected that the predictions can be improved when more samples are collected and more accurate measurements are obtained.

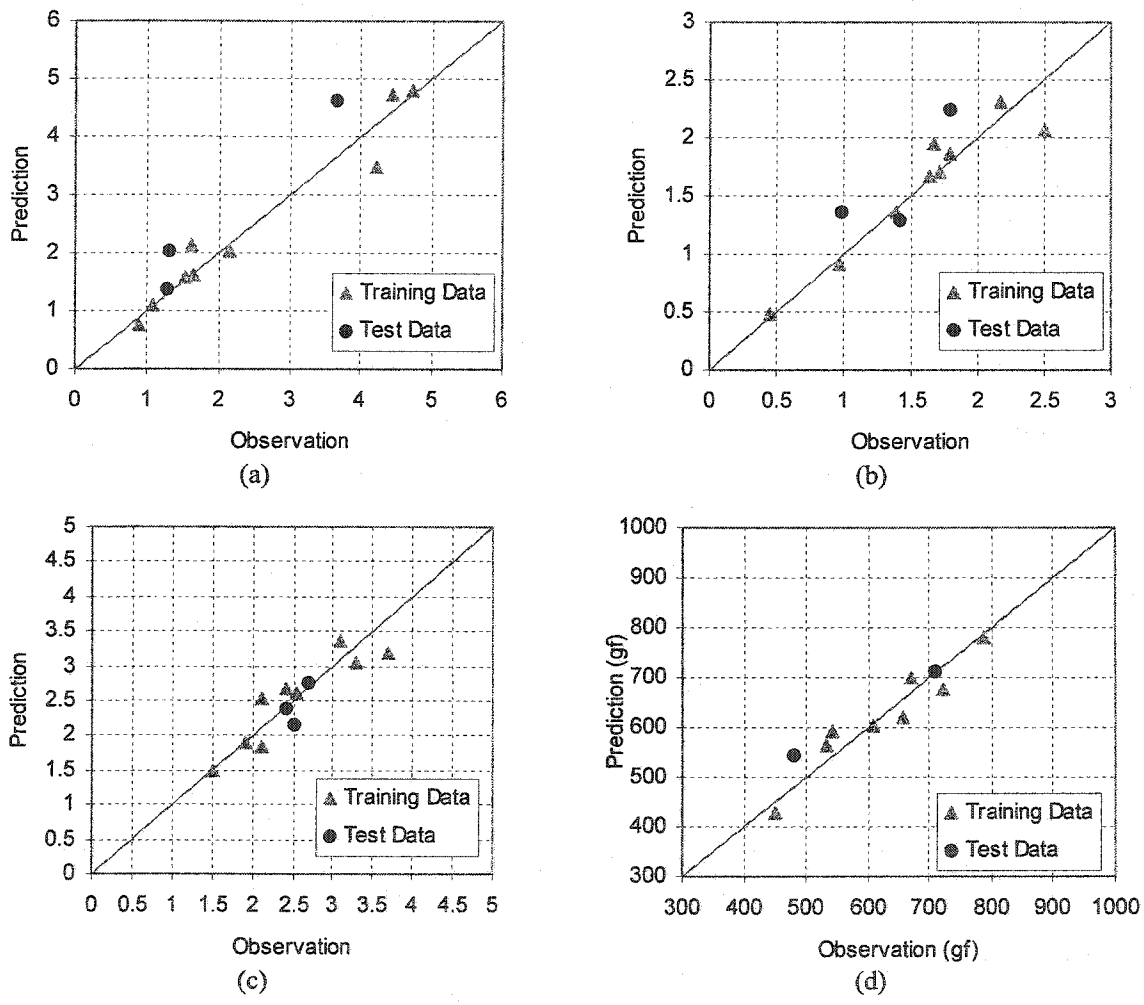



**Figure 5.11 Prediction vs. Observation plots for the four Y organileptic properties. (a) Blister level, (b) Toast point level, (c) Taste property, (d) Peak break force**

**Table 5.4 Comparison of prediction errors of the inferential model and analysis measurement error**

| | Number of LV's | RMSEP for Test set | Std of Measurement Error for taking 1 piece of chip |
|---|---|---|---|
| Blister Level* | 5 | 0.36 | 0.44 |
| Toast Point Level | 5 | 0.34 | 0.64 |
| Taste Property | 3 | 0.29 | 0.87 |
| Peak Break Force | 4 | 45.35 | 138.91 |

* For blister level, relative RMSEP and relative measurement error are computed

# Chapter 6

# Flame Application (Part I) — Methodologies

In this chapter and Chapter 7, a flame monitoring system for an industrial boiler is developed. Methodologies are developed in this chapter illustrated by two case studies. It shows that features extracted from flame images contain useful information and can be used to predict process properties. In Chapter 7, models for predicting heat of combustion of liquid fuel (waste streams from other processes) and emissive concentrations of $NO_x$ and $SO_2$ are presented.

## 6.1 Introduction

Combustion plays an important role in many industrial processes. The efficiency of the combustion usually has a great influence on the economics of the process and on its environmental impact.

In combustion processes fuel and oxidizer (typically air) are mixed and burned. Generally, two categories can be identified based on whether the fuel and oxidizer are mixed first and then burned (premixed) or whether combustion and mixing occurs simultaneously (nonpremixed). Each of these categories is further subdivided based on whether the fluid flow is laminar or turbulent. Among the four classes, the turbulent nonpremixed combustion processes are of interest in many industrial applications. They

appear in jet engines, diesel engines, steam boilers, furnaces etc. The use of the nonpremixed combustion is widespread because it is safer to handle than premixed combustions. However, nonpremixed combustion involves more complex chemistry and is harder to model or control. Unless very sophisticated mixing techniques are used, nonpremixed flames show a yellow luminescence, caused by glowing soot particles. The colors of the flames indicate the combustion region and the temperature of the field. This latter feature allows for the possibility of using color flame images to monitor the combustion process.

Visualization methods have been used to study combustion flames in laboratories. Shimoda et al. [1990] reported a combustion diagnostic system where the radiation energy and temperature profiles of flames were quantitatively identified from the flame images and the concentrations of un-burnt carbon and $NO_x$ in the exhaust gas were estimated in a coal-fired boiler. Yamaguchi et al. [1997] developed a fiber-optic imaging sensor to detect the air-ratio in a premixed-type gas-fired model combustor by monitoring the radiant intensities of flames over three spectral bands. Huang et al. [1999] set up a flame flicker monitoring system, where the flicker of a gaseous flame was quantified by computing the oscillation of the radiant intensity of individual pixels within the luminous region of flame images. Lu et al. [2000] designed and evaluated an instrument system for monitoring, characterization and evaluation of fossil-fuel-fired flames in a utility boiler. Geometrical and luminous parameters of the flame were determined from the images. Wang et al. [2002] reported a method to predict $NO_x$ emissive concentration for a coal boiler by using color flame images and neural networks. However, few of the above used turbulent nonpremixed flames as the object of their study.

Television systems have been installed to monitor the flame in many industrial furnace and boiler systems. However most of the time the only information the flame images are providing is whether the flame is burning. Due to the highly turbulent combustion, the flames are always bouncing around, and even the most experienced operator will have difficulty in judging the combustion performance.

In this situation, a monitoring system based on image analysis can be very helpful. This is the motivation of the research carried out in this chapter. An industrial steam boiler is used as the object of the study. To analyze the color images, nine features are extracted from the PCA score plot space for each image. Further analysis of the image features and the process measurements are performed using PCA and PLS to help understand the relationship between the feature variables and the process variables.

## 6.2 System Setup

A schematic diagram of the industrial boiler and flame monitoring system is shown in Figure 6.1. The steam boiler studied in this chapter uses both the waste liquid streams from other processes and natural gas as fuels. Consequently, the overall composition of the fuel often changes dramatically. An analog color camera was installed in the boiler and connected with a monitor for displaying the live images. There are four burners inside the boiler. The flames shown in the images captured by the camera are the results of all four burners. In this research, the analog signals were recorded by a normal VCR and then a video card is used to convert the signals on the video tapes into the digital images. In the future a frame grabber can be used for on-line analog-digital conversation. The resulting images are RGB color images, with the size of 120×160 pixels. Considering the processing time, the imaging sample time is set as 1 frame per second.

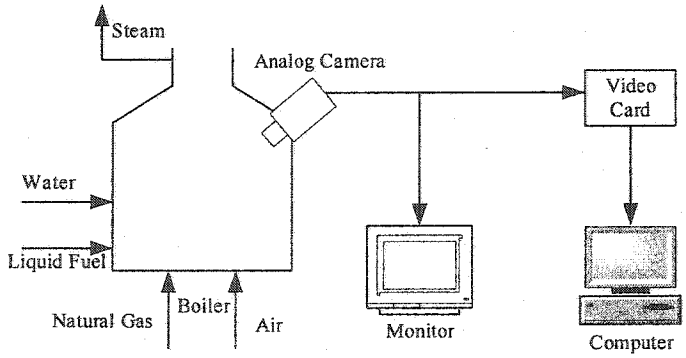



**Figure 6.1 Scheme of the flame monitoring system**

## 6.3 Case Studies

### 6.3.1 Case descriptions

Two cases are studied in this chapter.

Case I covers a 114 minute period (collected in November 2000, see Figure 6.2). In this period of time, only liquid fuel was used. In the first half of this period of time, the liquid fuel flow rate was decreased from 1.5 to 0.75 kg/s; then in the second half of the period, it was increased back to 1.5 kg/s. The steam generated followed the same trend as the fuel flow.

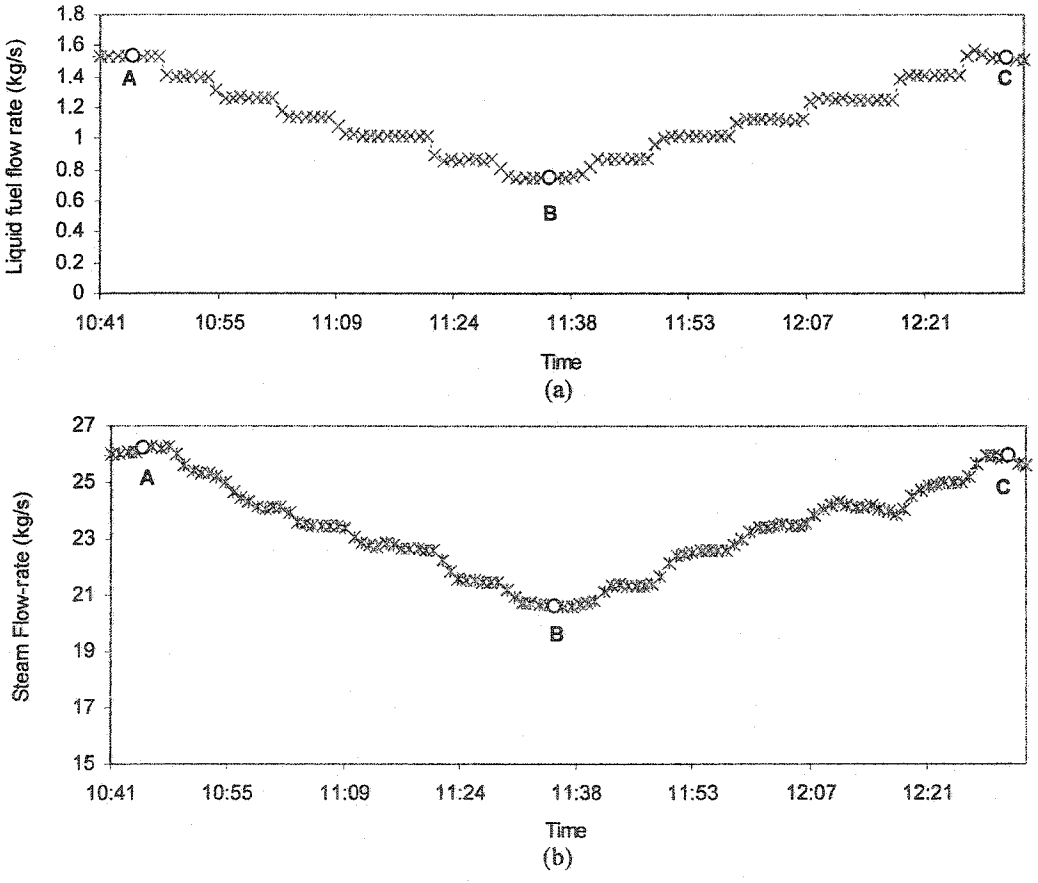



**Figure 6.2 Flow rates of fuel and steam for case I. (a) Flow rate of liquid fuel, (b) Flow rate of steam generated**

Case II covers a 56 minute period (collected in June 2001). In this case, both the liquid fuel and natural gas were used. During this period of time the flow rate of liquid fuel was gradually reduced from 0.889 kg/s to 0 and at the same time the flow rate of natural gas was increased from 1.60 to 1.99 scm/s to keep the steam generated at the desired level. The flow rate changes of the fuels are shown in Figure 6.3a and the resulting change of the steam flow rate is shown in Figure 6.3b.

