97/0672

J. agric. Engng Res. (1997) **68**, 115–124

Classification of a Granular Product using High-Level Fusion of Vision Features

F. Ros; S. Guillaume; V. Bellon-Maurel

CEMAGREF, 361 rue J.-F Breton, BP 5095, 34033 Montpellier Cedex 1, France

(Received 14 July 1995; accepted in revised form 18 June 1997)

The characterization of a granular product based on image analysis can be a difficult problem because it often requires the combination of a large number of features of different natures extracted from the image. This problem of classification can be solved by two approaches. One of these approaches consists of aggregating the qualitative information which is obtained by considering each individual feature, as a virtual sensor. This is a triple-step system: first, for each feature (i.e. virtual sensor), the samples are given a probability of belonging to a class (clustering); second, these probabilities are aggregated in order to give a global probability of the sample of belonging to each class (supervised neural network); third, the sample is assigned to the class which shows the maximal global probability. This procedure was applied to classify semolina samples. These were obtained by grinding wheat grains. Three classes were defined using three grinding roll gaps of 0·3, 0·4 and 0·5 mm, respectively. The average of correct classification was better than 80%. This methodology is particularly interesting because it gives a very satisfactory result and is quite versatile: new features added to the classification process require an update of one part of the procedure only.

Notation

P	perimeter of a connected set of pixels
A	area of a connected set of pixels
x, y	spatial coordinates of pixels
S_x	variance of x
S_y	variance of y
$S_{x,y}$	covariance between x and y
M^p	centred luminance moment order p
$f(i)$	number of pixels belonging to interval number i
C_{gl}	number of runs of length l and of grey level g in a given direction
M_l^r	moment of the lengths, order r.
M_g^r	moment of the greys, order r
g_m	average grey level of class g
C_{rc}	number of occurrences of transitions from class "r" to class "c" for two neighbor pixels in a given direction
g_{rc}	difference between the average grey levels of classes number "r" and "c"

1. Introduction

The qualitative characterization of granular products is acquiring increasing importance in many industries, particularly in the food industry. For accuracy, swiftness, non-contact detection and since there is no restriction on the type of products to be analysed, image analysis seems to be the most efficient method. Many authors have shown its efficiency in particle size measurement. Bertrand *et al.*[1] characterized different types of flour in bulk, Sapirstein *et al.*[2] classified cereal grains (wheat, barley, rye and oat), and Zayas *et al.*[3] identified foreign bodies in wheat product.

The problem we address concerns the classification of heterogeneous granular product populations. For this, we intend to develop a general method which is able to assign a pattern (a sample of granular products) to a predefined class of quality. Many different applications are concerned with such a procedure. For example, in the food industries, a consumer can define standard qualities of a product according to different blendings of the possible constituents. He can then characterize the different classes containing deficits or excess of some constituents. The developed classifying system then ought to recognize the given classes in order to satisfy the consumer. In semolina lines, classes of quality can also be associated with the adjustment of the parameters of a process.

Quality control is a difficult issue because both objective and subjective criteria have to be taken into account. In many industries, this is often executed by an expert who manages all the production process. He is not only

0021-8634/97/100115 + 10 $25.00/0/ag970189

guided by chemical and physical analysis but also proceeds by subjective characterization (vision, touch, sensation, etc.). An important problem stems from the expert's incapacity to clearly describe his decision making, how he characterizes the product and combines his different sensations to take the qualitative decision. This rules out the choice of relevant features for the problem, and the generation of a rule-based system. The only solution consists in building the decision system by using a learning phase, which means finding the relationships between granular product images and a given quality (provided by the expert) by considering a set of representative examples.

Two types of features can be extracted from an image of granular products. First, individual features resulting from measurements of individual granules (for instance, the size) can be represented by histograms of frequency and, second, global features corresponding to global measures such as texture. The features in use are summarized in Table 1 and have been described by Guillaume *et al.*[4] A detailed description of these features is provided in the appendix. To better manage the complexity of a granular population, we intend to use both individual and global features. Global features are vectors that can be considered as pseudo-histograms. The number of intervals of each pseudo-histogram is equal to the size of its corresponding vector. The value of each vector component is normalized in order to give a pseudo-frequency. This first step has been described by Ros *et al.*[5] Thus, all the features, including the global descriptors, can be represented as histograms having various numbers of intervals. Then, a supervised method optimizes the number and the width of distribution intervals, i.e. the size of the feature. In this procedure, the so-called "stability" and "discrimination" indices are created. Finally, a factorial analysis allows reduction of the size of each feature by creating new uncorrelated variables. Since the best classification obtained with only one

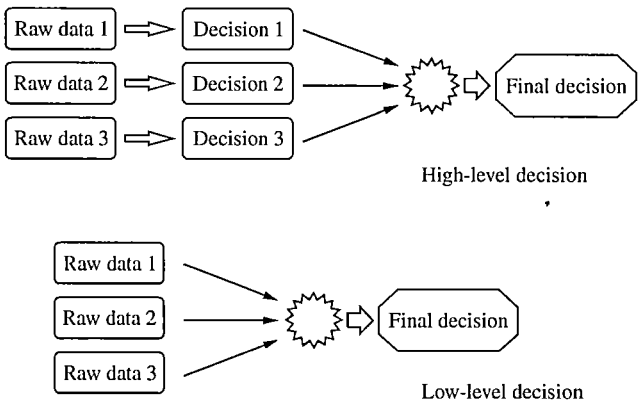

Fig. 1. High- and low-level decision-making systems

feature (convex hull area) was found by Ros *et al.*[5] to be 64%, it is suggested that information is fused to increase the performance. For this, we consider each feature as the output of an individual sensor. This corresponds to using vision as a multisensor system, in which virtual sensors are partially redundant. For instance, feature numbers 9, 13, 14, 15, and 16 (see Table 1 and the appendix 1) are shape descriptors and are likely to give similar information.

