Published in final edited form as:

Cryst Growth Des. 2013 July 3; 13(7): 2728–2736. doi:10.1021/cg3016029.

Real-Time Protein Crystallization Image Acquisition and Classification System

Madhav Sigdel^{1,*}, Marc L. Pusey², and Ramazan S. Aygun¹

¹Department of Computer Science, University of Alabama in Huntsville, Huntsville, USA

²iXpressGenes, Inc., 601 Genome Way, Huntsville, USA

Abstract

In this paper, we describe the design and implementation of a stand-alone real-time system for protein crystallization image acquisition and classification with a goal to assist crystallographers in scoring crystallization trials. In-house assembled fluorescence microscopy system is built for image acquisition. The images are classified into three categories as non-crystals, likely leads, and crystals. Image classification consists of two main steps – image feature extraction and application of classification based on multilayer perceptron (MLP) neural networks. Our feature extraction involves applying multiple thresholding techniques, identifying high intensity regions (blobs), and generating intensity and blob features to obtain a 45-dimensional feature vector per image. To reduce the risk of missing crystals, we introduce a max-class ensemble classifier which applies multiple classifiers and chooses the highest score (or class). We performed our experiments on 2250 images consisting 67% non-crystal, 18% likely leads, and 15% clear crystal images and tested our results using 10-fold cross validation. Our results demonstrate that the method is very efficient (< 3 seconds to process and classify an image) and has comparatively high accuracy. Our system only misses 1.2% of the crystals (classified as non-crystals) most likely due to low illumination or out of focus image capture and has an overall accuracy of 88%.

INTRODUCTION

Protein crystallization is the most important part of protein crystallography studies. Numerous factors like protein purity, pH, temperature, protein concentration, the type of precipitant and the crystallization methods play an important role in crystallization [12]. The correct combination of all these factors is essential for the formation of crystals. However, it is difficult to predict exact conditions for protein crystallization [7]. Therefore, thousands of crystallization trials are often required for successful crystallization. Several robotic systems have been developed to automate crystallization process. Berry et al. (2006) [2] provide a review of the developments in high-throughput robotic set ups to automate the crystallization experiments.

Crystallization trials should be observed periodically to assess the evolving progress of crystal growth or crystallization pathway. Knowledge about the crystallization phase helps in making several decisions. Unsuccessful crystallization trials can be discarded. X-ray diffraction can be applied on single optically clear crystals. Likewise, if a protein is in the pathway of crystallization, the conditions can be optimized to get crystalline outcome [13]. Therefore, we desire a system that not only distinguishes between crystal or non-crystal classes but also identifies the likely lead conditions for optimization.

Corresponding Author: Madhav Sigdel, Department of Computer Science, University of Alabama in Huntsville, 300 Technology Hall, Huntsville, AL 35899, Fax: 256-824-6239, ms0023@cs.uah.edu.

Analysis of protein crystallization trials has the following challenges:

1. Since a large number of crystallization trials is required and these trials should be periodically assessed, manual analysis takes significant time.
2. Existing automated systems are very expensive and not portable.
3. Crystal detection is a complex process and usually requires complex image processing algorithms to extract features related to shapes of objects in an image. This makes it difficult to process and classify images in real time.
4. While achieving a real-time and automated system, a good level of accuracy needs to be maintained.

Because of the high throughput crystallization approach, manual review becomes impractical. Therefore, automated image scoring systems have been developed to collect and classify the crystallization trial images. The fundamental aim is to discard the unsuccessful trials, identify the successful trials and possibly also identify the trials which could be optimized. Significant amount of previous work (for example, Zuk & Ward (1991) [21], Cumba et al. (2003) [3], Cumba et al. (2005) [4], Zhu et al. (2006) [20], Berry et al. (2006) [2], Pan et al. (2006) [10], Po & Laine (2008) [11]) has described the classification of crystallization trials into non-crystal or crystal categories. Yang et al. (2006) [19] described classification into three categories (clear, precipitate and crystal). Bern et al. (2004) [1] classify the images into five categories (empty, clear, precipitate, microcrystal hit and crystal). Likewise, Saitoh et al. (2006) [15] described classification into five categories (clear drop, creamy precipitate, granulated precipitate, amorphous state precipitate and crystal). Spraggon et al. (2002) [17] have described classification of the crystallization imagery into six categories (experimental mistake, clear drop, homogenous precipitant, inhomogeneous precipitant, microcrystals, and crystals). Cumba et al. (2010) [5] have developed the most optimistic system which classifies the images into three categories or ten categories. It should be noted that there is no standard for categorizing the images and different research studies have proposed different categories in their own way.