In both cases, the air/fuel ratio of the boiler is automatically modulated based on the pre-set control scheme.

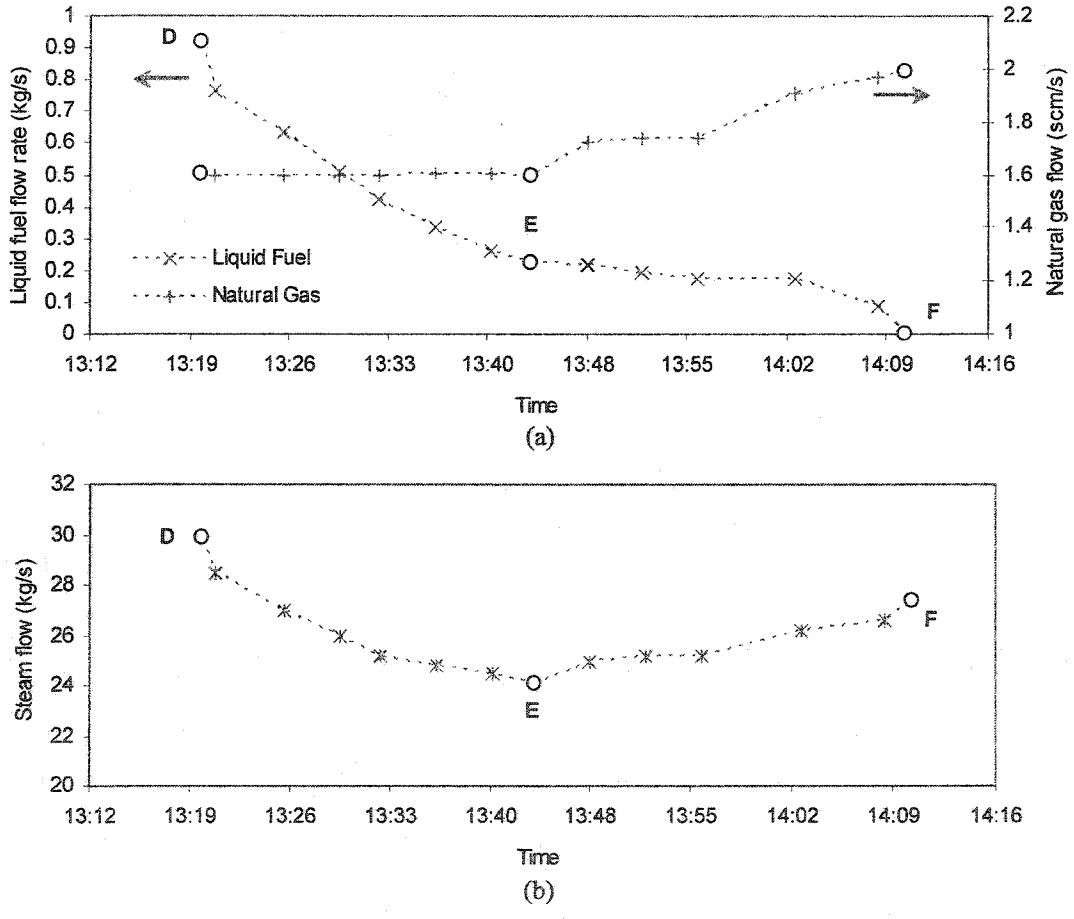
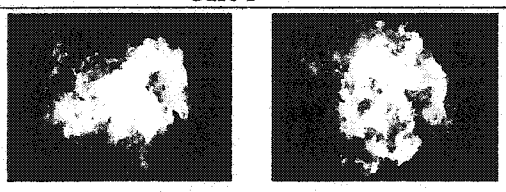
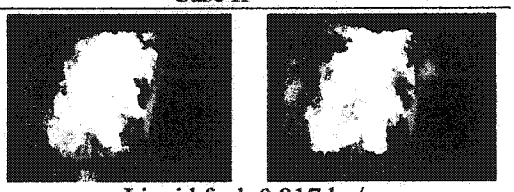
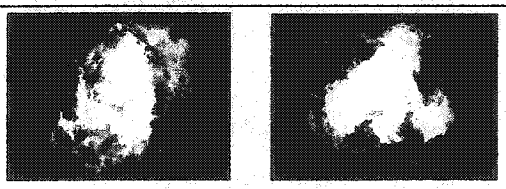
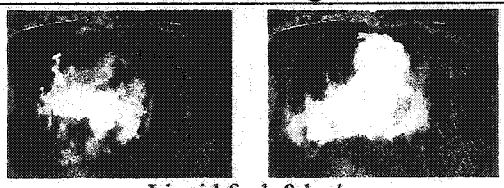



**Figure 6.3 Flow rates of fuels and steam for case II (a) Flow rate of liquid fuel and natural gas (b) Flow rate of steam generated**

### 6.3.2 Flame images

A total of 6840 frames for case I and 3360 frames for case II were obtained. Table 6.1 shows some sample images corresponding to the points (A-F) marked in Figure 6.2 and Figure 6.3. For each point, two consecutive images with 1 second time difference are shown. It is reasonable to assume that during this one second, the feed and composition conditions in the combustion process did not change. It can be observed that the flames in the boiler appear highly turbulent, with the images changing significantly over every 1 second interval. This poses considerable difficulty in trying to extract stable information from a rapidly varying combustion process.

**Table 6.1 Sample images taken one second apart at different conditions**

| Case I | | Case II | |
|---|---|---|---|
| A |  Liquid fuel: 1.5 kg/s<br>Steam: 26.1 kg/s | D |  Liquid fuel: 0.917 kg/s<br>Natural Gas: 1.61 scm/s<br>Steam: 30.6 kg/s |
| B |  Liquid fuel: 0.75 kg/s<br>Steam: 20.6 kg/s | E |  Liquid fuel: 0.225 kg/s<br>Natural Gas: 1.58 scm/s<br>Steam: 24.1 kg/s |
| C |  Liquid fuel: 1.5 kg/s<br>Steam: 26.1 kg/s | F |  Liquid fuel: 0 kg/s<br>Natural Gas: 2 scm/s<br>Steam: 27.2 kg/s |

## 6.4 PCA Score Space

A PCA is performed and the features are then extracted from the reduced score space. 100 flame images are used in this study to compute the loading matrix and score scaling range (see computation details in section 2.5). The first two components explained 99% of the total variance. Figure 6.4 shows three score images for a sample image. A reconstructed image from the first two components is also shown. The $T_3$ score image represents the residuals and contains little information. The reconstructed image from the first two components is almost the same as the original image.

Table 6.2 shows the corresponding $t_1$-$t_2$ score plots (computation of score plots can be found section 2.5) for the sample images shown in Table 6.1. We can see from Table 6.2 that images captured at the same combustion condition have very similar score plot histograms of pixel intensities. However, as the combustion conditions change (i.e. down the columns in Table 6.1), the locations of pixels in the score plots change noticeably. This is the key result that enables one to use flame images to analyze and monitor the process for changing conditions. For any given process condition, even though the flame images are bouncing around, the PCA score space of that image is very stable, and it only changes shape and location with changing process conditions.

Each location in the $t_1$-$t_2$ score plot represents a certain kind of color (ignoring the residual variation in $t_3$). We can compute the color for each location by

$$\begin{gathered}[R\ G\ B]_{ij} = t_1(i)\cdot \mathbf{p}_1{}^T + t_2(j)\cdot \mathbf{p}_2{}^T,\ i,j=0,\ldots 255 \\ t_1(i) = i\cdot(t_{1,\max} - t_{1,\min}) + t_{1,\min},\ t_2(j) = j\cdot(t_{2,\max} - t_{2,\min}) + t_{2,\min}\end{gathered} \tag{6-1}$$

A color plane then can be computed for the score plot space, as shown in Figure 6.5. We can see the relation between the position in the score plot and the color. Since the colors of the flames are different under different combustion conditions we observe movement in the score plot.

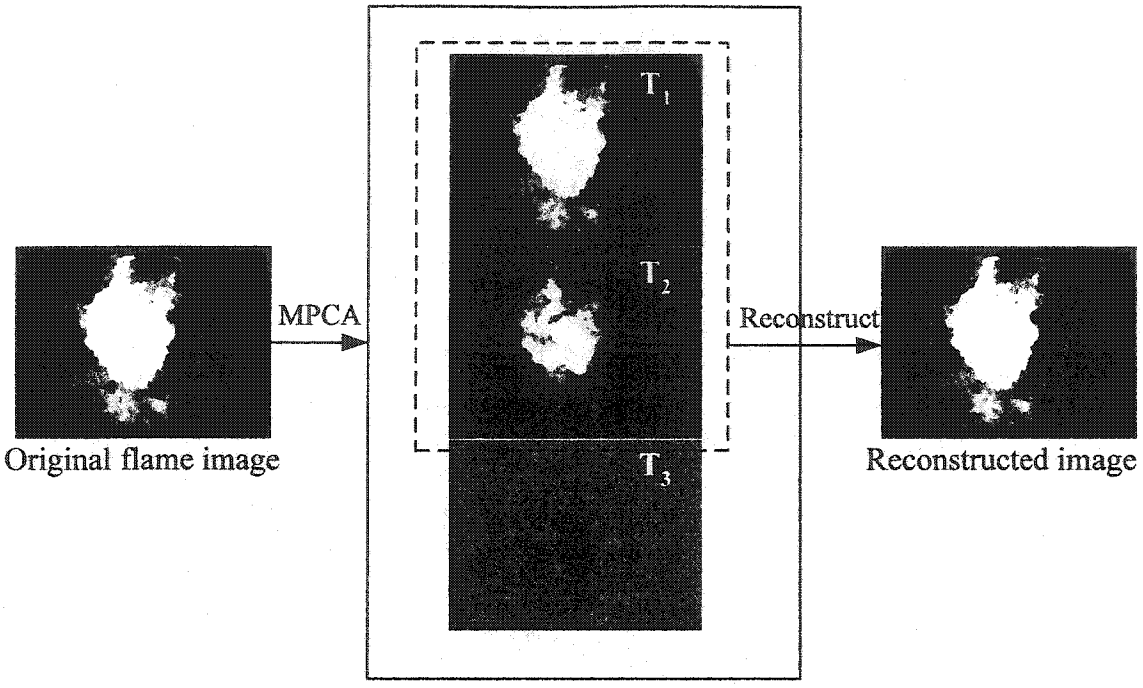
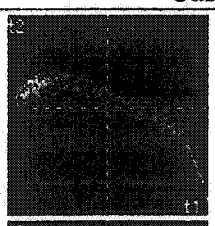
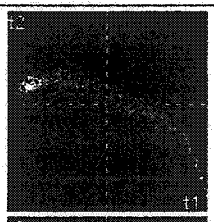
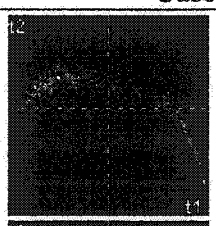
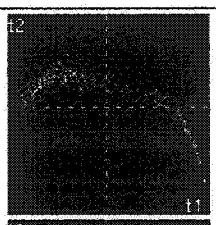
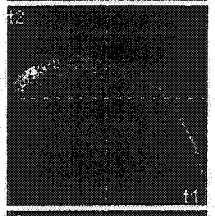
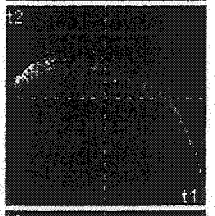
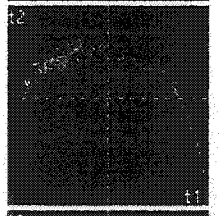
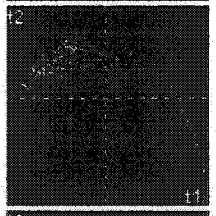
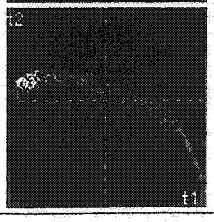
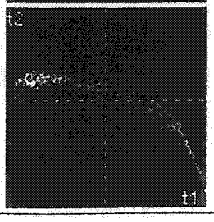
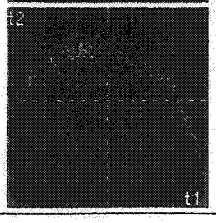
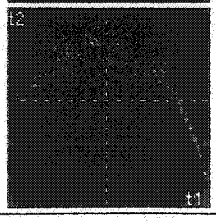



**Figure 6.4 Score images for a sample flame image**

**Table 6.2 Score plots for the sample images shown in Table 6.1 (order is the same)**

| | Case I | | Case II | |
|---|---|---|---|---|
| A |  |  | D |   |
| B |  |  | E |   |
| C |  |  | F |   |

## 6.5 Feature Extraction

As seen in the last section, the score plot contains important information about the flame and the corresponding process conditions; however, direct monitoring of the process based on the appearance of the score plot is not practical. This is because it is hard to monitor a time series process by watching changes in a two dimensional matrix, and even if one were able to detect some changes occurring in the score plot, it is hard to interpret such changes. Therefore, we need to extract features from the score plot that have more physical meaning and to further relate those features to the process variables to help understand more about the flames and the combustion process.

