Proceedings of the 2006 IEEE International Conference on Robotics and Automation
Orlando, Florida - May 2006

Design of Classifier to Automate the Evaluation of Protein Crystallization States

Kanako Saitoh
Saitama University
255 Shimo-okubo, Sakura, Saitama Saitama, 338-0825, Japan
Email:kana@me.ics.saitama-u.ac.jp

Kuniaki Kawabata
RIKEN
2-1 Hirosawa, Wako, Saitama, 351-0198, Japan
Email:kuniakik@riken.jp

Hajime Asama
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8568, Japan
Email:asama@race.u-tokyo.ac.jp

Taketoshi Mishima
Saitama University
255 Shimo-okubo, Sakura, Saitama Saitama, 338-0825, Japan
Email:mishima@me.ics.saitama-u.ac.jp

Mitsuaki Sugahara
RIKEN
1-1-1 Kouto, Mikazuki-cho, Sayo-gun, Hyogo, 679-5148, Japan
Email:sugah@spring8.or.jp

Abstract – This paper presents a method for designing classifier to automate an evaluation process of protein crystallization growth states. The classifier is designed by binary decision tree and its classification sequence is decided in descending order of classification accuracy. Furthermore, for applying to automated crystallization system, some functions reducing mis-classification are appended to the classifier. The performance of this method has been evaluated by comparing classification result by using proposed method with the result by manual classification.

Index Terms – classification; protein crystallization; crystallization growth states; decision tree; support vector machine

I. INTRODUCTION

In life science field, high-throughput protein structure determination by the X-ray crystal structure analysis has been advanced. For realizing the high-throughput protein structure determination, automation of crystallization process is indispensable. Therefore, there have been several attempts to automate the crystallization process [1] [2]. Through these attempts, the efficiency of crystallization work has been improved. However, such systems have mainly realized the automation of setup and storage of the crystallization droplets and the observation process has not yet been completely automated.

In the observation process, the crystallization droplets are observed and scored according to their growth states. The accumulated data is very valuable because such data is used as reference of initial condition at the future crystallization experiments. Therefore in the process, not only detecting crystals but also observing and evaluating the crystallization growth states from start to finish are very important. However it is not easy to automate the evaluation process because there is no definition of the detailed criteria for the evaluation and it is unpredictable how the crystallization progress.

In the relevant previous works, various methods for evaluating crystallization growth states have been proposed. Spraggon et al. used texture analysis and a self-organizing neural network to categorize individual crystal trials into six classes [3]. Bern et al. used Hough transform and curve tracking and classified into five classes [4]. Saitoh et al. used texture analysis and multiple linear discriminant analysis and classified into five classes [5]. All these methods can distinguish only some of classes with high probability, but it is not to say that they can distinguish all of classes. Therefore the evaluation results obtaining by applying these methods are not reliable.

The crystallization growth states have a variety of states and they are really complicated. Feature values that express crystallization states completely have not been revealed. For this reason the previous studies extracted different features respectively, but enough accuracy that can realize full automatic operation was not obtained. Our goal is to divide the classes to limit that keeps high reliability and construct the classifier that assists current manual observation work. In the method presented here, classification methodology is decided in descending order of classification accuracy. Furthermore the classes in which low accuracy is measured are grouped and the original classification is redefined. And in order to deal with vague classification boundary we attempt to introduce a new "gray-zone" class.

This paper consists of six sections. Section 2 introduces the data using for classification of protein crystallization growth states. Section 3 presents a classification method. Section 4 describes the examination and Section 5 discusses the result. And finally, Section 6 concludes this paper.

II. CLASSIFICATION SUBJECT

A. Protein crystallization image

Protein crystallization images vary depending on the crystallization techniques and imaging devices. The images used in this paper are acquired by the crystallization system "TERA" developed by RIKEN. The system employs the microbatch method as the crystallization technique and takes images of each drop using an inverted microscope with a cooling CCD camera. Fig. 1 shows some example images taken with "TERA". The image size is 1392 x 1040 pixels and the pixel size is 4.65 x 4.65 μm.