Two ways can be explored to combine information. Primary information given by the sensor (i.e. each feature) is called raw data and they are quantitative items that have been pre-processed (histogram reduction, factorial analysis). This primary information can also be individually processed (i.e. for each sensor) to give a qualitative elementary decision (one per sensor). Fusion can be applied either to raw data (low-level fusion)[6] or to elementary decisions (high-level fusion), as shown in *Fig. 1.* Low-level fusion has already been presented by Ros *et al.*[7] and its performance will be compared with that of high-level fusion. There are several classical

Table 1
Features extracted from the images (Guillaume *et al.*[4])

Variable		*Variable*	
1	Object area	2	Object perimeter
3	Convex hull area	4	Convex hull perimeter
5	var 1/var 3	6	var 2/var 4
7	$(\text{var } 2)^2/\text{var } 1$	8	$(\text{var } 4)^2/\text{var } 3$
9	Aspect ratio	10	Centred luminance moment order 1
11	Centred luminance moment order 2	12	Centred luminance moment order 3
13	Length along largest variance axis/$\sqrt{\text{var } 1}$	14	Width along second variance axis/$\sqrt{\text{var } 1}$
15	Length along largest variance axis/var 2	16	Width along second variance axis/var 2
17	Run length parameters	18	Run length histogram features
19	Co-occurrence parameters	20	Co-occurrence histogram features
21	Opening grey-level distribution		

methods of high-level fusion such as vote and weighted vote.[8] This paper presents another method, namely, the supervised training of a classification system in which inputs are the elementary decisions.

2. Procedures

2.1. *Acquisition of experimental data*

The building of the experimental set has already been described by Guillaume *et al.*[4] The granular product comes from the wheat milling process. The breaking of grains is carried out by rolls. The resulting product quality, including the size distribution, depends on the grain variety, the grain size and the gap between the rolls. Three quality classes have been defined by three roll gap settings of 0·30, 0·40, and 0·50 mm, respectively. Images are obtained from a derivative of the particle flow as shown in *Fig. 2.* The derivative flow passes through a regulator designed to distribute the product over the whole width of the slot without sorting the particles according to their size. A CCD (charge coupled device) camera synchronized with a stroboscope allows the falling products to be "captured" in flight. Control and data storage are done by a personal computer. For each class, 350 images were stored, each of them made up of about 120 particles. The background of the scene is black (the value of each pixel is zero), while the product pixel value ranges from 1 to 255. *Figure 3* shows a picture of the milled product, first in grey tones, and then in a binary representation (each pixel whose value is different from zero is set to 255). The features are then pre-processed according to Ros *et al.*[5] in order to reduce the data space without losing any important information.

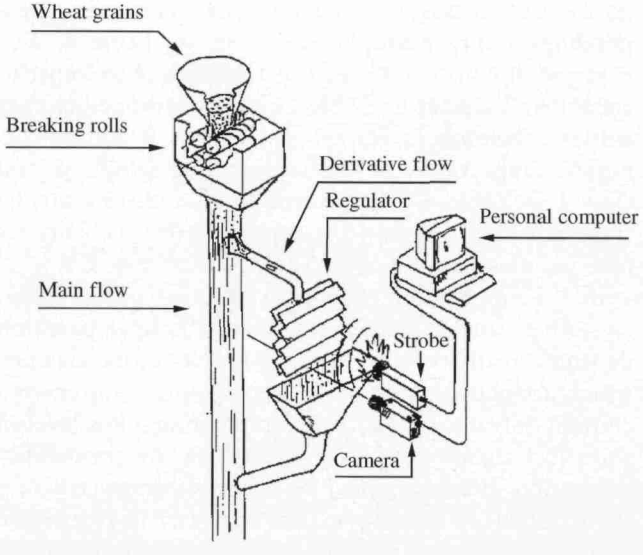

Fig. 2. Scheme of the experimental system for producing semolina samples

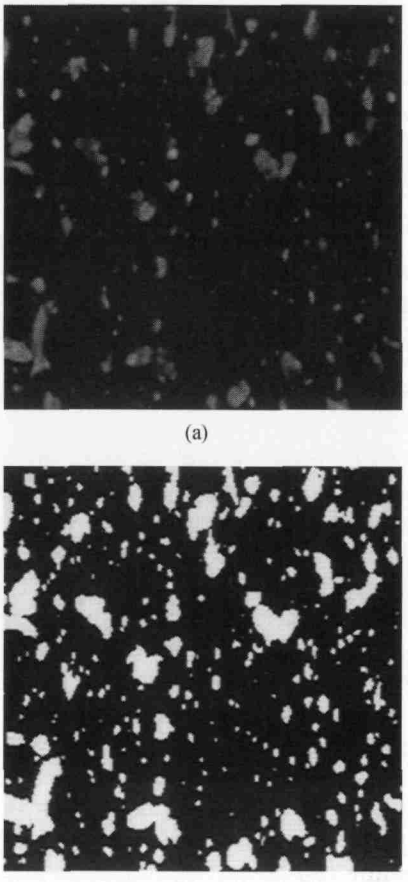

Fig. 3. (a) Grey tones and (b) binary images of granular product

The data set is made up of:

- k quality classes; here $k = 3$ according to the three roll gaps,
- p features; here $p = 21$ (see Table 1),
- n samples in the training set; here $n = 90$, that means 30 for each class.

2.2. *The fusion process*

The fusion system follows a strategy of step-by-step reduction of the information collected. This strategy is summarized here and will then be described in detail. First, the raw data are transformed into elementary decisions: this step is achieved by an unsupervised method (clustering); the result is the probability for each sample, according to each feature, of belonging to each class.

These results are represented in a matrix k (number of classes) of p columns (number of features) and n rows (number of samples); cell $Mjxy$ corresponds to the probability of sample y, according to feature x of belonging to class j. The information is represented by $n \times k \times p$ elements. Second, the elementary decisions are fused within each class: the result is a probability for each sample of belonging to each class, according to all the features. The information is now represented by $n \times k$ elements. Third, the final decision, each sample is assigned to one class; thus the information is contained in n elements.

2.2.1. *Transformation of raw data into elementary decisions*

A method of unsupervised clustering such as hierarchical clustering is used to cluster the sample collection. Let c be the number of clusters which are created by the clustering procedure. A given sample belongs to one of c possible clusters. Let k be the number of known qualitative classes of particles. For each cluster C_i, it is easy to assess the observed conditional probabilities p_{ij} of belonging to a given qualitative class K_j. This probability is computed by n_{ij}/n_i, with n_{ij} the number of samples of the training set belonging both to C_i and K_j, and n_i the total number of samples belonging to the cluster C_i. For each sample belonging to a cluster, this probability is equal to the probability of the cluster of belonging to each class. This procedure can be applied independently for all the features. Therefore, the $n \times k \times p$ $Mjxy$ cells are filled: each $Mjxy$ cell is the probability for sample y, according to a feature x, of belonging to class j. These probabilities can also be represented by k matrices of $n \times p$ cells.

As an example, let us consider ten samples, numbered from A to J. These samples are classified in three classes according to two features. The actual classes the samples belongs to are:

Class 1: Samples A, B, C.

Class 2: Samples D, E, F.

Class 3: Samples G, H, I, J.

For each feature, these samples, submitted to the clustering, are distributed in four different clusters as shown in Table 2. Let us compute, for each feature, the probabilities of the clusters of belonging to one class. For instance, let us consider the first column of the matrix dealing with feature 1. The probability of cluster I belonging to class 1 is 2/4, i.e. 1/2 because 2 is the number of samples of cluster I belonging to class 1, i.e. B and C, and 4 is the number of examples of cluster I. The probability of cluster II belonging to class 1 is 0, because there is no sample belonging to class 1 in cluster II. The probability of cluster III belonging to class 1 is $1/1 = 1$, because A is the only sample in cluster III and belongs to class 1. Continuing in this way, two (one per feature) 4×3 matrices of probabilities can be created (as in Table 3).

Table 2
Assignment, feature per feature, of each sample to one cluster

Feature 1		*Feature 2*	
Cluster	*Sample*	*Cluster*	*Sample*
I	B, C, F, H	I	B, D, F, H
II	D, I	II	C, I
III	A	III	A, J
IV	E, G, J,	IV	E, G

Table 3
Probability matrix for each cluster of belonging to one class, for features 1 and 2, respectively

	Class		
Cluster	*1*	*2*	*3*
		Feature 1	
I	1/2	1/4	1/4
II	0	1/2	1/2
III	1	0	0
IV	0	1/3	2/3
		Feature 2	
I	1/4	1/2	1/4
II	1/2	0	1/2
III	1/2	0	1/2
IV	0	1/2	1/2

Then, two 10×3 matrices can be created, by assigning to each sample the probability, given by the cluster composition, of the class to which it belongs. The ones corresponding to the example are given in Table 4. For example, if considering feature 1, sample A belongs to class 1 and cluster 3 (Table 2), so the probability that cluster 3 belongs to class 1 is 1 (Table 3) and so this number appears in the cell representing sample A and class 1 in Table 4. These three 10×3 matrices can be transformed into three 10 (samples) $\times$ 2 (features) matrix (one per class) as shown in Table 5.

In this application, two drawbacks of unsupervised clustering can be highlighted. First, the final partition depends greatly on the centres of gravity of the clusters, which are chosen randomly. Second, some clusters can contain only few elements thus presenting a low level of statistical significance. Both may bias the probability estimation. However, this bias is partly compensated by the redundancy of features. The best way to reduce the influence of these errors is to supervise the fusion of elementary decisions. This can be achieved by training a neural network.