The features that are discussing here can be divided into two categories. The first category is called luminous features, including the flame luminous region area, flame brightness, uniformity of flame brightness and the average brightness of the non-luminous area. The second category is called color features, including average color of the whole image, average color of the flame luminous region and the number of colors appearing in the flame region.

### 6.5.1 Flame luminous region

The flame luminous region is extracted from the image by choosing a mask in the score plot. The boundary of the mask is easily obtained by a trial and error process, whereby one selects a mask area in the score plot, selects the pixels lying under it and highlights them in the image space, and iterates until one obtains a mask that segments the feature of interest. The final mask selected for extracting the flame luminous region is shown as the green area in Figure 6.6a. To illustrate the segmentation ability of the mask a sample flame image is shown in Figure 6.6b. If we set all pixels falling outside this mask to have a gray color, the image shown in Figure 6.6c is obtained. We can see that the luminous flame region is separated from the other part of the image.

Let us define a 256×256 binary matrix **M** that fits over the 256×256 score plot as a matrix which has an element equal to 1 if the element location lies under the luminous flame mask and 0 if it does not. This binary matrix **M** is used to define several useful features of the flame image.

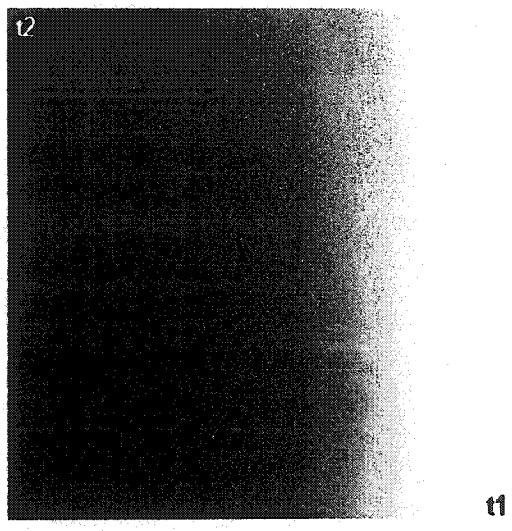



**Figure 6.5 The color plane for the $t_1$-$t_2$ score plot**

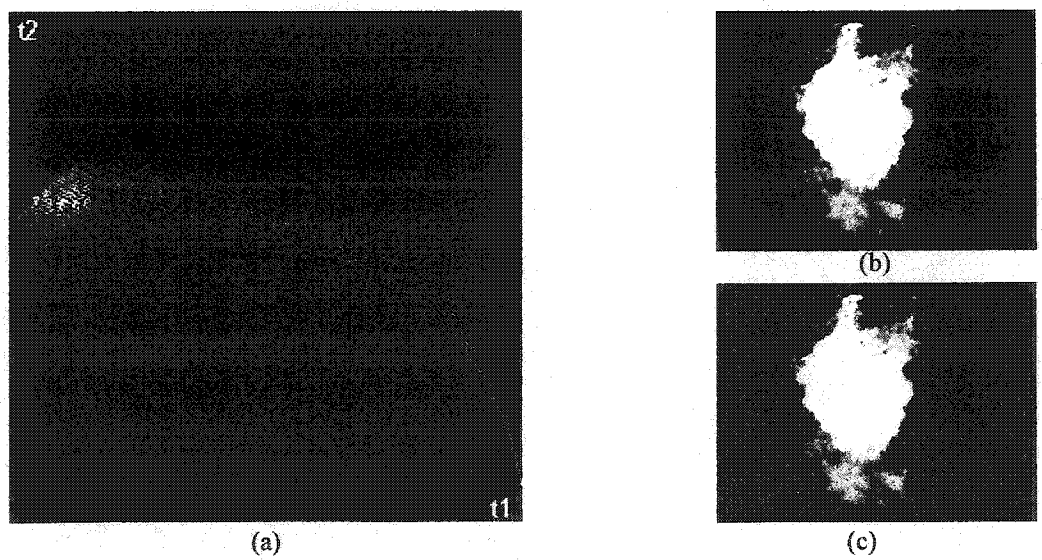



**Figure 6.6 An illustration of flame luminous region mask. (a) Score plot and luminous region mask, (b) One sample image, (c) The flame region decided by the mask**

### 6.5.2 Luminous features

Luminous region area ($A$)

The area of the flame image that corresponds to the luminous region can be easily computed by counting the number of the pixels falling into the flame mask. This can be defined mathematically using the binary matrix **M** as:

$$A = \sum_{i,j} TT_{i,j}, \quad \forall (i,j), M_{i,j} = 1$$

where **TT** is the 256×256 $t_1$-$t_2$ score plot (see section 2.5).

Flame brightness ($B$)

Flame brightness can be obtained by integrating the luminous intensity level contributed from all pixels falling inside the luminous region. The luminous intensity level for any location in the score plot is computed by converting the color plane obtained by Eq6-1 to a gray scale plane with elements corresponding to the luminous intensity at each location. The relationship is

$$\Omega_{i,j} = [R\ G\ B]_{i,j} \cdot \begin{bmatrix} 0.299 \\ 0.587 \\ 0.114 \end{bmatrix}, \quad i,j = 0,\ldots,255$$

In which $\Omega_{i,j}$ is the luminous intensity level for location $(i,j)$ in score plot, and the conversion coefficient vector is the one which converts the element RGB signal into a corresponding luminous intensity [Matlab Image Processing toolbox].

The overall flame brightness is then calculated as:

$$B = \sum_{i,j} TT_{i,j} \cdot \Omega_{i,j}, \quad \forall (i,j), M_{i,j} = 1$$

Uniformity of flame brightness ($U$)

The uniformity of the flame brightness is defined as the standard deviation of the flame brightness throughout the luminous region:

$$U = \sqrt{\frac{\sum_{i,j} TT_{i,j} \cdot \Omega_{i,j}^2 - \sum_{i,j} TT_{i,j} \cdot \Omega_{i,j}}{\sum_{i,j} TT_{i,j}}} = \sqrt{\frac{\sum_{i,j} TT_{i,j} \cdot \Omega_{i,j}^2 - B}{A}}, \quad \forall (i,j), M_{i,j} = 1$$

Average brightness of the non-luminous area ($W$)

As we can see from the sample images shown in Table 6.1, the camera installed is facing the flame, which means we can not see the length of the flame. Moreover, the flame image is a two dimensional projection of a three dimensional field and only part of the flame is captured in the image. Therefore, the brightness of the non-luminous area somehow gives information on the length and/or the volume of a flame. We compute the average brightness of the non-luminous area by:

$$W = \frac{\sum_{i,j} TT_{i,j} \cdot \Omega_{i,j}}{\sum_{i,j} TT_{i,j}}, \quad \forall (i,j), M_{i,j} = 0$$

### 6.5.3 Color features

Average color of the whole flame image ($s_{1m}$, $s_{2m}$)

The average color of the whole flame image is expressed as the average location of pixels in the score plot, which is the average values of scaled scores $s_1$ and $s_2$ (definition of $s_1$ and $s_2$ can be found in section 2.5).

$$s_{1m} = \frac{\sum_{i,j} TT_{i,j} \cdot i}{N}, \; s_{2m} = \frac{\sum_{i,j} TT_{i,j} \cdot j}{N}, \; i,j = 0,\ldots,255$$

in which $N$ is the total number of pixels in the image.

Average color of the flame luminous region ($s_{1f}$, $s_{2f}$)

The average color of the flame luminous region is defined as the average location of the pixels belonging to the flame luminous region, which are the average $s_1$ and $s_2$ values for pixels in the flame luminous region.

$$s_{1f} = \frac{\sum_{i,j} TT_{i,j} \cdot i}{A}, \quad s_{2f} = \frac{\sum_{i,j} TT_{i,j} \cdot j}{A}, \forall (i,j), M_{i,j} = 1$$

Total number of the colors appearing in the flame luminous region ($N_c$)

The number of the different colors appearing in the flame region is the area of the flame region in the score plot space.

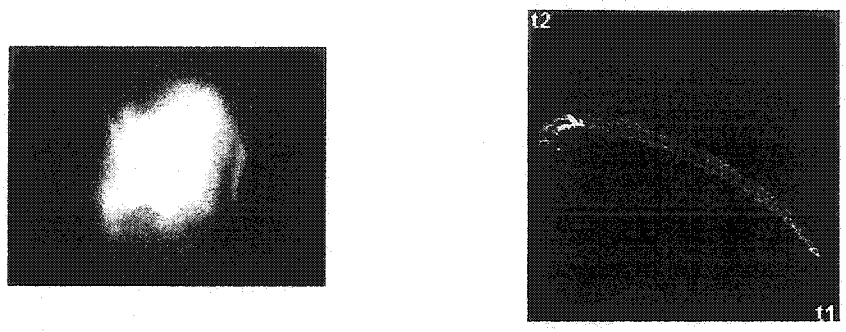
$$N_c = \sum_{i,j} 1, \quad \forall (i,j), (TT_{i,j} \cdot M_{i,j}) \neq 0$$

### 6.5.4 Filtering the feature variables

Because of the highly turbulent nature of the flame the calculated feature variables have a large variation from image to image. Furthermore, since the process characteristics are not changing rapidly, one is usually not interested in monitoring on a one second basis. Therefore, some form of filtering should be performed to reduce the variability of the extracted features.

There are several possible filtering techniques that can be used at the different stages of the calculation. The first one is to perform the filtering operation directly in the image space. This is the mostly common used preprocessing step used in the flame studies from the literature [Shimoda et al., 1990; Lu et al., 2000]. However, in the highly turbulent circumstances, averaging in the image space leads to a loss of the flame's characteristics. An example of an averaged image is shown in Figure 6.7a. It can be seen

that the shape of the score plot for the averaged image (Figure 6.7b) has also been 'distorted' compared to the single frame images (the score plots of point 'A' in Table 6.2).




(a) (b)

**Figure 6.7 Effect of averaging flames in the image space. (a) An example averaged image (over 60 consecutive images), (b) The corresponding score plot**

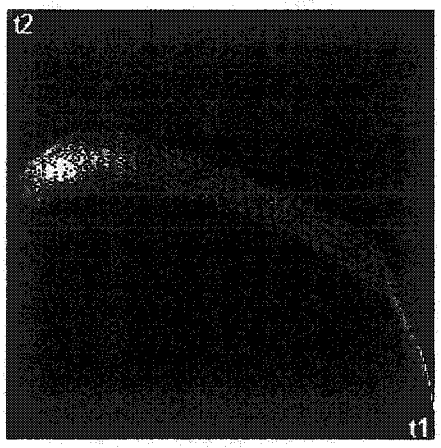



**Figure 6.8 An averaged score plot (over 60 consecutive images)**

A second filtering technique is to perform filtering in the score plot space. This represents a much better solution because, as shown earlier, the score plot for each rapidly changing flame image remains very stable as long as the process conditions remain unchanged. Figure 6.8 shows an averaged score plot over 60 consecutive images. Compared to Figure 6.7b, this averaged score plot keeps the basic shape of the score plot of the individual images ('A' in Table 6.2). The feature variables extracted from the averaged score plot are expected to summarize the characteristics of the flame during the

short period of time over which the average is computed. However, the values of certain features extracted from these filtered score plots may have a different level from the features calculated directly from the raw images because some of the feature extraction calculations are not linear.

A third filtering approach is to apply a time domain filter to the extracted feature variables of each individual frame image. This approach has some advantages compared to averaging the score plots. First, it is much easier to handle the time series data such that at each time point, the data is a vector rather than a matrix. Second, the filtered feature values are directly related to the raw feature variables. The filter used here is an averaging filter with a window length 60.

## 6.6 Results

### 6.6.1 Feature variables for two case studies

Figure 6.9 plots three feature variables for the two case studies. Raw features computed from each individual image are shown as the dark grey lines. Two types of filtered features are also shown: filtered values of the individual features from each image (indicated by the lighter grey lines) and features computed from the averaged score plots (indicated by the black lines). Filtered data by both filters are almost identical for all the feature variables (e.g. Figure 6.9a and 6.9c), except for number of the colors ($N_c$) in the flame luminous region (Figure 6.9b). The values of $N_c$ computed from the averaged score plot are much higher than those obtained by filtering the $N_c$ values from the time series of the individual image values. However, the two filtered values have similar trends. This is expected since the total number of colors in all 60 images averaged in the score plot must be much larger than that in any individual image. In the following computations, only filtered data from the individual image feature time series are used. Using filtered features obtained from the average score plot leads to similar results.