B. *Classification*

The growth states of protein crystallization take various states; for example, clear, precipitate, amorphous agglutinate and crystals with varied shape and combinations of these. For these variety states, the previous studies defined classification into from three to six classes and the defined number of classes is respectively different because requirements are different independently. In this paper, five classifications that are similar to reference [5] are used (Fig. 2). Brief descriptions of each category (: 0, 1, 2, 3, 4) are as follows.

0, Clear drop

1, Precipitate (i): creamy and grainless precipitate

2, Precipitate (ii): fine or granulated sugar-like precipitate

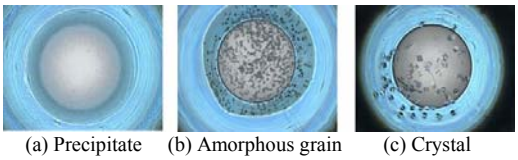
3, Precipitate (iii): amorphous state (whether it will crystallize in the future or not is not known)

4, Amorphous circular grain and crystals: (The precipitated objects have some shape)

The number of utilized images totaled 872 and in each category is class0: 102, class1: 116, class2: 78, class 3: 88 and class 4: 488. The numbers of images that belong to each class are not even because all images that were able to be acquired are used in this paper.

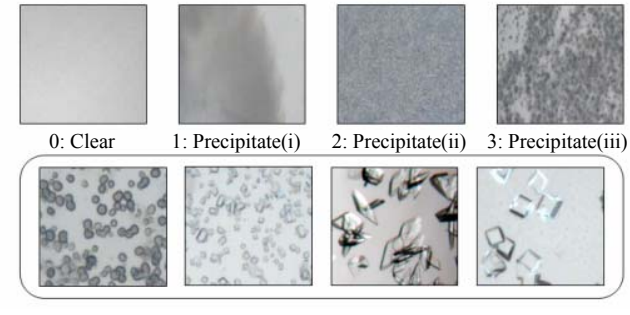
C. *Feature extraction*

For computational image analysis, texture analysis, smooth, sharpen and edge detect/enhance are often utilized to quantify images. In the method presented here, we decided to introduce texture analysis [6] to extract features from the images. Spraggon et al. and Saitoh et al. have already used texture analysis to extract feature values from crystallization images [3] [5]. The method calculates global features. But it is necessary to evaluate a local part in the image because proteins crystallize locally in a droplet. This problem is solved by dividing the whole original image into some small areas as done in [5].



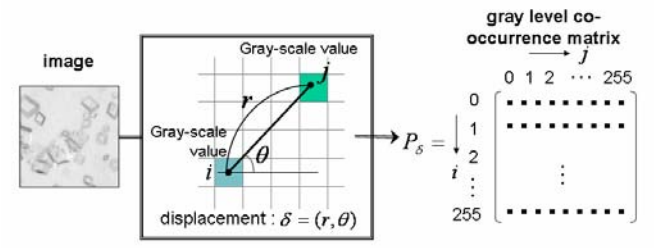
(a) Precipitate (b) Amorphous grain (c) Crystal

Fig. 1 Examples of drop images taken with the TERA system.



4: Amorphous circular grain and crystals

Fig. 2 Five classification for evaluation



(a) (b)

Fig. 3 Algorithm of deriving gray level co-occurrence matrix.

Texture analysis calculates by using a gray-level co-occurrence matrix. Fig. 3 shows the algorithm for deriving the matrix. Each element of the matrix $P_\delta(i, j)$ in Fig. 3 (b) expresses the probability that is at once gray-scale value of one pixel is i and gray-scale value of another pixel located at θ direction and r pixels away from the former pixel is j (Fig. 3 (a)). Where, the displacement between two pixels is denoted $\delta=(r, \theta)$. The distance between pixels: r is 1, and the direction: θ are 0˚, 45˚, 90˚, 135˚. The crystallization growth states are not anisotropic, so we take the average of the results calculated by using four directions. By using the matrix P_δ, 14 texture feature values as shown in Table 1 are calculated. More detailed explanation is shown in [6].