Most of the proposed algorithms start image processing by determining the region of interest (droplet boundary) to define the search region for crystals, a computationally expensive process. The general technique applied here is to first apply an edge detection algorithm such as Sobel edge detection or Canny edge detection which is followed by some curve fitting algorithms like Hough transform (Berry et al. (2006) [2], Pan et al. (2006) [10], Spraggon et al. (2002) [17], Zhu et al. (2004) [20]). Bern et al. (2004) [1] determined the drop boundary by applying edge detection followed by dynamic programming curve tracking algorithm. Yang et al. (2006) [19] used dynamic contour method on Canny edge image to locate the droplet boundary. Cumba et al. (2003) [3] applied a probabilistic graphical model with a two-layered grid topology to segment the drop boundary. Po & Laine (2008) [11] have used multiple population genetic algorithms for region of interest detection. Saitoh et al. (2004) [14] and Saitoh et al. (2006) [15] simplify this process by defining a fixed $150[\text{pixel}] \times 150[\text{pixel}]$ portion inside a well as the region of interest for search of crystals.

For feature extraction, a variety of image processing techniques has been proposed. Zuk & Ward (1991) [21] used the Hough transform to identify straight edges of crystals. Bern et al. (2004) [1] extract gradient and geometry related features from the selected drop. Pan et al. (2006) [10] used intensity statistics, blob texture features, and results from Gabor wavelet decomposition to obtain the image features. Research studies Cumba et al. (2003) [3], Saitoh et al. (2004) [14], Spraggon et al. (2002) [17], and Zhu et al. (2004) [20] used a combination of geometric and texture features as the input to their classifier. Saitoh et al. (2006) [15]

used global texture features as well as features from local parts in the image and features from differential images. Yang et al. (2006) [19] derived the features from gray-level co-occurrence matrix (GLCM), Hough Transform and Discrete Fourier Transform (DFT). Liu et al. (2008) [8] extracted features from Gabor filters, integral histograms and gradient images to obtain 466-dimensional feature vector. Po & Laine (2008) [11] applied multi-scale Laplacian pyramid filters and histogram analysis techniques for feature extraction. Cumba et al. (2010) [5] present the most sophisticated feature extraction techniques for the classification of crystallization trial images. Features like basic statistics, energy, Euler numbers, Radon-Laplacian features, Sobel-edge features, microcrystal features and GLCM features are extracted to obtain a 14,908 dimension feature vector. Although increasing the number of features may help improve accuracy, it may slow down the classification process. In addition, the use of irrelevant features may deteriorate the performance of some classifiers.

Because of the high-throughput rate of image collection, the speed of processing an image becomes an important factor. One of the most time consuming step is the determination of a region of interest or the drop boundary. Likewise, extraction of a large number of geometric and texture features increases the time and image processing complexity. The system by Pan et al. (2006) [10] required 30 seconds per image for feature extraction. Po & Laine mention that it takes 12.5 seconds per image for the feature extraction in their system [11]. Due to high computational requirement, they are considering implementation of their approach on the Google computing grid. Feature extraction described by Cumba et al. (2010) [5] is the most sophisticated which could take 5 hours per image on a normal system. To speed up the process, they execute the feature extraction using a web-based distributed computing system. Overall, the image processing and feature extraction has been computationally expensive making it infeasible for real time processing.