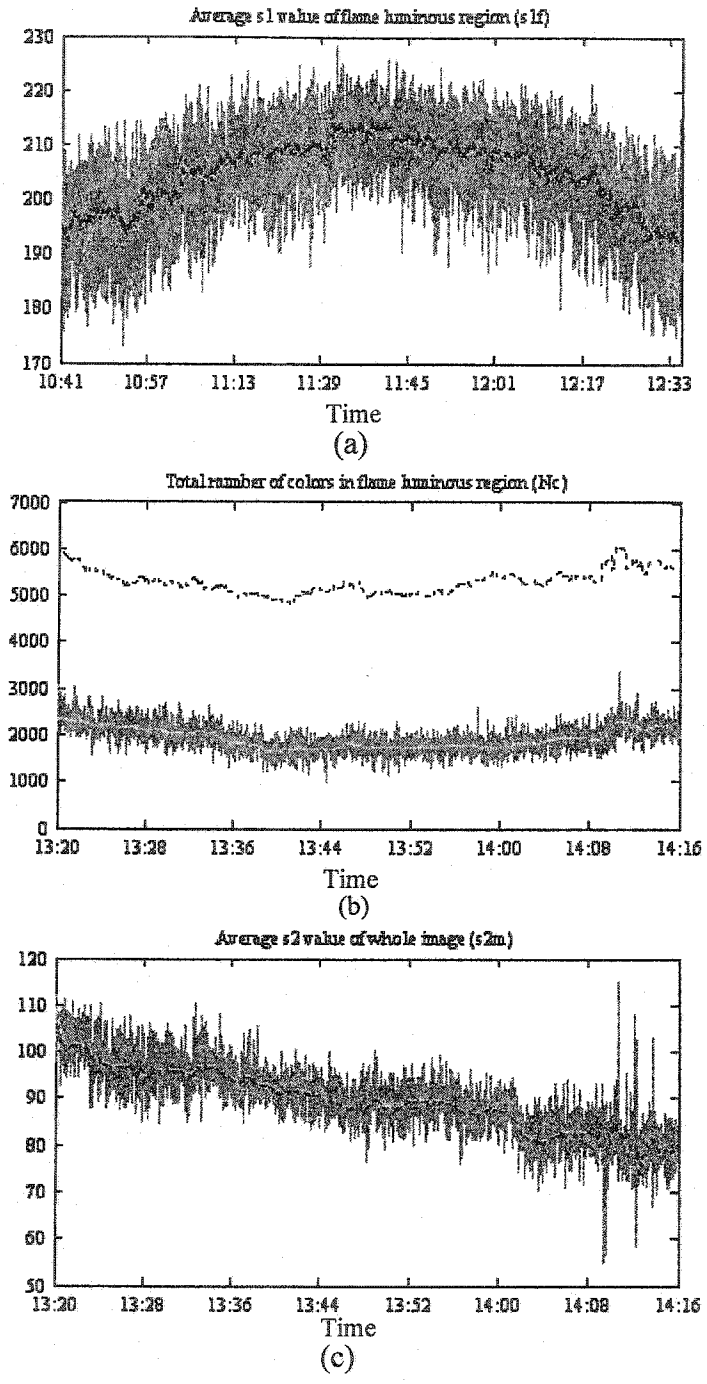



**Figure 6.9 (a) One of the feature variables for case I: Average $s_1$ value of flame luminous region ($s_{1f}$). (b) One of the feature variables for case II: Total number of colors in flame luminous region ($N_c$). (c) One of the feature variables for case II: Average $s_2$ value of whole image (s2m). The darker gray lines represent raw values from each 1 second frame image, the lighter gray lines represent the filtered values obtained from directly filtering the raw feature variables and the black lines represent the filtered values obtained from averaged score plot.**

In Case I, the liquid fuel flow rate first decreased and then increased back to the original value. All the feature variables (e.g. Figure 6.9a) follow the same trend or the inverse trend of the liquid fuel flow rate change in Figure 6.2.

In Case II, both the liquid fuel and natural gas flow rate were changing. In this case the trends of feature variables are different. Some of the trends are similar to the liquid fuel flow rate change, such as $s_{2m}$ (average $s_2$ value of the image, see Figure 6.9c) while other feature trends are similar to the natural gas flow rate change, such as $s_{2f}$ (average $s_2$ value of the flame luminous area). To get a better understanding of the relation between these image feature variables and the process measurements (steam flow and fuel flow rates), Principal Component Analysis (PCA) and Partial Least Squares (PLS) studies are performed in the next section.

## 6.7 Analysis of the Data using Multivariate Statistical Methods

### 6.7.1 Analysis using PCA

In this section, two PCA models are obtained, one for the Case I data and another for the Case II data. The purpose is to reveal the relationship between the changes of the fuel flow rates and the flame properties. The variables included in the PCA models are the 9 filtered flame feature variables and the fuel flow rates.

In the PCA model for case I, the first principal component explained 88.4% of the total variance. This is reasonable since only one factor (flow rate of liquid fuel) is changing in this case. The loading plot of PCA (Figure 6.10) shows the relative importance of the variables to this principal component. All the features contribute and are highly correlated with the fuel flow. We can see that when the liquid fuel flow increases so does the flame brightness ($B$), the luminous area ($A$), etc., while at the same time the average scores ($s_{1f}$, $s_{2f}$) decrease.

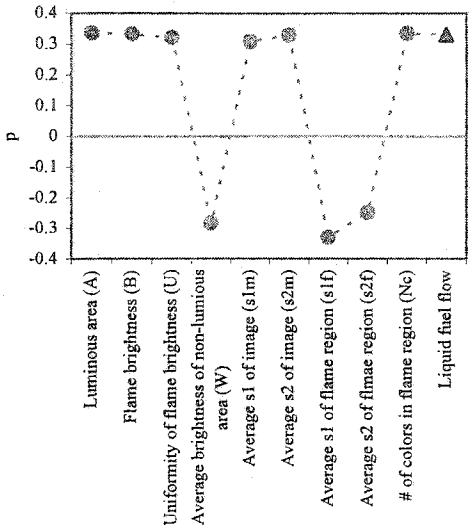



**Figure 6.10 loading plot of first component of PCA model for case I**

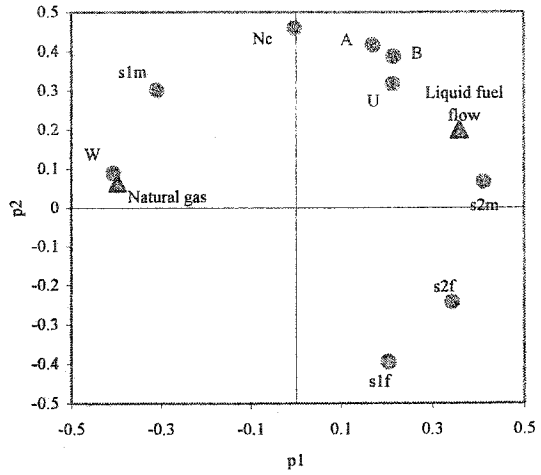



**Figure 6.11 $p_1$-$p_2$ loading plot of PCA model for the data of case II**

The PCA model for the case II data has two significant components since there are two sources of variation. The first component explained 52.3% and the second component explained 42.6% of the total variance. From the scatter ($p_1$-$p_2$) loading plot in Figure 6.11 we can see that $s_{2m}$ (average $s_2$ value of the image) has positive correlation with liquid fuel flow rate, $W$ (average brightness of the non-luminous area) has positive correlation and $s_{2f}$ (average $s_2$ value of the flame luminous area) has negative correlation with natural gas flow rate. This conclusion is confirmed by the feature variable time series plots (e.g. $s_{2m}$ shown in Figure 6.9c has a similar trend to the change of liquid fuel flow rate shown in Figure 6.3a).

### 6.7.2 Prediction of flow rate of steam generated

In this section, a single linear PLS model is built using the combined data from both cases to demonstrate that information extracted from the flame images is also sufficient to obtain an accurate prediction of the steam flow rate from the boiler. All 9 feature variables from the flame images are used as predictors. 33 observations are used as the training set and 133 observations are used as the test set.

Eight latent variables were determined to be significant by cross validation. The root mean square of prediction error (RMSEP) for the training set is 0.398 Kg/s and is 0.498 Kg/s for the test set, demonstrating that the model can predict results from new images almost as well as it fits the training data. The prediction vs. observation plot is shown in Figure 6.12. Figure 6.13a and 6.13b are the time series plots of the predicted and observed steam flow rates for the two case studies, respectively. We can see that the steam flow rates are well predicted using only information from the flame images.

## 6.8 Conclusions

Methodologies for monitoring flame have been developed for an industrial steam boiler. Experimental results demonstrate that the system is capable of characterizing the luminous and color parameters of a flame qualitatively and quantitatively and can be used to predict the performance of the boiler system over a wide range of conditions.

The rapidly time varying flame images were shown to give very stable score plot histograms in the principal component space. These score plots were very stable for the flames imaged under constant process conditions, but changed in a consistent way whenever the process conditions changed. A masking method was used to define a flame luminous region. A number of features of the flames were then extracted and these features were shown to be highly related to the process fuel feed rates. Using these features as regressors, a Partial Least Squares (PLS) regression model was developed to accurately predict the steam flow rate from the boiler. Although, by itself, the prediction of the steam flow rate is not important (this flow rate is easily measured), the fact that it could be predicted so well just from two-dimensional RGB images of the boiler flame, using no other information, is extremely important. It demonstrates that there is a tremendous amount of information in these images, which if combined with additional process data, could greatly improve the inferential models typically used to monitor the operating performance and emissions from these combustion processes. In Chapter 7,

three inferential models for predicting the heat of combustion of liquid fuel and the emissive concentrations of $NO_x$ and $SO_2$ are developed.

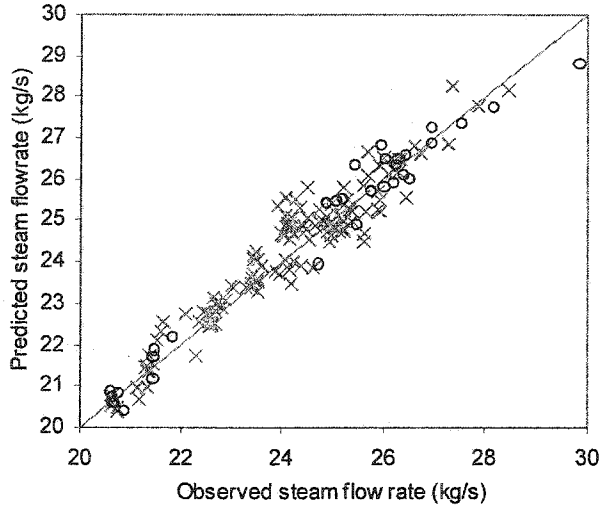



**Figure 6.12 Predicted vs. observed steam flow rates (○: Training set; ×: Test set)**

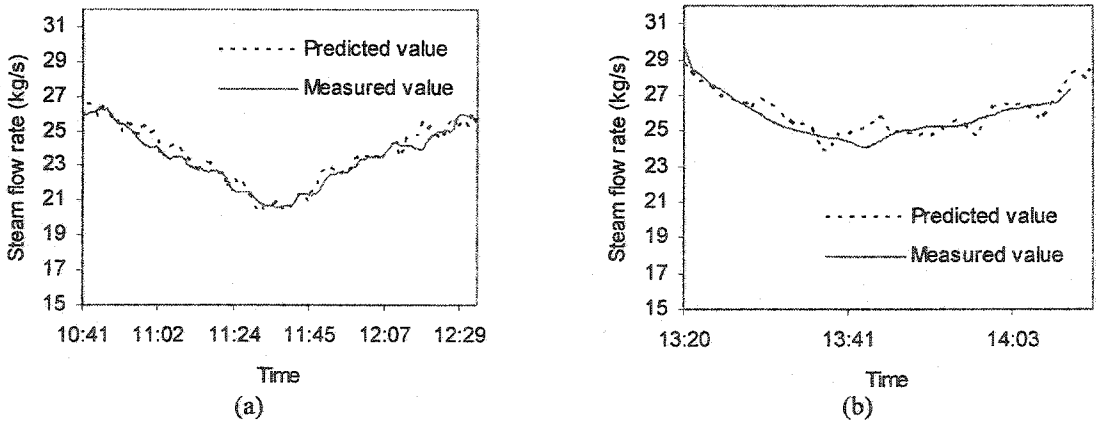



**Figure 6.13 Time series plots of predicted and measured steam flow rates. (a) Case I, (b) Case II**

# Chapter 7

# Flame Application (Part II)

## — Prediction of Fuel and Gaseous Emission Properties

In Chapter 6, methodology for extracting features from highly turbulent flame images was developed and a PLS model was shown to give good prediction of steam flow rate using these feature variables. This clearly demonstrated the power of multivariate image analysis applied to industrial flame data. This chapter considers the potential use of this approach to predict meaningful economic and environment variables, such as the heat of combustion of the liquid waste fuel stream and the concentrations of $NO_x$ and $SO_2$ in the off-gas from the boiler system.

## 7.1 Introduction

As mentioned in Chapter 6, the boiler studied in this thesis uses waste streams of other processes as part of the fuel source. The purpose for doing this is twofold: to recover energy, and more importantly, to get rid of waste materials that can not be directly disposed of to the environment.

One characteristic of the liquid waste stream that often changes dramatically is its composition, leading to large variations in the heat of combustion. For better control performance, a sensor that can measure the heat of combustion of this incoming waste liquid stream would be useful. In this chapter, an inferential sensor for predicting heat of

combustion is developed using features extracted from flame images as well as the flow rates of liquid fuel and natural gas.