If multiple type images are acquired from one original image by pre-processing it is possible to extract 14 feature values from each of them. In this paper, we utilize not only original images but also differential images in which the edge highlights. Therefore the number of utilized feature values is 28 in total.

Table 1 Listing of extracted feature values

No	texture feature values	
1	angular second moment :	$\sum_i \sum_j P^2(i,j)$
2	contrast:	$\sum_k k^2 P_{x-y}(k)$
3	correlation:	$\sum_i \sum_j ijP(i,j) - \mu_x \mu_y / \sigma_x \sigma_y$
4	sum of square : variance:	$\sum_i \sum_j (i-\mu_x)^2 P(i,j)$
5	inverse difference moment:	$\sum_i \sum_j P(i,j)/(1+(i-j)^2)$
6	sum average:	$\sum_k k P_{x+y}(k)$
7	sum variance:	$\sum_k (k - \sum_i l P_{x+y}(l))^2 P_{x+y}(k)$
8	sum entropy:	$-\sum_k P_{x+y}(k) \log P_{x+y}(k)$
9	entropy:	$-\sum_i \sum_j P(i,j) \log P(i,j)$
10	difference variance:	$\sum_k \{k - \sum_k k P_{x-y}(k)\}^2 P_{x-y}(k)$
11	difference entropy:	$-\sum_k P_{x-y}(k) \log P_{x-y}(k)$
12	information measure of correlation 1:	$HXY - XXY1 / \max\{HX, HY\}$
13	information measure of correlation 2:	$[1 - \exp\{-2.0(HXY2 - HXY)\}]^{1/2}$
14	maximal correlation coefficient: (second biggest eigen value of Q)$^{1/2}$	

These are calculated by using co-occurrence matrix P.

Where, $P_x(i) = \sum_j P(i,j)$, $P_y(j) = \sum_i P(i,j)$, $P_{x+y}(k) = \sum_i \sum_j P(i,j)$, $P_{x-y} = \sum_i \sum_j P(i,j)$, $\mu_x = \sum_i i P_x(i)$, $\mu_y = \sum_j j P_y(j)$, $\sigma_x^2 = \sum_i (i-\mu_x)^2 P_x(i)$, $\sigma_y^2 = \sum_j (j-\mu_y)^2 P_y(j)$, $HXY = -\sum_i \sum_j P(i,j) \log P(i,j)$, $HX = -\sum_i P_x(i) \log P_x(i)$, $HY = -\sum_j P_y(j) \log P_y(j)$, $HXY1 = -\sum_i \sum_j P(i,j) \log\{P_x(i) P_y(j)\}$, $HXY2 = -\sum_i \sum_j P_x(i) P_y(j) \log\{P_x(i) P_y(j)\}$ and $Q(i,j) = \sum_k P(i,k) P(k,j) / P_x(i) P_y(j)$

III. CLASSIFICATION METHOD

A. Design of classifier

1) Decision tree: Classification of protein crystallization states is typical sort of multi-classes and complicated problem. To deal with this classification, we divide the multi-class problem into some 2-class problems and construct a classifier consists of them. So the important point that we should think about is which classes in what order to classify. The classes for evaluation of the crystallization states are qualitatively defined by human. It is thought that there are some classes that are classified easily and other classes not so. Therefore, this study decides the classification sequence in descending order of classification accuracy. The sequence is able to be described in shape like a binary decision tree. The decision tree is generally used as the method to find potential subsets of one set, but the use in this paper is absolutely a description method of classification sequences.

The node of the tree consists of 5 elements:

Node $t = \{\varphi_t, \Omega_t, \Omega^L{}_t, \Omega^R{}_t, P_t\}$.

Here, φ is element classifier, Ω is total set of divided classes, Ω^L and Ω^R are subsets of classes that are send to left and right child node, respectively, and P is evaluation value.