Table 4
For each feature, probability of each sample belonging to a class

	Class		
Sample	1	2	3
		Feature 1	
A	1	0	0
B	1/2	1/4	1/4
C	1/2	1/4	1/4
D	0	1/2	1/2
E	0	1/3	2/3
F	1/2	1/4	1/4
G	0	1/3	2/3
H	1/2	1/4	1/4
I	0	1/2	1/2
J	0	1/3	2/3
		Feature 2	
A	1/2	0	1/2
B	1/4	1/2	1/4
C	1/2	0	1/2
D	1/4	1/2	1/4
E	0	1/2	1/2
F	1/4	1/2	1/4
G	0	1/2	1/2
H	1/4	1/2	1/4
I	1/2	0	1/2
J	1/2	0	1/2

2.2.2. *Fusion of elementary decisions within each class*

In order to reduce the size $(n \times p)$ of the matrix k to a $n \times 1$ matrix, it is necessary to aggregate the information. This is done through a supervised training phase using neural networks. For each different class a neural network classifier is created. Each has as many inputs as features, i.e. two inputs in the above example and 21 in the milled product application (see Table 1). To train the neural network, the desired outputs are assigned a 1 if the sample really belongs to the class concerned and a 0 to the contrary. Thus, for each class, each sample is given a score (the neural network output) which is similar to a global probability of belonging to this class.

Table 5
For each class, probability of each sample of belonging to it, considering each feature

Class 1			*Class 2*			*Class 3*		
	Feature			*Feature*			*Feature*	
Sample	*1*	*2*	*Sample*	*1*	*2*	*Sample*	*1*	*2*
A	1	1/2	A	0	0	A	0	1/2
B	1/2	1/4	B	1/4	1/2	B	1/4	1/4
C	1/2	1/2	C	1/4	0	C	1/4	1/2
D	0	1/4	D	1/2	1/2	D	1/2	1/4
E	0	0	E	1/3	1/2	E	2/3	1/2
F	1/2	1/4	F	1/4	1/2	F	1/4	1/4
G	0	0	G	1/3	1/2	G	2/3	1/2
H	1/2	1/4	H	1/4	1/2	H	1/4	1/4
I	0	1/2	I	1/2	0	I	1/2	1/2
J	0	1/2	J	1/3	0	J	2/3	1/2

2.2.3. *Final decision*

The final decision consists of assigning each sample to one class, i.e. to switch from an $n \times k$ matrix to an n vector. This aggregation is made by assigning each sample to the class for which the highest score (i.e. neural network output) is seen.

3. Results and discussion

3.1 *Partition of the feature space into clusters*

For each feature, a hierarchical clustering analysis has been applied. At the beginning, there are as many clusters as the number of samples, each cluster containing one sample. At each step two clusters are merged in order to minimize the sum of the within-group variances of all the clusters. The procedure is done until all the samples are gathered in an unique cluster. At this step, the variance is maximal. At each step each cluster can be characterized by the proportion of the population it contains and by its variance, expressed in percentage of the maximum variance. During the process, both the number of elements per cluster and the variance (i.e. their heterogeneity) increase globally. The final partition must be made of clusters which contain enough elements to be statistically valid and which remain sufficiently homogeneous (low variance). To find this compromise (i.e. the point where the procedure must be stopped), we use the function of the global variance versus the steps: the points of slope break are the suitable ones. For instance, for feature 1 (*Fig. 4*), the shape of the variance function enables us to

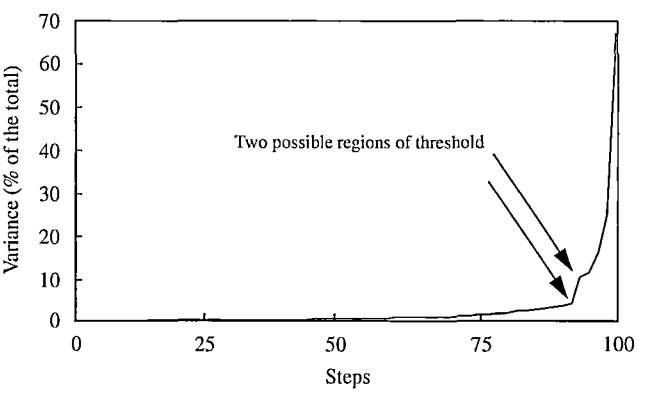

Fig. 4. Variance function shape for feature 1

find the possible threshold at around 10% of the maximum variance. This threshold occurs at the step 86. Because feature-1 histogram is made up of 100 intervals, this means that the population is partitioned into 14 groups as shown at the bottom of the tree of *Fig. 5*. On this tree, each node is associated with a cluster. The first figure in parenthesis corresponds to the percentage of the maximum variance and the second figure to the percentage of the population contained in the cluster. The tree does not represent the whole process, but only the final steps (after the threshold value has been reached). The process is ascending along the axis "Steps", the last merging is at the top of the figure. A switch from 14 to 13 groups is made by artificially fusing the clusters k and 1 because both are made up of elements belonging to class 3 (gap of 0·50 mm), and they are fused in a further step of the clustering procedure. In Table 6 the clusters are sorted following their number of elements. For each cluster, the number of elements of each class are counted and the probability of belonging to each class is computed.