The other problem considered in this study is to predict the gaseous emission of pollutants, such as $NO_x$ and $SO_2$. Two major mechanisms exist for $NO_x$ formation in combustion processes. The thermal $NO_x$ formation mechanism involves thermal dissociation and subsequent reaction of nitrogen and oxygen molecules in combustion air, while the fuel $NO_x$ formation mechanism is based on oxidation of fuel-bound nitrogen contained in certain fuels. $SO_2$ is formed when the sulfur contained in fuels combines with the oxygen during combustion. $NO_x$ and $SO_2$ are called precursors to acid deposition, because, under the right set of conditions, they react with other chemicals in the atmosphere to form nitric acid and sulfuric acid, respectively. These two acids do not accumulate in the atmosphere, but are absorbed by rain droplets, resulting in, 'acid rain'. Therefore, the reduction of $NO_x$ and $SO_2$ emissions is a very serious problem in combustion processes. Industrial emission sources such as boilers and furnaces are required to be equipped with Continuous Emission Monitoring Systems (CEMS). However, a hardware CEMS (such as analyzer) is usually expensive and so a software CEMS based on an inferential sensor is then very attractive.

Several studies have been carried out using inferential models to predict $NO_x$ concentration. Qin et al. [1997] built soft sensors for the prediction of $NO_x$ emissions from a boiler process using Principal Component Regression (PCR) and Neural Network Principal Component Regression (NNPCR). Seven process variables, such as air flow rate and fuel flow rate, were used as inferential variables. Krijnsen et al. [1999] predicted $NO_x$ emissions from a transiently operating diesel engine using a neural networks. Wang et al. [2002] reported a method to predict $NO_x$ emissive concentration for a coal boiler by using color flame images and a neural network. In this last method, the color images were used to estimate the temperature inside the boiler through a bicolor model. Three characteristic parameters related to average temperature and temperature distribution, one characteristic parameter describing the length of the warm-up area, and two process

operating parameters including excessive air amount and fuel composition are used as the input to the neural networks. In this chapter, preliminary studies on building models to predict $NO_x$ and $SO_2$ emissive concentrations are presented using both process measurements and color image data.

## 7.2 Dataset

Twelve video tapes were collected during a 2 month period of time (from September 9 to November 13, 2002). Each video tape contained a half an hour of recorded flame images. The average process variables during the time when the video tapes were recorded were also collected. This dataset became available when the final version of this manuscript was almost complete, therefore only preliminary studies were conducted and are presented here.

**Table 7.1 Process variables for 12 samples**

| | # | Name | Min | Max | Average | Std. | Unit |
|---|---|---|---|---|---|---|---|
| Flow rate | 1 | Natural gas flow rate | 1.08 | 1.59 | 1.20 | 0.16 | scm/s |
| | 2 | Fuel flow rate | 1.13 | 1.85 | 1.44 | 0.27 | Kg/s |
| | 3 | Oxygen flow rate | 0.31 | 2.45 | 2.18 | 0.59 | scm/s |
| | 4 | Steam flow rate | 22.78 | 30.71 | 26.41 | 2.64 | Kg/s |
| Pressure | 5 | Drum pressure | 3.23 | 3.45 | 3.30 | 0.03 | MPa |
| | 6 | Steam pressure | 3.17 | 3.29 | 3.22 | 0.03 | MPa |
| Temperature | 7 | Feed oxygen temperature | 1.05 | 14.59 | 5.69 | 4.23 | °C |
| | 8 | Steam temperature | 238.42 | 240.31 | 239.2 | 0.52 | °C |
| | 9 | Saturated steam temperature | 220.08 | 238.45 | 234.55 | 6.60 | °C |
| | 10 | Feed water temperature | 128.78 | 138.67 | 132.68 | 2.95 | °C |
| Emissive concentration | 11 | $NO_x$ concentration | 88.46 | 243.3 | 159.4 | 40.86 | ppm |
| | 12 | $SO_2$ concentration | 0 | 8.32 | 1.41 | 2.29 | ppm |
| Heat of combustion of liquid fuel | 13 | Heat of combustion – gross | 17.39 | 21.69 | 19.65 | 1.45 | MJ/Kg |
| | 14 | Heat of combustion – net | 15.47 | 19.69 | 17.59 | 1.42 | MJ/Kg |

Table 7.1 lists the information on the process variables. Figure 7.1 shows 12 sample images and their corresponding $t_1$-$t_2$ score plots. One image is selected from each video tape. The loadings of the principal components are the ones obtained in Chapter 6. Not too much can be seen from the flame images. However in the score plots, we can see

that the shapes and the locations of the pixels do have slight differences from sample to sample. However, nothing can be concluded directly from these score plots.

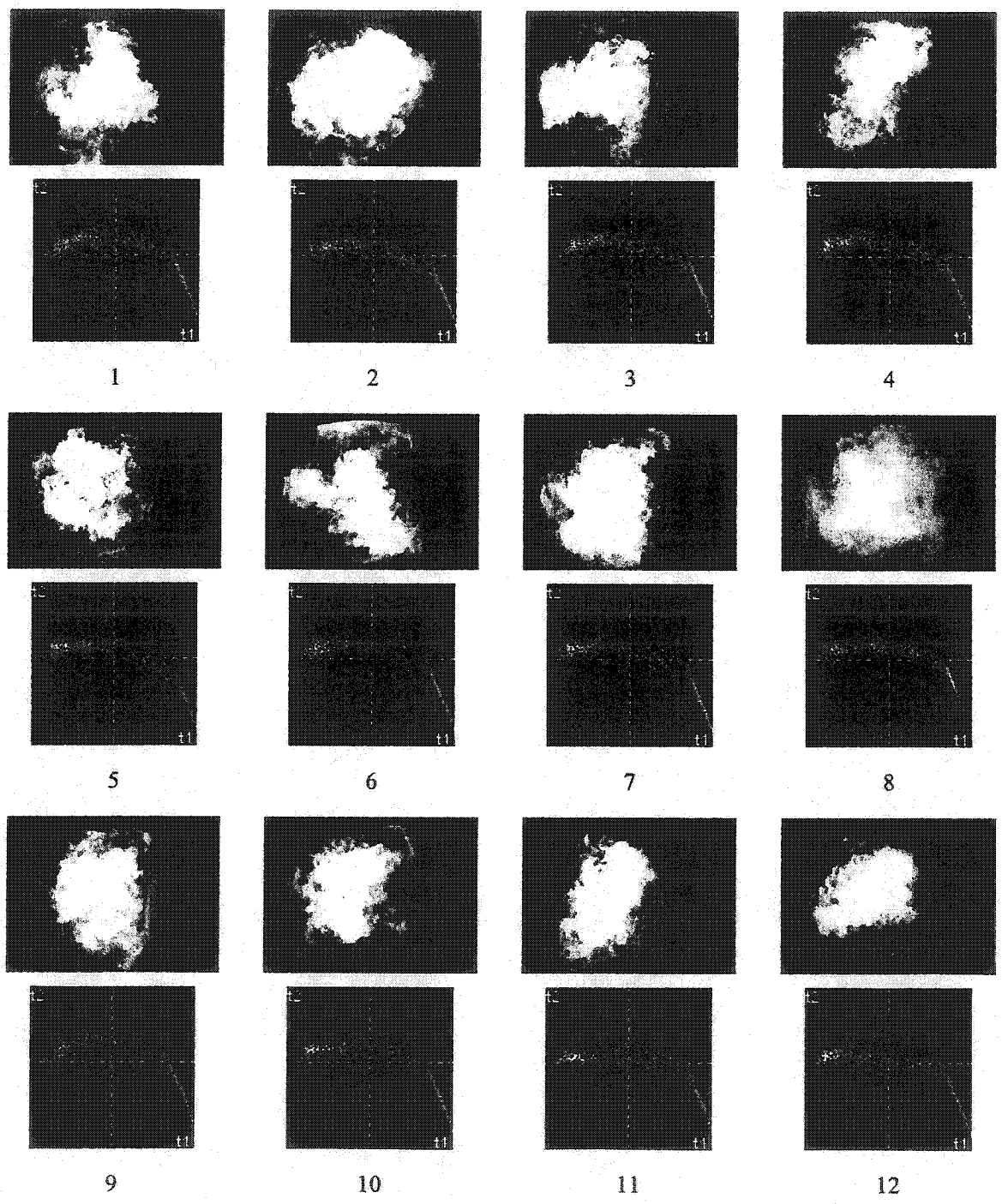



**Figure 7.1 Sample images and their score plots**

## 7.3 Prediction of Heat of Combustion of Waste Fuel Stream

Nine feature variables can be computed for each flame image as described in Chapter 6. Even though the samples were collected almost two years later than the samples in case I (see section 6.3.1), the flame luminous region mask developed in Chapter 6 was still adequate. An average feature vector was used to represent the image data of each tape.

Similar to the prediction of the steam flow rate in Chapter 6, a simple PLS model is built to predict the product of heat of combustion and the liquid fuel flow rate. The heat of combustion can then be obtained by dividing the PLS model prediction by the liquid fuel flow rate. This is shown in equation (7-1):

$$\hat{Q} = \mathbf{v} \cdot \hat{\boldsymbol{\gamma}}, \text{ where } Q = H_{Lf} \cdot F_{Lf}$$
$$\hat{H}_{lf} = \hat{Q} / F_{lf} \tag{7-1}$$

where $\mathbf{v}$ is the feature vector extracted from the image data, $\hat{\boldsymbol{\gamma}}$ is the model regression coefficient vector, $H_{Lf}$ is the heat of combustion and $F_{lf}$ is the flow rate of liquid fuel.

We can also incorporate the flow rates of liquid fuel and natural gas into the predictors.

$$\hat{Q} = \left[F_{lf} \;\; F_{ng} \;\; \mathbf{v}\right] \cdot \hat{\boldsymbol{\gamma}} \tag{7-2}$$

in which $F_{ng}$ is the natural gas flow rate.

Let us denote the model from equation (7-1) as Model 1 and the model from equation (7-2) as Model 2. Either the gross heat of combustion or the net heat of combustion (net heating value does not include the latent heat of water from combustion that condenses in the test procedure) can be used as response variable. Similar prediction results can be obtained. In the thesis, only prediction results of the net heat of combustion are reported.

Ten samples are used as a training set and 2 samples are used as a test set. 6 latent variables and 8 latent variables are selected for Model 1 and Model 2 respectively by cross validation. Figure 7.2a and b shows the prediction vs. observation plots for both models, as well as the values of Root Mean Square Prediction Error (RMSPE). Figure 7.3a and b show the time series plots. We can see that by including the information of fuel flow rates Model 2 has better performance than Model 1.

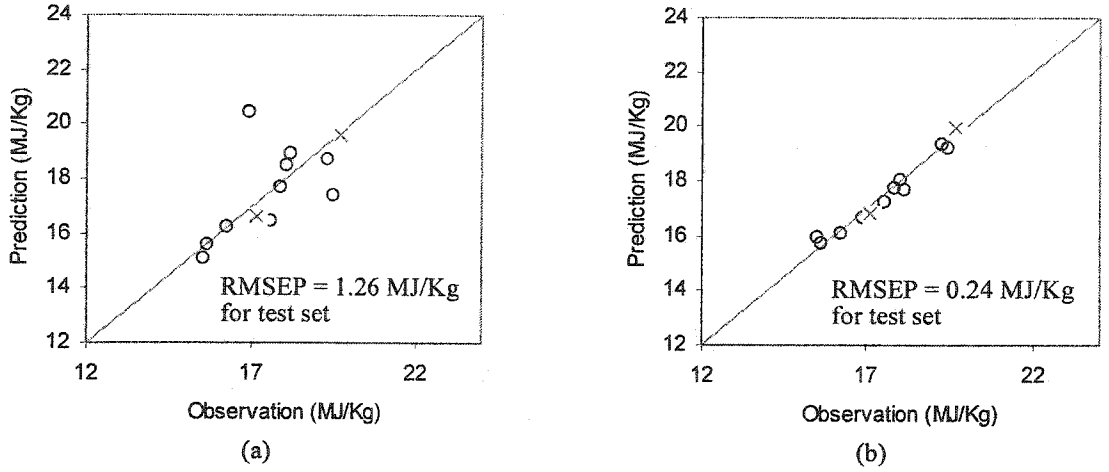



**Figure 7.2 Prediction vs. observation plots for prediction of net heat of combustion of liquid fuel. (a) Model 1: using only image data, (b) Model 2: using image data and fuel flow rates. (○: Training set; ×: Test set)**

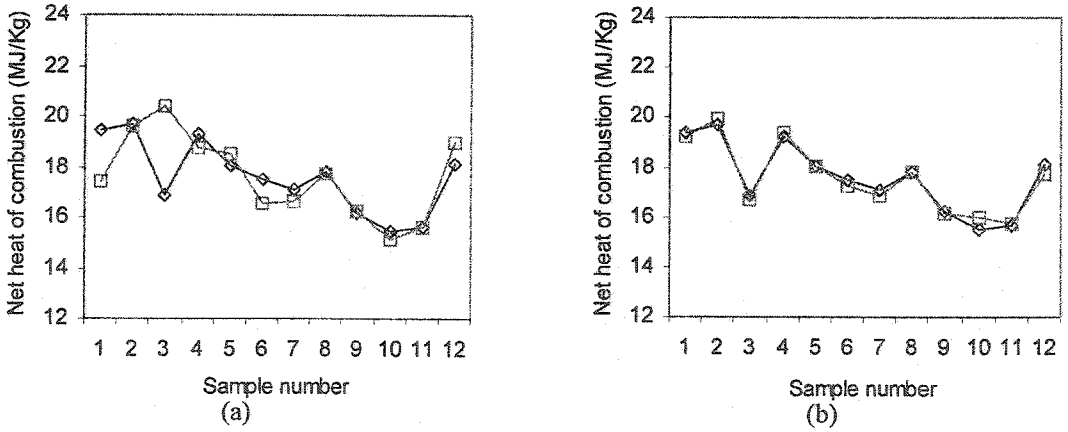



**Figure 7.3 Time series plots for the prediction of net heat of combustion of liquid fuel. (a) Model 1: using only image data, (b) Model 2: using image data and fuel flow rates. (◊: Measured data; □: Predicted data)**

## 7.4 Prediction of the Emissive Concentrations of $NO_x$ and $SO_2$

PLS models are developed in this section to predict the emissive concentrations of $NO_x$ and $SO_2$ respectively. Three types of predictors are considered for predicting each of the concentrations: i) using only process variables; ii) using only feature variables extracted from the image data; and iii) using both types of variables. In predictor type (i) and (iii), only process variables having relatively large variations among the 12 samples are included in the predictors. These include temperature of the feed oxygen, and the flow rates of liquid fuel, natural gas, oxygen and steam. These models (6 in total) are listed in Table 7.2. For each model, 10 samples are used as a training set while 2 samples are used for evaluation (test set).