In the following, derivation methods of these elements are described.

2) Criterion to decide subset of classes: This section describes a criterion to decide subset of classes: Ω^L and Ω^R. In this paper classification accuracies are used as the criterion.

Let the total set of divided classes at node t be Ω_t, all combinations of two subsets: $\Omega^L{}_x$ and $\Omega^R{}_x$ that fulfill $\Omega^L{}_x \cup \Omega^R{}_x = \Omega_t$ are targeted. Fig. 4 is an example of the flow diagram deciding the subset of classes at the top node for 3-class classification.

At first, Ω_t is classified into each combination and classification accuracies P_L_x and P_R_x are derived. Where, P_i = (the number of samples classified into Ω^i correctly) / (the total number of samples in Ω^i) (i = L, R). These accuracies are calculated by using k-fold cross-validation which is the most commonly used method to estimate generalization error. In k-fold cross-validation, the training data is divided into k subsets of approximately equal size and k times training are performed. Each time leaving out one of the subsets from training data, only the omitted subset is used as training data and other subsets is used as test data. (Fig. 4 (a))

In the second, evaluation value of each combination is $P_x = P\_L_x \times P\_R_x$. (Fig. 4 (b))

And finally, the subsets in which P_x takes the maximum value are determined as $\Omega^L{}_t$ and $\Omega^R{}_t$. (Fig. 4 (c))

3) Support Vector Machine for element classifier: It is possible to use any classifier to the element classifier of each node. In this paper, Support Vector Machine (SVM) is utilized.

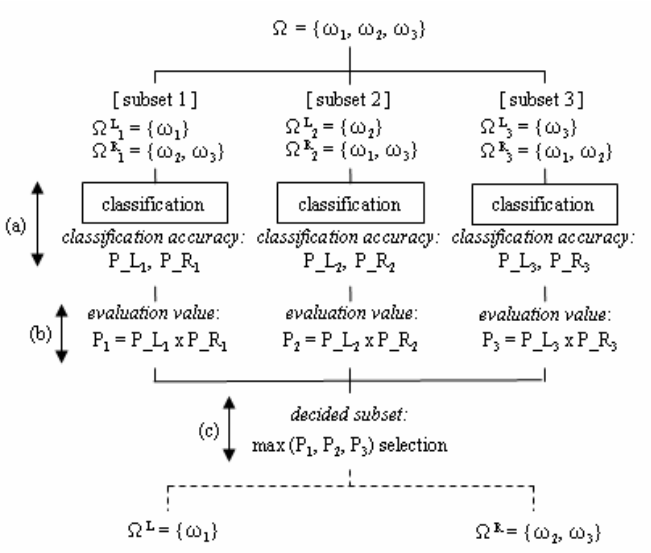


Fig.4 Flow diagram deciding the subset classes at node 1 in Fig.3

SVM is a technique to train classifiers, regressors and probability densities that is well-founded in statistical leaning theory [7]. One of the main attractions of using SVM is that they are capable of leaning in sparse, high dimensional spaces with very few training examples. SVM accomplish this by minimizing a bound on the empirical error and the complexity of the classifier, at the same time. The following is briefly overview the main concepts of SVM.

SVM perform pattern recognition for two-class problems by determining the separating hyperplane with maximum distance to the closest points of the training set. These points are called support vectors. If the data is not linearly separable in the input space, a non-linear transformation $\Phi(\cdot)$ can be applied which maps the data points $\mathbf{x} \in \mathbf{R}^n$ into a high dimensional space H which is called feature space. The data in the feature space is then separated by the optimal hyperplane as described above.

The mapping $\Phi(\cdot)$ is represented in the SVM classifier by a kernel function $K(\cdot,\cdot)$ which defines an inner product in H, i.e. $K(x, x') = \Phi(x)^T \Phi(x')$. The decision function of the SVM has the form:

$$f(\Phi(\mathbf{x})) = \sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i), \quad (1)$$

where m is the number of data points, and $y_i \in \{-1,1\}$ is the class label of training points x_i. Coefficients α_i in (1) can be found by solving a quadratic programming problem with linear constrains. The support vectors are the nearest points to the separating boundary and are the only ones for which α_i in (1) can be nonzero.