The same procedure is applied to each feature and the number of clusters for each of the 21 features is reported in Table 7.

3.2. *Classification*

Three neural networks have been designed, one for each decision class. Each has been trained on a subset of samples, called the training set, and tested on the complementary subset, called the test set. The number of samples of the training and the test sets are, respectively, 30 and 20 for each class. The input layer is made up of 21 cells, one for each feature and the input data is the probability of belonging to the considered class according to this feature. It should be noted that only three cells were necessary in the hidden layer to find the

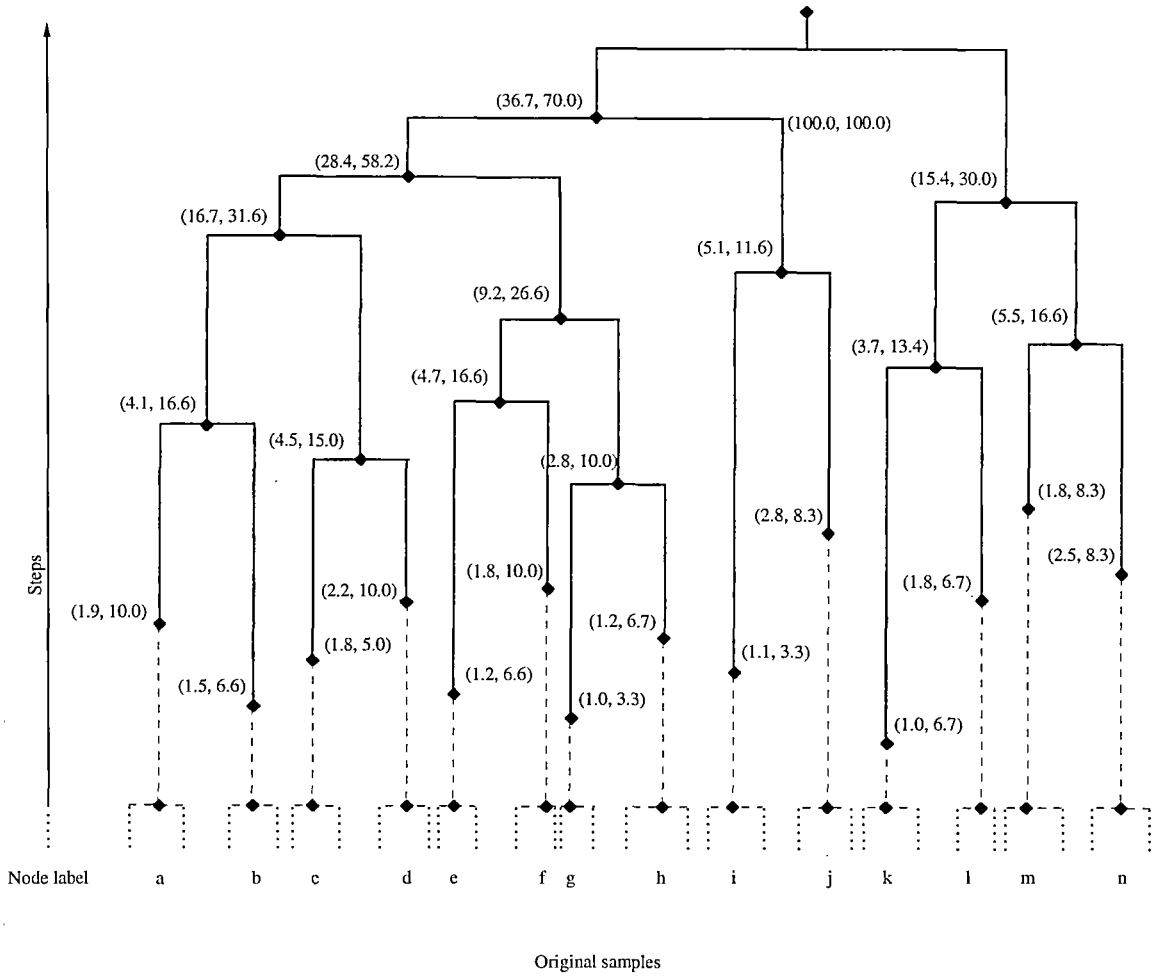

Fig. 5. Final steps of the hierarchical clustering for feature 1

Table 6
Probability matrix for each cluster of belonging to one class, for feature 1

Cluster number	*Population (%)*	*Node label (Fig. 5)*	*Class 1*	*Class 2*	*Class 3*
1	10·0	a	0·1667	0·1667	0·6667
2	10·0	d	0·0000	0·5000	0·5000
3	10·0	f	0·3333	0·3333	0·3333
4	8·3	j	0·1667	0·6667	0·1667
5	8·3	m	0·7500	0·2500	0·0000
6	8·3	n	0·1667	0·5000	0·3333
7	6·7	h	0·0000	0·3333	0·6667
8	6·7 + 6·7	k + 1	0·0000	0·0000	1·0000
9	6·6	b	0·0000	0·2000	0·8000
10	6·6	e	1·0000	0·0000	0·0000
11	5·0	c	1·0000	0·0000	0·0000
12	3·3	g	0·0000	1·0000	0·0000
13	3·3	i	1·0000	0·0000	0·0000

relationships between the elementary decisions and the desired output for class 1. Four cells were however necessary for the other classes and a longer computation time to obtain robust classifiers. This shows that class 2 and class 3 are close and partially overlap. A similar conclusion has been made when a low-level approach was applied.[7]

Table 8 shows the results of the final classification. For each sample, we consider the outputs of each of the three neural networks, each neural network being attached to a class. The sample is assigned to the class for which the neural network shows the maximum output. The average of good classification over the three classes in better than 80%. As expected, the results are best for class 1.