**Table 7.2 Model names of the six models for predictions of $NO_x$ and $SO_2$ emissive concentration**

| Predictors / Response | Process variables | Feature variables from image data | Both process variables and feature variables from image data |
|---|---|---|---|
| $NO_x$ concentration | Model_$NO_x$_1 | Model_$NO_x$_2 | Model_$NO_x$_3 |
| $SO_2$ concentration | Model_$SO_2$_1 | Model_$SO_2$_2 | Model_$SO_2$_3 |

**Table 7.3 Summary of the six models for prediction of emission $NO_x$ and $SO_2$ concentration**

| | Number of LV's | RMSEP for training set | RMSEP for test set |
|---|---|---|---|
| Model_$NO_x$_1 | 5 | 18.97 | 19.96 |
| Model_$NO_x$_2 | 6 | 9.10 | 15.47 |
| Model_$NO_x$_3 | 7 | 13.12 | 25.83 |
| Model_$SO_2$_1 | 1 | 3.94 | 3.11 |
| Model_$SO_2$_2 | 7 | 0.41 | 0.27 |
| Model_$SO_2$_3 | 8 | 0.29 | 2.72 |

The number of latent variables (determined by cross validation) and RMSEP for both training set and test set of each of the six models are listed in Table 7.3. We can see that the best models are obtained by using only the feature variables extracted from the image data as predictors. It seems that model predictions are not improved by using process data.

The prediction vs. observation plots and time series plots for Model_$NO_x$_2 and Model_$SO_2$_2 (using only the feature variables extracted from the image data) are shown in Figure 7.4 and Figure 7.5. The results show good agreement between the predicted and the measured data.

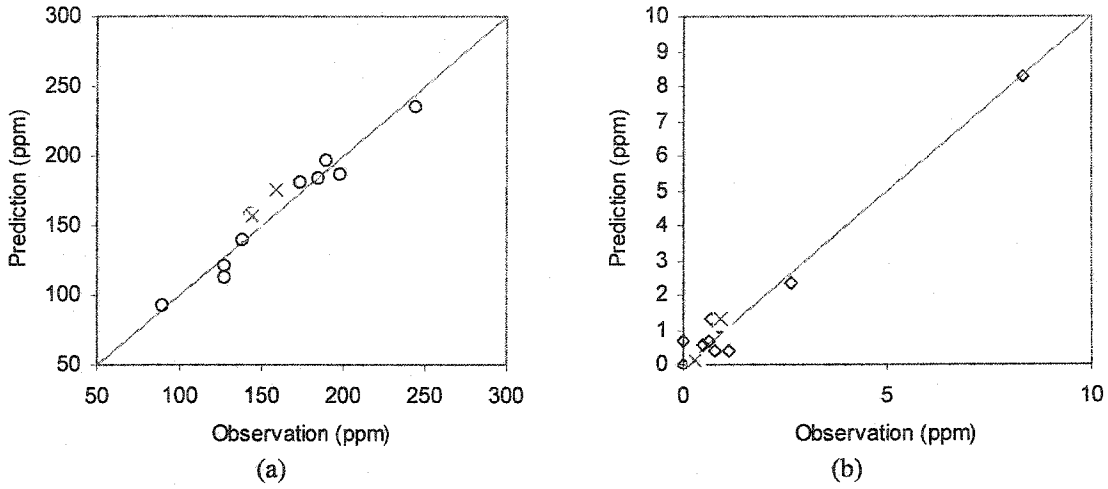



**Figure 7.4 Prediction vs. observation plots for the $NO_x$ and $SO_2$ concentrations. (a) $NO_x$ Concentration, (b) $SO_2$ Concentration. (○: Training set; ×: Test set)**

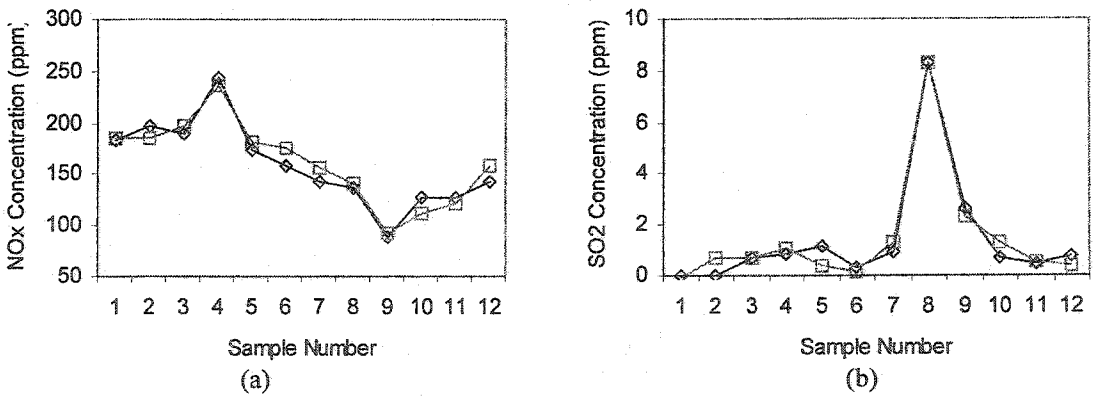



**Figure 7.5 Time series plots for estimation of the $NO_x$ and $SO_2$ concentrations. (a) $NO_x$ Concentration, (b) $SO_2$ Concentration. (◊: Measured data; □: Predicted data)**

## 7.5 Discussion and Conclusions

In this chapter, a preliminary study on the prediction of the heat of combustion of liquid waste fuel and emissive concentrations of $NO_x$ and $SO_2$ using feature variables extracted from flame images and process variables is developed.

Two PLS models are developed for predicting the heat of combustion of liquid fuel: one uses only feature variables from images as predictors and the other uses feature variables as well as flow rates of liquid fuel and natural gas. The results show that the latter one has a better prediction performance.

In building the models to predict concentrations of $NO_x$ and $SO_2$ emissions in the off-gas, three types of predictors are compared: i) using process measurements; ii) using feature variables extracted from flame images; and iii) using both process measurements and feature variables. The results in some sense are surprising because the models developed using only feature variables extracted from the images have the best performance and the predictions are quite good. It seems that the information in the flame images has much more correlation with the $NO_x$ and $SO_2$ concentration than with the information in the process measurements. This is more evident when predicting $SO_2$ concentration.

The models developed in this chapter show promising results and verify the great potential for utilizing flame images. However, these preliminary results on an image-based soft sensor need to be validated using much more data under different operating conditions.

# Chapter 8

# Summary and Conclusions

## 8.1 Summary and Conclusions

In this thesis practical methodologies for building inferential sensors based on multivariate image analysis are developed. Multivariate statistical methods, such as Principal Component Analysis (PCA) and Partial Least Squares (PLS), are the key methods used for handling the highly correlated data and for extracting useful information from images.

Two industrial applications are studied.

The first application is to an industrial snack food production process. In this application, an inferential sensor based on RGB color images for measuring the coating content of snack foods is developed. In Chapter 3, the methodologies are presented using three different types of snack foods as examples. Six feature extraction methods, including two overall features and four distribution features, are proposed. All six methods are shown to have good predictions when the test images have the same size as the training images. However, only method 6 (a method based on segmentation of the score plot based on covariance properties) shows robustness to smaller image sizes. Two approaches, an approach obtained directly from method 6 and an approach based on a small window strategy, to estimate the coating distributions are proposed. In order to obtain robust prediction in on-line operation, a procedure to remove background pixels (mainly the conveyor belt) by defining a product mask in the score plot space is proposed. A novel way to locate such a product mask is developed.

The models developed in Chapter 3 (by method 6) have been implemented in several industrial production lines. Some of the on-line results are shown in Chapter 4. The prediction of the coating concentration has shown good agreement with the measurements obtained from the laboratory. Several plant tests show clearly how the coating concentration variations can be tracked by the image-based sensor, and how process problems can be detected. Finally, results of closed-loop feed back control are presented. We compare the process performances under manual control and automatic control and show that using closed loop control with the image-based sensor the control performance is effectively improved.

Variation of the vision system components, such as a drift of lighting intensity, is always a strong factor that influences the performance of the model. In the first part of Chapter 5 some of the robustness issues are addressed. Three methods are presented to detect variations of the vision system and test for the validity the model, and two methods are presented for on-line model correction. A new self validation and correction approach based on monitoring changes in the $T_3$ residual score image is shown to provide a powerful tool to obtain a robust prediction of the coating concentration.

In the second part of Chapter 5, preliminary results for the prediction of organileptic properties (taste and texture properties) are presented. Four types of organileptic properties are studied: blister level, toast point level, taste property and peak break force. Color variation caused by coating concentration is first removed by projecting the image onto a subspace obtained by PCA. A series of 9 Laplacian of Gaussian spatial filters with different window size are then performed on the score images to obtain the texture information. A PLS model is used to predict four organileptic properties and the model prediction results are shown to be promising.

The second industrial application involves monitoring turbulent nonpremix flames in an industrial boiler and is presented in Chapter 6 and Chapter 7. In Chapter 6, basic methodologies are developed using two case studies. It has been shown that in the $t_1$-$t_2$ score plot space computed from PCA; fairly stable information can be obtained even

though in the image space, the flame is always bouncing around. A mask is selected in the $t_1$-$t_2$ score plot to indicate the location of flame luminous area. Nine feature variables, including four luminous features and five color features, are extracted from each flame image. To reduce the variation caused by highly turbulent conditions, three filtering techniques are also discussed. Taking averages in the image space, an approach which is often used in literature, is shown to lead to a significant loss of information. Good results are shown to be obtained either by filtering in the score plot space or by filtering the extracted feature variable time series. The use of PCA and PLS is also illustrated. PLS models are built to predict steam flow rate. Good prediction can be achieved by only using the image data.

In Chapter 7, the methodologies developed in Chapter 6 are used for predicting the heat of combustion of the liquid fuel whose composition often has large variation because it comes from waste streams of other processes. The results show that good prediction can be obtained when using both the feature variables extracted from images and the flow rates of liquid fuel and natural gas as predictor variables. Another type of inferential sensor is also developed which is for estimating the concentrations of emissive pollutants, such as $NO_x$ and $SO_2$. The inferential models that use only feature variables extracted from the image data give good prediction of both $NO_x$ and $SO_2$. It appears that the flame image data contains much more information on the $NO_x$ and $SO_2$ concentrations than the process data. This is more evident when predicting $SO_2$ concentration. The research presented in Chapter 7 reveals a great potential for utilizing the flame images. These flame images, which are always displayed on the monitors in the operation room, can give us much more information about the combustion process than people have realized.

## 8.2 General Framework

Methodologies developed in this thesis are for different applications. However, they can be summarized into a general framework (see Figure 8.1).

Figure 8.1 shows a model procedure to predict process/product properties using multivariate images. Two basic steps are required: feature extraction and regression. The regression step can use any appropriate regression method. PLS is used in this thesis.

Extracting appropriate feature vectors from the multivariate image is the key step to obtain a successful model. Three types of feature variables are considered. Overall features (see section 3.4.3), describes the properties of each channel ('variable') of the multivariate image. The properties can simply be average values (method 1 in section 3.4.3), standard deviations, moments or other statistical measurements computed from all the pixels in that channel. Overall features can also be obtained from the loading vectors of PCA (method 2 in section 3.4.3).