Examples of admissible kernel functions are the polynomial kernels:

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + 1)^d, \quad (2)$$

with d the degree of the polynomial, and the Gaussian kernels:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\| / 2\sigma^2), \quad (3)$$

with σ the variance of the Gaussian. In this paper, the Gaussian kernel is applied.

B. Introduction of the "gray-zone" class

Most of previous studies that attempted to automate the evaluation of crystallization growth states classified all samples into any class defined by human as well as a lot of other pattern classification problem. However, protein crystallization proceeds with time and evaluation of the samples around a classification boundary depends a lot on expert's empirical knowledge. Therefore we add new class named "gray-zone" for the area around boundary and the samples in this class are reclassified by human.

The following is proposal setting method of gray-zone. Consider the samples in 2-class as shown in Fig. 5 (a), where black and white points are class1 and class2, respectively. Fig. 5 (b) is the classification result using SVM, where the decision function $f(\cdot)$ in (1) = 0 at separating boundary. The gray-scale value of the figure background is according to $f(\cdot)$ in (1). The gray-scale value of the area where value of the function becomes the maximum is 255 (: white) and the gray-scale value where value becomes the minimum is 0 (: black). Focusing on this property, $f(\cdot)$ is applied to the decision of the area of gray-zone.

Step size of each class d_i, i = 1,2, are given by

$d_1 = \max(f(\cdot)) / 20$

$d_2 = \min(f(\cdot)) / 20$.

Then let r^i_n be the ratio of the samples in class i in the range 0 to the nth step (d_i x n), (n = 1,2,…,20). r^i_n is derived by

m^i_n / m^{all}_n,

where m^i_n is the number of the samples in class i in the range 0 to the nth step and m^{all}_n is the number of all samples in the range 0 to the nth step. Fig.6 shows the ratio r^i_n, (a) and (b) indicate Class1-side and Class2-side, respectively. The horizontal axis is step number and the vertical axis is the ratio. In both cases, the ratio around the boundary is not stabilized. The phenomena are attributable to the presence of miss-classification samples. Therefore the start step number of the range in which the ratio stably continues to increase k is decided, and the range 0 to kth step is defined as gray-zone. Suppose $\Delta r(n) = r(n+1) - r(n)$, k is the minimum value which satisfy the conditional expression

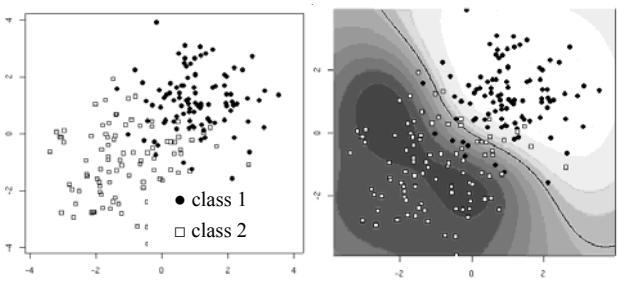
$$\Delta r(k) > 0 \wedge \Delta r(k+1) > 0 \wedge \ldots \wedge \Delta r(20) > 0. \quad (4)$$

In the case of Fig.6, k of (a) and (b) are 7 and 13, respectively.

Fig.7 shows the results of applying gray-zone to classification of the samples in Fig.5 (a). ((a) is only Class1-side, (b) is only Class2-side and (c) is both sides) Table 2 is a classification accuracy, where (a) is before applying gray-zone and (b) is after applying. Fig.8 shows transition of error rates.