3.3. *Comparison of high-level and low-level fusion*

Our former approach[7] used low-level fusion and combined statistical methods, such as factorial analysis and information selection, to find the most relevant components with three techniques of decision making: *k* nearest neighbours,[9] discriminant analysis and multilayer neural networks[10,11] (MNN). The results were close to the ones obtained in this paper: 100% of good results in classification for class 1 using the three techniques (compared with 93% for high-level fusion), 81% for class 2 using discriminant analysis and MNN (compared with 70% for high-level fusion), and 76% for class 3 using MNN (compared with 82% for high-level fusion).

In the low-level approach, the construction of such a decision-making system is time-consuming and not versatile: adding new features(s) leads to starting the whole procedure again because a common space for all the features is required from the beginning. In the high-level approach, there is no need to start the whole process again: the only additional work is to build clusters for the new features and to proceed to the aggregation phase.

The important number of features (mainly in the case of functional features) is not a problem in low-level fusion. If a vote procedure is used in high-level fusion, the number of features and their partial redundancy may bias the decision. For instance, if three features among four are highly correlated, the decision of the system will never take into account the information of the fourth one: the vote will always be won by the group of the three correlated ones. Using a supervised method such as neural networks in a high-level approach, enables this drawback to be avoided and increases the system robustness; this means that if a sensor gives wrong information, this information is compensated by the data of the other sensors.

4. Conclusions

The efficiency of high-level multi-variable fusion has been tested within the framework of a classification problem of granular products in the food industry. Three

Table 7
Number of clusters for each extracted feature

Feature	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Clusters	13	15	12	16	13	16	17	13	16	17	14	10	8	9	10	10	15	13	11	12	13

Table 8
Results in classification for the training and test sets (% of good classification)

	Class 1	*Class 2*	*Class 3*
Training set	95·0	75·0	80·0
Test set	92·8	70·0	82·3

classes of a milled product were obtained by adjusting the grinding roll gaps at three different values.

Three phases are identified in solving this classification problem. The first phase deals with assigning samples to clusters according to each feature. Clustering can be done automatically (non-supervised phase) and leads, for each feature, to a probability for the sample of belonging to the different classes. In the second phase, these probabilities are aggregated by neural networks. There are as many networks as classes, three in this example. The inputs of the neural networks, correspond to the number of features and are the probabilities for the sample of belonging to the considered class. The output of each neural network is a membership index which represents the likelihood for the sample of belonging to each class. This step reinforces robustness and brings non-linearity, which is often met in human decision-making. The third phase, final classification, consists in choosing the class for which neural network output is maximum.

The results obtained by this method are as good as the ones obtained by low-level data fusion. With this method, the samples of class 1 were classified with a performance of 93%, the samples in class 2 with 70% and the samples in class 3 with 82%.

Compared with low-level fusion where the whole procedure must be started again when a feature is added, this model can easily be adapted: if a new feature is added, the only additional work is to proceed to the clustering using this feature and to start training the neural network again. The adopted approach is quite easy to perform and it can also be used for other types of problems where different pieces of information are to be aggregated. Thus, it seems to offer wide use in classification problems in conjunction with image analysis.

References

[1] **Bertrand D; Robert P; Melcion J P; Sire A** Characterisation of powders by video image analysis. Powder Technology, 1991, **66**, 171–176

[2] **Sapirstein H D; Neuman M; Wright E H; Shwedyk E; Bushuk W** An instrumental system for cereal grain classification using digital image analysis. Journal of Cereal Science, 1987, **6**, 3–14

[3] **Zayas I; Pomeranz Y; Lai F S** Discrimination of wheat and nonwheat components in grain samples by image analysis. Cereal Chemistry, 1989, **66**(3), 233–237

[4] **Guillaume S; Ros F; Chaurand M; Bellon-Maurel V; Abecassis J** Characterization of mill products by analysis of in-flow-digitalized images. Journal of Food Engineering, 1996, **27**, 311–322

[5] **Ros F; Guillaume S; Rabatel G; Bellon-Maurel V** Building and preprocessing of image data using indexes of representativeness and classification applied to granular product characterization. Journal of Chemometrics, 1997 (in press)

[6] **Hall L D** Mathematical Techniques in Multisensor Data Fusion. Boston: Artech House, 1992

[7] **Ros F; Guillaume S; Rabatel G; Sevila F** Combining global and individual image features to characterize granular products populations. Journal of Chemometrics, 1997 (in press)