Distribution features (see section 3.4.4) can be obtained from histogram bins that are defined in the score plot space as masks. By counting the number of pixels falling into each mask, a histogram or cumulative histogram can be formed. Selection of masks is problem dependent and is not unique. However, for successful model building, it is recommended that pixels falling into each histogram bin should not have large variations in their property values. For example, in section 3.4.4, when we derive method 3 to method 6 for snack food application, the goal of choosing histogram bin is that pixels falling in each bin have similar coating content.

The third type of feature variables are computed from the properties of different regions of the image. The regions are segmented by choosing proper masks in the score plot. One such example is the feature extraction method developed in section 6.4 for the flame application, where a flame luminous region is first separated from non-luminous region and nine feature variables are computed from these two regions.

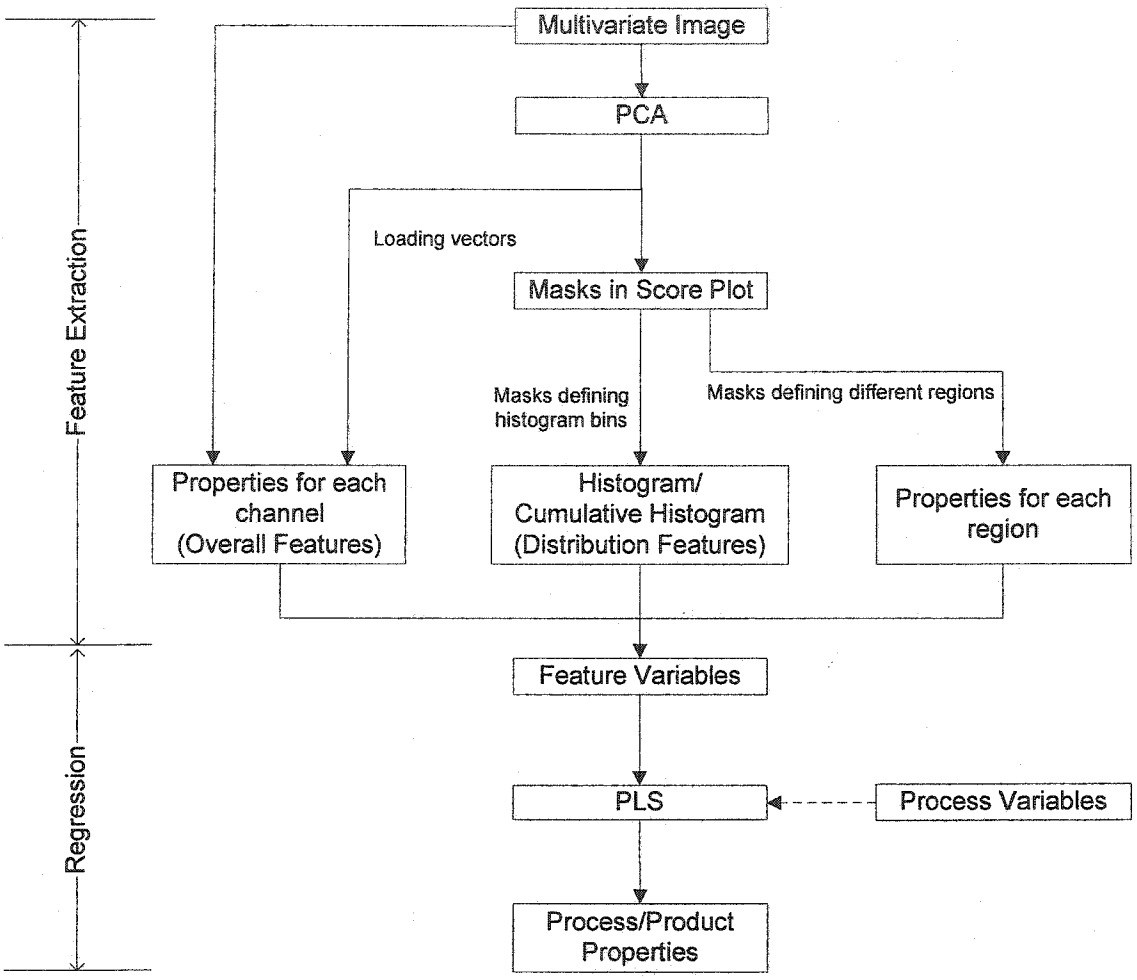



**Figure 8.1 General framework for predicting process/product properties using multivariate images**

The latter two types of feature extraction approaches are based on masks in the score plots. There are three different types of methods used to choose masks in this thesis. The first method is a trial-and-error procedure, by which the masks are decided manually through interactive mapping between the score plot and the original image space. The flame luminous region mask is chosen by this method (see section 6.4). The second method is a semi-automatic procedure. In this method, the boundaries and shape of the mask are decided manually but based on some properties of the pixels. In section 3.7, a product mask is decided based on a covariance plot and the projection of the belt image.

The third method is an automatic procedure. An example is the segmentation masks for defining histogram bins of method 6 for snack food application, which is obtained by segmenting the angle plots obtained from a covariance analysis of the score images and some response vectors.

For both applications, the multivariate images studied in this thesis are color images. However the methodologies developed are not limited to color images and can be easily extended and used for high dimensional multivariate images such as multispectral NIR images. In the case that the number of principal component is larger than 2, multidimensional masks can be obtained by studying combinations of pairs of scores. Furthermore, other dimension reduction methods mentioned in Chapter 2 can also be used depending on the individual problems and the same feature extraction methods presented using the PCA space can still be applied to these reduced spaces.

## 8.3 Future Work

Future work required for the snack food application is mainly concerned with robustness issues and with transferring a developed model to other production lines. Chapter 5 has presented a preliminary study on some robustness issues. However the work is based on a simulated case study. The methods presented for on-line model invalidation detection and on-line correction using $T_3$ score images need to be verified using more on-line data collected under variations of the vision system. Color charts can be a very good tool for calibration and/or for on-line correction, but more work need to be done on calibration methods and implementation issues.

In the flame application, we have shown that the flame images contain very useful information. However for building a successful model, especially an empirical model, large amounts of data are required and the data need to be collected under different operating conditions. If necessary designed experiments can be carried out. The feature extraction method and regression method presented in this thesis may not be adequate to handle the very large range of data from these studies. Other additional features may need

to be extracted and nonlinear PLS regression or other regression methods may be required. However, these are generally straightforward extensions. The results of this thesis show that image data should be incorporated in building industrial soft sensors for prediction of process properties, such as environmental emissions in combustion processes.

# Reference

Artyushkova, K. and J. E. Fulghum, "Multivariate Image Analysis Methods Applied to XPS Imaging Data Sets," Surface and Interface Analysis, **33**, pp.185-195, 2002

Bharati, M. H., and J. F. MacGregor, "Texture Analysis of Images using Principal Component Analysis," Proceedings of SPIE – Process Imaging for Automatic Control, Boston, MA, November 5 – 6, 2000

Bharati, M. H., J. F. MacGregor and M. Champagne, "Using Near-Infrared Multivariate Image Regression Techniques to Predict Pulp Properties," *submitted to* Tappi Journal, 2002a

Bharati, M. Multivariate Image Analysis for Process Monitoring and Control. Ph. D. thesis, 2002b

Bonnet, N., M. Herbin, and P. Vautrot, "Multivariate Image Analysis and Segmentation in Microanalysis", Scanning Microscopy, **11**, pp.1, 1997

Boudraa, A., S. M. R. Dehak, Y. Zhu, C. Pachai, Y. Bao, and J. Grimaud, "Automated Segmentation of Multiple Sclerosis Lesions in Multispectral MR Imaging Using Fuzzy Clustering," Computers in Biology and Medicine, **30**, Jan, pp. 23-40, 2000

Bright, D. S. and D. E. Newbury, "Concentration Histogram Imaging --- A Scatter Diagram Technique for Viewing Two or Three Related Images," Analytical Chemistry, **63**, Feb, pp.243A-250A, 1991

Browning, R., "New Methods for Image Collection and Analysis in Scanning Auger Microscopy", J. Vac. Sci. Technol. **A3**, pp.1959-1964,1985

Chang, Y. and J. F. Reid, "RGB Calibration for Color Image Analysis in Machine Vision," IEEE Transactions on Image Processing, **5**, Oct., pp.1414-1422, 1996

Daul, C., R. Rösch, and B. Claus, "Building a Color Classification System for Textured and Hue Homogeneous Surfaces: System Calibration and Algorithm," Machine Vision and Applications, **12**, pp.137-148, 2000

Donker N. and N. Mulder, "Analysis of MSS Digital Imagery with the Aid of Principal Component Transform," ITC Journal, **3**, pp.434-466, 1977

Duda, R. D., and P. E. Hart, "Pattern Classification and Scene Analysis," New York, John Wiley and Sons, 1973

El Gomati, M. M., D. C. Peacock, M. Protton, and C. G. Walker, "Scatter Diagrams in Energy Analysed Digital Imaging: Application to Scanning Auger Microscopy," Journal of Microscopy, **147**, Aug., pp.149-158, 1987

Esbensen, K., and P. Geladi, "Strategy of Multivariate Image Analysis (MIA)," *Chemometrics and Intelligent Laboratory Systems*, **7**, pp. 67 – 86, 1989.

Frazier, C. H., Q. Tian, A. Ozguler, S. A. Morris, and W. D. Jr. O'Brien, "High Contrast Ultrasound Images of Defects in Food Package Seals," IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, **47**, May, pp.530-539, 2000

Gonzalez, R. C. and R. E. Woods, Digital Image Processing, Addison-Wesley, 1992

Geladi, P., B. R. Kowalski, "Partial Least-Squares Regression: A Tutorial," Anal. Chim. Acta, 185, pp.1, 1986

Geladi, P., H. Isaksson, L. Lindqvist, S. Wold, and K. Esbensen, "Principal Component Analysis of Multivariate Images," Chemometrics and Intelligent Laboratory Systems, **5**, pp. 209 – 220, 1989.

Geladi, P., and H. Grahn, Multivariate Image Analysis, John Wiley & Sons, Chichester, UK, 1996

Haralick, R. M., K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Transaction on Systems, Man, and Cybernetics, **SMC3**, Nov, pp.610-621, 1973

Haralick, R. M., I. Dinstein, "A Spatial Clustering Procedure for Multi-Image Data," IEEE Transactions on Circuits and Systems, **CAS22**, May, pp.440-449, 1975

Hätönen, J., H. Hyötyniemi, J. Miettunen, and L. E. Carlsson, "Using Image Information and PLS for Predicting Mineral Concentrations in the Flotation Froth," Proceedings of he Second International Conference on Intelligent Processing and Manufacturing of Materials (IPMM'99), July 10–15, Hawaii, USA, 1999

Höskuldsson, A. "PLS Regression Methods," J. Chemometrics, **2**, pp.211, 1988

Huang, P. C., "Color Rendition Using Three-Dimensional Interpolation," Proc. SPIE: Imaging Applications in the Work World, pp.111-115, 1988

Huang, Y., Y. Yan, G. Lu, and A. Reed, "On-line Flicker Measurement of Gaseous Flames by Image Processing and Spectral Analysis," Meas. Sci. Technol, **10**, pp.726-733, 1999

Ifarraguerri, A. and C. Chang, "Unsupervised Hyperspectral Image Analysis with Projection Pursuit", IEEE Transactions on Geoscience and Remote Sensing, **38**, pp. 2529-2538, 2000

Jackowski, M., A. Goshtasby, S. Bines, D. Roseman, and C. Yu, "Correcting the Geometry and Color of Digital Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, **19**, Oct., pp.1152-1158, 1997

Jackson, J. E., A User's Guide to Principal Components, John Wiley and Sons, Inc., New York, 1991

Jia, X. and J. A. Rechards, "Segmented Principal Components Transformation for Efficient Hyperspectral Remote-Sensing Image Display and Classification," IEEE Transactions on Geoscience and Remote Sensing, 37, Jan., pp.538-542, 1999

Jimenez, L. O. and D. A. Landgrebe, "Hyperspectral Data Analysis and Supervised Feature Reduction via Projection Pursuit," IEEE Transactions on Geoscience and Remote Sensing, **31**, Nov, pp.2653-2667, 1993

Jollife, I. T., Principal Component Analysis, Springer, New York, 1986

Kaarna, A., P. Zemcik, H. Kälviäinen, and J. Parkkinen, "Compression of Multispectral Remote Sensing Images Using Clustering and Spectral Reduction," IEEE Transactions on Geoscience and Remote Sensing, **38**, 2, pp.1073-1082, 2000

Kanamori, K. and Kotera, H., "Color Correction Technique for Hard Copies by 4-Neighbors Interpolation Method," Journal of Imaging Science and Technology, **36**, pp.73-80, 1992

Kang, H. R., "Color Scanner Calibration," Journal of Imaging Science and Technology, **36**, pp.162-170, 1992