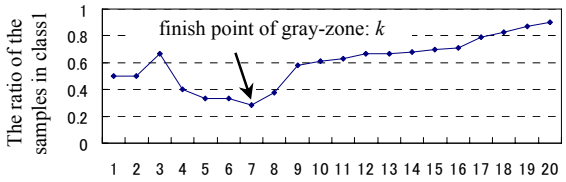
IV. EXPERIMENTS

In this section the method presented in previous section is sequentially applied to crystallization images.

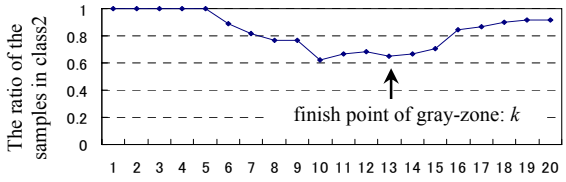


(a) Samples in 2-class problem (b) Classification result using SVM

Fig.5 Example of classification using SVM in 2-class problem

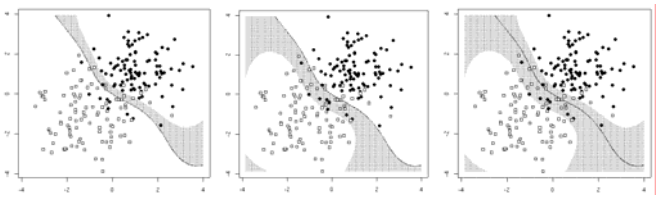


(a) Transition of ratio of the samples in Class1 at Class1-side



(b) Transition of ratio of the samples in Class2 at Class2-side

Fig.6 Transition of rate of the samples in each class



(a) Class1-side (b) Class2-side (c) Both sides

Fig. 7 Results of applying "gray-zone"

Table 2 Classification accuracy

(a) Before applying "gray-zone" class

		Classification result		Error rate
	Total	Class 1	Class 2	(%)
Class 1	100	92	8	8.00
Class 2	100	10	90	10.00

(b) After applying "gray-zone" class

		Classification result				
	Total	Class 1	Class 2	Gray-Zone	Total eorr	Error rate (%)
Class 1	100	90	0	10	0	0.00
Class 2	100	5	75	20	5	5.00

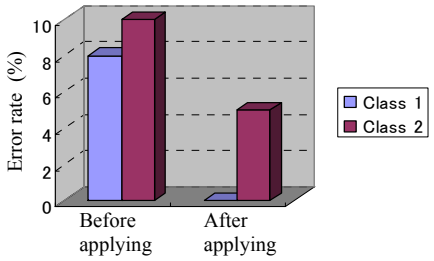


Fig. 8 Transition of error rates

A. *Preprocessing and feature extraction*

Half of the acquired images were used as a training set and others were used as a test set. They have already been scored by the experts. Before feature extraction, some pre-processing steps are carried out. Initially, the original color images are transformed into 256-level grayscale images because color information is not utilized in this experiment. Next, a 150[pixel] x 150[pixel] portion inside a well of the original image is manually extracted. The size of this area is determined by considering approximate average size of microcrystals and crystals in original images. Finally, the extracted image is differentiated with a Sobel first-order differential filter. Both of the differentiated and non-differentiated images are utilized in this paper.

From these pre-processed images, 28 feature values described in section II-C are extracted and the following experiments are carried out using these feature values.

B. *Design of decision tree*

By using training data, a classier was trained according to the algorithm provided in section III-*A* and evaluated by test data. The result is shown in Table 3 (C*i*, (*i* = 0, 1, 2, 3, 4) in the table means Class *i*). We compared the results obtained by this presented classification method ('automatic classification' in Table 3) with the classes by human ('manual classification') and calculated the concordance rates by the expression (accuracy) = (number of images classified correctly) / (total number) x 100. The accuracies were class 0: 96.08 %; class 1: 84.48 %; class 2: 61.54 %, class 3: 40.91% and class 4: 99.18%.