[8] **Steinmetz V; Crochon M; Bellon-Maurel V; Garcia Fernandez J L; Barreiro Elorza P; Verstreken L** Sensors for fruit firmness assessment: comparison and fusion. Journal of Agricultural Engineering Research, 1996, **64**(1), 15–28

[9] **Cover T; Hart P** Nearest-neighbor pattern classification. IEEE Trans Information Theory, 1967, **IT-13**, 21–27

[10] **Lippmann R L** An introduction to computing with neural nets IEEE ASSP Magazine, 1987, **4**, 4–22

[11] **Ros F; Brons A; Sevila F; Rabatel G; Touzet C** Combination of neural network and statistical methods for sensory evaluation of biological products: On line beauty selection of flowers. Proceeding of IWANN, 1993, Pp. 726–731

[12] **Galloway M** Texture analysis using grey level run lengths. Computer Graphics and Image Processing, **4**, 171–179

[13] **Haralick R M** Statistical and Structural Approaches to Texture. Proc. IEEE, **67**, 786–804

[14] **Serra J** Image Analysis and Mathematical Morphology. London: Academic Press, 1982

Appendix

Description of the image features

The individual measurements used (Table 1) are: object area (feature 1) and perimeter (feature 2), convex hull area (feature 3) and perimeter (feature 4), aspect ratio (feature 9) and centred luminance moments of orders 1, 2 and 3 (features 10, 11 and 12 respectively).

The convex hull of an object (features 3 and 4) is defined as the surface which is found inside the lines tangent to the object, as shown in *Fig. 6.*

The combination of these primary variables leads to dimensionless variables which are not dependent on the optical system geometry: object area/convex hull area (feature 5), object perimeter/convex hull perimeter (feature 6).

The circularity is one of the most commonly used shape indices: it is defined as $P^2/4\pi A$ where P stands for perimeter and A for area; a disc has a value of unity. For both the object (feature 7) and its convex hull (feature 8), a similar index has been computed without using the coefficient of 4π.

The aspect ratio (feature 9) is the ratio of the smaller axis of the object to the larger one. It ranges between 0 and 1 (for the

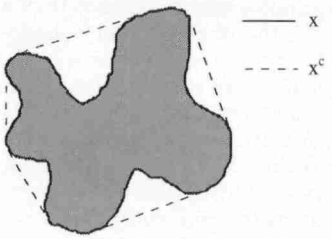

Fig. 6. x^c is the convex hull of x

circle). It is equal to the ratio of the eigenvalues of the inertia matrix (or covariance matrix):

$$\mathbf{In} = \left\langle \begin{matrix} \mathbf{S}_x & \mathbf{S}_{x,y} \\ \mathbf{S}_{x,y} & \mathbf{S}_y \end{matrix} \right\rangle$$

where x, y are the spatial coordinates of the pixels, S_x, S_y and $S_{x,y}$ are computed as follows:

$$S_x = \frac{\sum_n (x - \bar{x})^2}{n};\ S_y = \frac{\sum_n (y - \bar{y})^2}{n}$$

$$S_{x,y} = \frac{\sum_n (x - \bar{x})(y - \bar{y})}{n}$$

Let M^p be the centred luminance moments of order p, $p = 1,2,3$ (features 10–12). The grey-level distribution of the object is used to compute the moments. The distribution is made up of 16 grey-level intervals ranging from 0 to 255, so that the limits of the interval number i are $[16\,(i-1);\ 16i[$. Let $f(i)$ be the number of pixels belonging to interval i and $\bar{i}$ the average interval (empty ones are not taken into account):

$$M^p = \frac{\sum_i [f(i)\,(i - \bar{i})^p]}{[\sum f(i)]^p}$$

Features 13–16 are dimensionless combinations of the length along the two variance axes combined with the area and the perimeter of the object.

The global features are mainly textural ones. Texture is not formally defined. The term is generally understood as the visual or tactile characteristic of a surface. Two techniques were employed. The grey levels were grouped into four classes and the class limits were computed dynamically for each image to create evenly populated classes. The background (level 0) represents a particular class which is not taken into account in the statistics. As the products were not oriented the scene was analysed in three directions (east, north east, north).

The first technique is the constant grey-level run length.[12] This consists of counting in a given direction the number of length "l" runs with a grey level "g". Eleven length classes are computed, the first ten correspond to lengths of 1 to 10 pixels and the last class corresponds to all the lengths greater than 10 pixels. For each analysed direction a matrix is obtained. The number of columns is equal to the number of grey-level classes (five in this case, i.e. four useful classes and a background) and each row represents a length. Therefore, the value C_{gl} indicates the number of length "l" runs having a grey-level "g". The parameters proposed in the literature[12] are as follows.

Length level uniformity: let M_l^r be the moment of the lengths, order r, l the class number of the lengths: $M_l^r = \frac{\sum_l \sum_g C_{gl}(l)^r}{\sum_g \sum_l C_{gl}}$ By definition $U_{\text{length}} = \frac{M_l^4}{(M_l^2)^2}$.	Grey-level uniformity: let M_g^r be the moment of the greys, order r, g_m the average grey level of the class $M_g^r = \frac{\sum_g \sum_l C_{gl}(g_m)^r}{\sum_g \sum_l C_{gl}}$. By definition $U_{\text{grey}} = \frac{M_g^4}{(M_g^2)^2}$.
Short run indicator: $I_c = \frac{\sum_l \sum_g C_{gl}/(l)^2}{\sum_g \sum_l C_{gl}}$	Long run indicator: $I_l = \frac{\sum_l \sum_g C_{gl}(l)^2}{\sum_g \sum_l C_{gl}}$

This set of four variables added to the ratio I_c/I_l are computed for the three directions and gathered in a variable of size 15 (feature 17).