Kargacin, M. E. and B. R. Kowalski, "Ion Intensity and Image Resolution in Secondary Ion Mass Spectrometry," Analytical Chemistry, **58**, pp.2300-2306, 1986

Kartikeyan, B., K. L. Majumder, D. K. Pal, A. Sarkar, M. K. Biswas, and V. Kumar, "A MRF Model-based Segmentation Approach to Classification for Multispectral Imagery," IEEE Transactions on Geoscience and Remote Sensing, **40**, May, pp.1102-1113, 2002

Kettig, R. L. and D. A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," IEEE Transactions on GeoScience Electronics, **GE4**, Jan, pp.19-26, 1976

Kovalev, V. A., F. Kruggel, H. Gertz, and D. Y. Von Gramon, "Three-Dimensional Texture Analysis of MRI Brain Datasets," IEEE Transactions on Medical Imaging, **20**, May, 2001

Kohonen, T., Self-organization and Associative Memory, Springer, Berlin, 1984

Krijnsen, H. C., W. E. J. van Kooten, H. P. A. Calis, R. P. Verbeek, and C. M. van den Bleek, "Prediction of NOx Emissions from a Transiently Operating Diesel Engine Using an Artificial Neural Network," Chem. Eng. Technol. **22**, pp. 601-607, 1999

Kruse F. "Spectral Mapping with Landsat Thematic Mapper and Imaging Spectroscopy for Precious Metal Exploration," in Proc. Remote Sensing for Exploration Geology, Calgary, Oct., pp. 17-28, 1989

Kumar, S., J. Ghosh, M. M. Crawford, "Best-Bases Feature Extraction Algorithms for Classification of Hyperspectral Data", IEEE Transactions on Geoscience and Remote Sensing, **39**, 7, July, pp. 1368-1379, 2001

Lee, C. and D. A. Landgrebe, "Decision Boundary Feature Extraction for Nonparametric Classification," IEEE Transactions on Systems, Man, and Cybernetics, **23**, 2, Mar, pp.433-444, 1993

Liang, Z., J. R. MacFall, and D. P. Harrington, "Parameter Estimation and Tissue Segmentation from Multispectral MR Images," IEEE Transactions on Medical Imaging, **13**, Sept, pp.441-449, 1994

Lied, T. T., "Multivariate Image Regression (MIR) for Quantitative Predictions," Ph.D. Thesis, Telemark University College and the Norwegian University of Science & Technology, Porsgrunn, Norway, 2000.

Lied, T. T. and K. H. Esbensen, "Principles of MIR, Multivariate Image Regression I: Regression Typology and Representative Application Studies," Chemometrics and Intelligent Laboratory Systems, **58**, pp.213-226, 2001

Liu, J. and J. F. MacGregor, "Multiresolutional Multivariate Image Analysis and its Application to Color Texture Analysis," *submitted to* Pattern Recognition, 2002

Lu, G., Y. Yan, and D. D. Ward, "Advanced Monitoring, Characterisation and Evaluation of Gas Fired Flames in a Utility Boiler," Journal of the Institute of Energy, **73**, pp.43-49, 2000

McCauley, J. D. and B. A. Engel, "Comparison of Scene Segmentations: SMAP, ECHO, and Maximum Likelihood," IEEE Transactions on Geoscience and Remote Sensing, **33**, Nov, pp. 1313-1316, 1995

Maxwell, E. L., "Multivariate Systems Analysis of Multispectral Imagery", Photogrammetric Engineering and Remote Sensing, **42**, Sep., pp.189-210, 1976

Nason, G. P., "Three-dimensional Projection Pursuit", J. R. Statist. Soc. C, **44**, pp. 411-430, 1995

Pla, F., J. S. Sanchez, and J. M. Sanchiz, "On-line Machine Vision System for Fast Fruit Colour Sorting Using Low-cost Architecture," Proceedings of SPIE – The International Society for Optical Engineering, **3836**, pp.244-251, 1999

Qin, S. J., H. Yue, and R. Dunia, "Self-Validating Inferential Sensors with Application to Air Emission Monitoring," Ind. Eng. Chem. Res. **36**, pp.1675-1685, 1997

Raya, S. P., "Low-Level Segmentation of 3-D magnetic Resonance Brain Images --- A Rule-Based System," IEEE Transactions on Medical Imaging, **9**, Sep, 1990

Ready, P. J., P. A. Wintz, "Information Extraction, SNR Improvement, and Data Compression in Multispectral Imagery," IEEE Transactions on Communictions, **Com21**, Oct., pp.1123-1131, 1973

Recce, M., A. Plebe, J. Taylor, and G. Tropiano, "Video Grading of Oranges in Real-time," Artificial Intelligence Review, **12**, Feb, pp.117-136, 1998

Reddick, W. E., J. O. Glass, E. N. Cook, T. D. Elkin and R. J. Deaton, "Automated Segmentation and Classification of Multispectral Magnetic Resonance Images of Brain Using Artificial Neural Networks," IEEE Transactions on Medical Imaging, **16**, Dec, pp.911-918, 1997

Robert, A. S., Techniques for Image Processing and Classification in Remote Sensing, Academic Press, New York, 1983

Robert, P., D. Bertrand, M. F. Devaux and M. F. Sire, "Identification of Chemical Constituents by Multivariate Near Infrared Video Image Analysis First Investigation," Analytical Chemistry, **64**, pp.664-667, 1992

Rosenfeld, A. and A. C. Kak, Digital Picture Processing, Academic Press, New York, 1976

Rosenfeld, A. "From Image Analysis to Computer Vision: An Annotated Bibliography, 1955-1979," Computer Vision and Image Understanding **84**, pp. 298-324, 2001

Sammon, J. W., "A Nonlinear Mapping for Data Structure Analysis", IEEE Trans. Comput. **C18**, pp. 401-409, 1969

Serpico, S. B. and L. Bruzzone, "A New Search Algorithm for Feature Selection in Hyperspectral Remote Sensing Images," IEEE Transactions on Geoscience and Remote Sensing, **39**, July, pp.1360-1367, 2001

Shan, Y., M. D. Normand, and M. Peleg, "Estimation of the Surface Concentration of Adhered Particles ay Color Imaging," Powder Technology, **92**, Jul, pp.147-153, 1997

Shibayama, M. and T. Akiyama, "Estimating Grain Yield of Maturing rice Canopies Using High Spectral Resolution Reflectance Measurements," Remote Sensing Environ., **36**, pp.45-53, 1991

Shimoda, M., A. Sugano, T. Kimura, Y. Watanabe, and K. Ishiyama, "Prediction Methods of Unburnt Carbon for Coal Fired Utility Boiler Using Image Processing Technique of Combustion Flame," IEEE Transactions on Energy Conversion, 5, pp.640-645, 1990

Tsai, F. and W. D. Philpot, "A Derivative-Aided Hyperspectral Image Analysis System for Land-Cover Classification," IEEE Transactions on Geoscience and Remote Sensing, **40**, Feb, pp.416-425, 2002

Vailaya, A., A. Jain, and H. J. Zhang, "On Image Classification: City Images vs. Landscapes," Pattern Recognition, **31**, Dec, pp.1921-1935, 1998

Van Den Broek, E. P. P. A. Derks, E. W. Van De Ven, D. Wienke, P. Geladi, and L. M. C. Buydens, "Plastic Identification by Remote Sensing Spectroscopic NIR Imaging Using Kernel Partial Least Squares (KPLS)," Chemometrics and Intelligent Laboratory Systems, **35**, pp.187-197

Vittorio, A. V. D. and W. J. Emery, "An Automated, Dynamic Threshold Cloud-Masking Algorithm for Daytime AVHRR Images Over Land," IEEE Transactions on Geoscience and Remote Sensing, **40**, Aug., pp.1682-1694, 2002

Wang, F., X. J. Wang, Z. Y. Ma, J. H. Yan, Y. Chi, C. Y. Wei, M. J. Ni, and K. F. Cen, "The Research on the Estimation for the NOx Emissive Concentration of the Pulverized Coal Boiler by the Flame Image Processing Technique," Fuel, **81**, Nov, pp.2113-2120, 2002

Wold, J. P., and K. Kvaal, "Mapping Lipid Oxidation in Chicken Meat by Multispectral Imaging of Autofluorescence," Applied Spectroscopy, **54**, Jun, pp.900-909, 2000

Wold, J. P., F. Westad, and K. Heia, "Detection of Parasites in Cod Fillets by Using SIMCA Classification in Multispectral Images in the Visible and NIR Region," Applied Spectroscopy, **55**, 8, pp.1025-1034, 2001

Wulfsohn, D., Y. Sarig, and R. V. Algazi, "Defect Sorting of Dry Dates by Image Analysis," Canadian Agricultural Engineering, **35**, Arp-Jun, pp.133-139, 1993

Yamaguchi, T., K. T. V. Grattan, H. Uchiyama, and T. Yamada, "A Practical Fiber Optic Air-ratio Sensor Operating by Flame Color Detection," Rev. Sci. Instrum. **68**, pp.197-202, 1997

Yamazaki, T. and D. Gingras, "Unsupervised Multispectral Image Classification Using MRF Models and VQ Method," Transactions on Geoscience and Remote Sensing, **37**, Mar, pp.1173-1176, 1999

# Appendix A

# Overview of PCA and PLS

Principal Component Analysis (PCA) and Partial Least Squares (PLS) are two multivariate statistical methods.

PCA is a method that shows the correlation structure of a data matrix **X**, and approximates it by a matrix product of lower dimension ($\mathbf{TP}^T$), called the principal components plus a matrix of residuals (**E**). For a mean-centered matrix **X**, PCA can be expressed as

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T + \mathbf{E}$$

where **T** is the score matrix, **P** is the loading matrix, $A$ is the number of principal components, $\mathbf{t}_a$ and $\mathbf{p}_a$, $a$-th column vectors of **T** and **P**, are the score vector and loading vector of the $a$-th principal component respectively.

PCA geometrically corresponds to fitting a line, plane or hyper plane to the data in the multidimensional space with the variables as axes. The loading matrix **P** is orthonormal and can be computed from the eigenvectors of the covariance matrix $\mathbf{X}^T\mathbf{X}$. The score matrix **T** can be computed from

$$\mathbf{T} = \mathbf{XP}$$

Each score vector $\mathbf{t}_a$ is a linear combination of original variables and $\mathbf{p}_a$ is the combination weights. Score scatter plots (e.g. $t_1$ vs. $t_2$) are important tools in PCA, which can be seen as windows of the original X space. These score plots are often used to reveal

groups, trends, outliers, etc. of the observations. Loading plots display the weights relating to original variables and show the correlation structure of the variables. Other important tools in PCA including distance to model (DModx), which is the distance to the hyper plane defined by principal components of an observation, and Hotelling's $T^2$, which is a statistic computed from the scores. Both DModx and Hotelling's $T^2$ are used for checking whether the new observation has a similar correlation structure as the training data set.

PLS finds the linear relationship between a matrix **Y** (response variables) and a matrix **X** (predictor variables) and is expressed as

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}$$
$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^T + \mathbf{F}$$

where **T** is the score matrix, **P** is the loading matrix for **X**, **Q** is the loading matrix for **Y**, **E** and **F** are the residual matrices for **X** and **Y** space respectively.

Geometrically, PLS modelling consists of simultaneous projections of both the **X** and **Y** spaces onto low dimensional hyper planes. The objectives are to maximize the covariance between the projections of **X** and **Y**.

The NIPALS algorithm can be used to find the score and loading matrices. A weight matrix $\mathbf{W}^*$ is computed to obtain **T**.

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*$$

Score plot, loading (weight) plot, DModx and Hotelling's $T^2$, similar to those from PCA, can be used.

A PLS model can always be transformed into the general linear regression form:

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W}^*\mathbf{Q}^T = \mathbf{X}\hat{\mathbf{B}}, \text{ where } \hat{\mathbf{B}} = \mathbf{W}^*\mathbf{Q}^T$$

More detailed explanation and applications of using PCA and PLS can be found in the literature [Geladi et al., 1986; Jollife, 1986; Höskuldsson, 1988; Jackson, 1991].

# Appendix B

# Estimation of Lab Analysis Errors

In snack food applications, lab analysis errors for product properties (coating concentration in section 4.3.1 and organileptic properties in section 5.2.2) need to be estimated for evaluating the prediction power of the models developed. The lab analysis error considered here involves both sampling errors and instrument measurement errors. To estimate these analysis errors, several groups of data points are used. Each group contains samples taken within a short period of time whose properties can be assumed to be constant within that period of time. Therefore, points within one group can considered as replicate points.

Let us consider $K$ groups of data points and for a group $k$, the number of data points is $N_k$ ($k$=1,2,...,$K$). $y_{ki}$ is the property of the $i$-th point in group $k$ and $\bar{y}_k$ is the average property of group $k$. Then the lab analysis error (1 standard deviation) can be estimated as:

$$err = \sqrt{\frac{\sum_{k=1}^{K}\sum_{i=1}^{N_k}(y_{ki}-\bar{y}_k)^2}{\sum_{k=1}^{K}N_k - K}}$$