As performance evaluation, the result obtained by a classifier designed by "one versus others" was shown in Table 4. This is one of the usual construction types of two-class classifier for multi-class problem (Fig.9), where f_i is classifier for classification into class ω_i and other classes. Comparing the performances based on the accuracy, the total accuracy in our proposed method (Table 2) is (49+49+24+ 18+242) / 436 = 87.61 % and in "one versus others" method (Table 4) is (49+51+11+16+242) / 436 = 84.36 %, then the result of our method is higher than another.

C. *Redefining classification*

From Table 3, some classes are able to be classified with high accuracy. But there are many miss-classifications and it will not be possible to apply the current classification to automation system. Fig. 10 is the details of the classifier trained by the experiment in section IV-A, where P_L$_x$ and P_R$_x$ are classification accuracy and x is node number (x = 1,2,3,4). Total evaluation value of each node is P$_x$ = P_L$_x$ x P_R$_x$. Focus on the P$_x$ of each node,

Table 3 Classification result by proposed algorithm

			Automatic classificatation					Accuracy
		total	C0	C1	C2	C3	C4	(%)
Manual classification	C0	51	49	1	0	0	1	96.08
	C1	58	2	49	4	0	3	84.48
	C2	39	1	10	24	0	4	61.54
	C3	44	0	2	1	18	23	40.91
	C4	244	0	0	0	2	242	99.18

Table 4 Classification result by "one vs others" method

			Automatic classificatation					Accuracy
		total	C0	C1	C2	C3	C4	(%)
Manual classification	C0	51	49	1	0	0	1	96.08
	C1	58	3	51	1	0	3	87.93
	C2	39	1	15	11	0	12	28.21
	C3	44	0	3	0	16	25	36.36
	C4	244	0	0	0	2	242	99.18

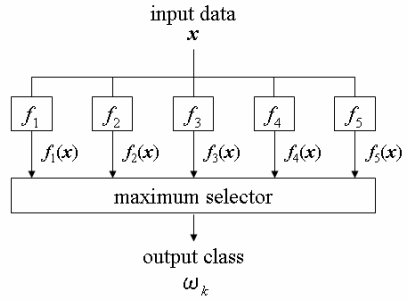


Fig. 9 Classifier designed by "one vs others"

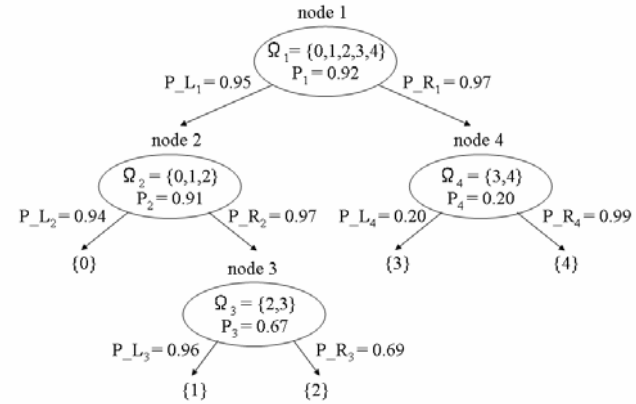


Fig. 10 Details of the classifier trained at the experiment in Table 3

Table 5 Classification result after redefining 3-class

			Automatic classificatation			Accuracy
		total	C0	C1 & C2	C3 & C4	(%)
Manual classification	C0	51	49	2	0	96.08
	C1 & C2	97	3	90	4	92.78
	C3 & C4	288	0	11	277	96.18

Table 6 Rewriting Table 5 based on mis-classification accuracy

			Automatic classification				
		Total	C0	C1 & C2	C3 & C4	Total error	Error rate (%)
Manual classification	C0	51	49	2	0	2	3.92
	C1 & C2	97	3	90	4	7	7.21
	C3 & C4	288	0	11	277	11	3.82

the evaluation value of P_3 and P_4 are low compared with other nodes. Therefore node3 and node4 are not be divided and we redefine classification from 5-class: $\{\omega_0\}$ $\{\omega_1\}$ $\{\omega_2\}$ $\{\omega_3\}$ $\{\omega_4\}$ to 3-class: $\{\omega_0\}$ $\{\omega_1,\omega_2\}$ $\{\omega_3,\omega_4\}$. Table 5 is the classification result. The result showed that the accuracies of all 3 classes are over 90.00 %.