The normalized histogram is processed as a variable (feature 18) in order to store as much information as possible. It is a vector of size 132. For three directions there are four grey-level classes and for each of them, 11 length classes.

In the second technique, i.e. the grey-level spatial interdependence,[13] the number of transitions from one pixel of grey level "r" to the next one of grey level "c" is computed in each analysed direction. The distance between two neighbouring pixels is one. The result is a square matrix, called the co-occurrence matrix, whose size is equal to the number of grey-level classes (five in this case). The statistical computations do not take into account the case corresponding to the background–background transitions. C_{rc} represents the number of transitions from class number "r" (row) to class number "c" (column) and g_{rc} the difference between the average grey levels of classes "r" and "c". From the matrix the maximum of probability can be extracted (maximum of C_{rc}) and the following terms can be computed.

$Inertia = \frac{\sum (C_{rc}(r-c)^2)}{\sum C_{rc}}$	$Heterogeneity = \frac{\sum (C_{rc})^2}{(\sum C_{rc})^2}$
$Entropy = \frac{\sum (C_{rc} \ln(C_{rc}))}{\sum C_{rc} \ln(\sum C_{rc})}$	$Contrast = \frac{\sum (C_{rc}(g_{rc})^2)}{\sum C_{rc}}$

As for the constant grey-level run length, the variables of the three directions are gathered in a vector of size 15 (feature 19). To preserve the integrity of the information, the co-occurrence matrix is stored in the form of a variable whose size is 72, i.e. 24 for each direction (feature 20).

In order to describe the opening grey-level distribution (feature 21), it is necessary to introduce the mathematical morphology,[14] and particularly the two basic operations: erosion and

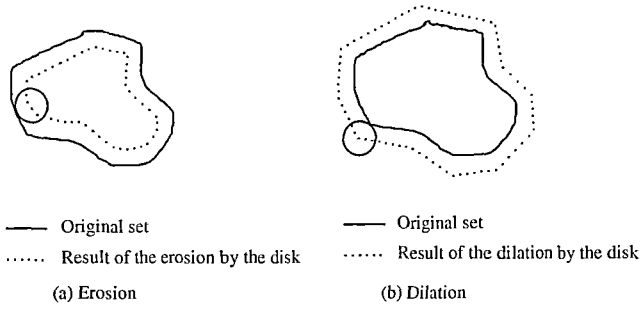

Fig. 7. Erosion and dilation of a particle

dilation (*Fig. 7*). Erosion is carried out by removing from the border of an object a band of a given thickness. The thickness depends on the size and the shape of the pattern used. The pattern is called the structuring element. Particles smaller than the structuring element are removed by erosion. Mathematically speaking, the eroded set Y of X by a pattern B is defined as the locus of all the points x belonging to X, for which B centred on x is included in X. In this application the set B was a square with a variable size. Dilation is the dual operation: it involves adding to the object border a band of a thickness corresponding to the size and the shape of the structuring element. Holes smaller then the structuring element are filled by dilation. From a mathematical point of view, the dilated set Y of X by a set B is the locus of all the points x for which B, centred on x, hits the set X. These techniques have been developed for binary sets but are now extended to numerical ones. Erosion means giving the minimum value observed in the area covered by the structuring element to a reference pixel. The maximum value is used for dilation, which allows erosion and dilation to be applied to the image without isolating or labelling each particle. The opening is made up of an erosion phase followed by a dilation phase. *Figure 8* shows an example of opening using a circle.

To obtain a size distribution of the particles in the image, it is possible to apply openings with an ever increasing pattern size. At each step the total area of the remaining particles is computed. The area decreases when the size of the square increases, and finishes by having a value of zero when the structuring element is bigger than the largest particle. In the application, ten steps of opening were carried out and the corresponding remaining areas were normalized by the original value, i.e. the areas have been divided by the total area computed before erosion. These ten values make up the last feature.

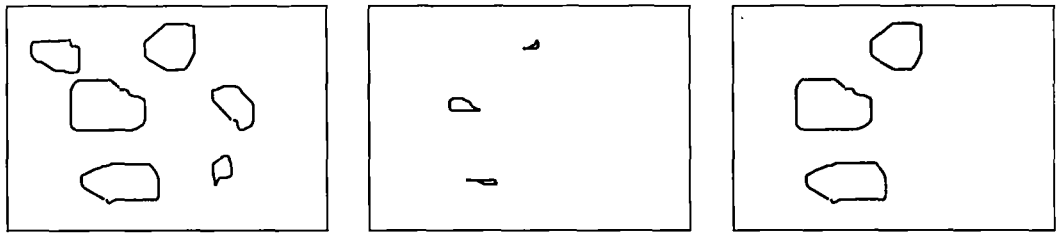

Fig. 8. Result of the opening of an image