D. Result utilizing the "gray-zone" class

After adding "gray-zone" class to redefined 3 classes, classification experiment is carried out. The effect of introducing "gray-zone" class is be able to be evaluated based on mis-classification accuracy. Table 5 is rewritten to Table 6 based on mis-classification accuracy. ("Error rate" in Table 6 correspond to mis-classification accuracy) Table 7 is the result applying "gray-zone" to the classification in Table 6.

VI. Discussion

In this paper the classification to automate evaluation of the crystallization states was constructed in 3 steps. First is design of classification by using decision tree, second is redefining classification and third is addition of "gray-zone" class. In all 436 test samples, the numbers of mis-classification were 54, 20 and 8 in the first, second and third step, respectively. Transition of the mis-classification rate is shown as Fig. 11. From Fig. 11, it can be confirmed that the mis-classification has decreased steadily in each step.

From the redefinition of the classification, the classes not classified have been caused. In future work it is an important problem to add feature sets enabling the classification of these classes.

VII. Summary

We have presented a classification method that evaluates protein crystallization growth states. The classifier was designed in descending order of classification accuracy by using decision tree and to each element classifier SVM was applied. Additionally division at lower-level node where classification accuracy was low was stopped and the original 5-class problem was redefined into new 3-class problem. And "gray-zone" class for the samples around classification boundary is introduced. The process has decreased mis-classification of the samples around boundary. From above-mentioned method it was possible to construct the classification that minimizes the mis-classification. These results indicate the meaningful contribution to the current evaluation work by human expert.

Table 7 Result applying "gray-zone" to the classification in Table 6

			Automatic classificatation					
		Total	C0	C1 & C2	C3 & C4	Gray -zone	Total error	Error rate (%)
Manual classification	C0	51	28	2	0	21	2	3.92
	C1 & C2	97	0	85	0	12	0	0
	C3 & C4	288	0	6	259	23	6	2.08

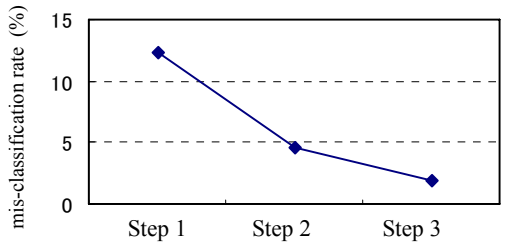


Fig. 11 Transition of mis-classification rate

Acknowledgment

The research was financially supported by the Sasakawa Scientific Research Grant from The Japan Science Society.

References

[1] E. Abola, P. Kuhn, T. Earnest and R. C. Stevens, "Automation of X-ray crystallography," Nature Struct. Biol., vol. 7, pp. 973-977, 2000.

[2] B. Rupp, "High-Throughput Crystallography at an Affordable Cost: The TB Structural Genomics Consortium Crystallization Facility," Acc. Chem. Res. 36, pp.173-181, 2003.

[3] G. Spraggon, S. A. Lesley, A. Kreusch and J. P. Priestle, "Computational analysis of crystallization trials," Acta Cryst. D58, pp. 1915-1923, 2002.

[4] M. Bern, D. Goldberg, R. C. Stevens and P. Kuhn, "Automatic classification of protein crystallization images using a curve-tracking algorithm," J. Appl. Cryst. vol. 37, pp. 279-287, 2004.

[5] K. Saitoh, et al, "Evaluation of protein crystallization states based on texture information derived from greyscale images," Acta Cryst. D61, pp. 873-880, 2005.

[6] R. M. Haralick, K. Shanmugam and I. Dinstein, "Texture feature for image classification," IEEE Trans. Syst., Man, Cybern., vol. SMC-3, no. 6, pp.610-621, 1973.

[7] V. Vapnik, *Statistical Learning Theory*, Willey & Sons, New York, 1998.