To obtain the decision model for classification, a variety of classification techniques have been used. Zhu et al. (2004) [20] and Liu et al. (2008) [8] applied a decision tree with boosting. Bern et al. (2004) [1] used a decision tree classifier with hand crafted thresholds. Pan et al. (2006) [10] applied a SVM learning algorithm. Saitoh et al. (2006) [15] applied a combination of decision tree and SVM classifiers. Spraggon et al. (2002) [17] applied self-organizing neural networks. Po et al (2008) [11] combined genetic algorithms and neural networks to obtain a decision model. Berry et al. (2006) [2] determined scores for each object within a drop using learning vector quantization, self-organizing maps and Bayesian algorithms. The overall score for the drop is calculated by aggregating the classification scores of the individual objects. Cumba et al. (2003) [3] and Saitoh et al. (2004) [14] applied linear discriminant analysis. Yang et al. (2006) [19] applied hand-tuned rules based classification followed by linear discriminant analysis. Cumba et al. (2005) [4] used association rule mining while Cumba et al. (2010) [5] used multiple random forest classifiers generated via bagging and feature subsampling.

With regards to correctness of classification, the best reported accuracy for the binary classification i.e., classification into two categories is by Po et al. (2008) [11] 93.5% average true performance (88% true positive and 99% true negative rates). Saitoh et al. have achieved accuracy in the range 80–98% for different image categories [14]. Likewise, the automated system by Cumba et al. (2010) [5] detects 80% of crystal-bearing images, 89% of precipitate images, and 98% of clear drops accurately. The performance of the various systems however cannot be compared directly as they have used different datasets, different class categories and number of categories. The current systems are not fully reliable and there is still much room for improvement in terms of performance.

An alternative approach to classification of crystallization trial images is offered by the use of fluorescence. Studies on trace fluorescently labeled proteins have shown image intensity to be proportional to the structure or packing density of the proteins solid state [7, 13]. The fluorescence approach considerably simplifies finding crystals in a droplet, reducing the problem to one of finding the high intensity regions, as opposed to finding the straight lines or particular shapes of objects that are often of low contrast. Morphological analysis can be carried on sufficiently intense regions to determine if they can be formally classified as a crystal (presence of straight lines), or as a 'bright spot' lead condition. This makes the feature extraction phase simple and faster than traditional pure image processing systems using white light images for protein crystal detection.

Since very few experiments lead to the crystallization state, there is a class imbalance in the images of different states. Typical classifiers are biased towards crowded classes, in this case, the non-crystal or clear drop category. Although overall accuracy is improved, crystals might be missed. Hence, an evaluation based on overall accuracy can be misleading. The most important goal for such an automated system is to accurately detect all existing crystals. This is best measured by sensitivity (recall) for the crystal category. This work presents an efficient and effective approach for the protein phase classification with high sensitivity for crystal detection. Here, phase refers to the state of the protein - in solution (soluble) or solid (crystalline or amorphous precipitate). In this paper, we describe the design and implementation of a stand-alone real-time system for protein image acquisition and classification of crystallization trials. The images are classified into three categories as non-crystals, likely leads, and crystals. We try to minimize the number of features to develop a real-time classification system while maintaining comparatively high accuracy. The proposed system can be employed in portable and real-time systems. The contribution of this paper can be summarized as follows:

1. We describe the design of a low cost assembled fluorescence microscopy system. We propose trace-fluorescent labeling of protein solution that results in higher image intensity for the solution containing crystals, thereby simplifying the feature extraction process.
2. We present efficient image processing steps to find an effective feature vector for classifying images. This makes it possible for real-time analysis of protein images. Our feature extraction takes less than 3 seconds to extract 45 features from an image.
3. We classify the images into three categories: non-crystal, likely leads, and crystals. We introduce max-class ensemble classifier that reduces the risk of missing crystals. Our system exhibits high sensitivity for crystal detection with an accuracy comparable to other systems.

The remaining part of the paper is organized as follows. The design of the acquisition system, image preprocessing, feature extraction and classification methodology are described in the experimental section. Next section provides the results and discussion on the classification. The last section provides concluding remarks and future enhancements on the proposed acquisition and classification system.

EXPERIMENTAL SECTION

This section describes the image acquisition, image preprocessing, feature extraction and the classification methods.

IMAGE ACQUISITION

To automate the image acquisition process, we use an in-house assembled fluorescence microscopy system. It has been shown that trace fluorescent labeling can be a powerful tool for visually finding protein crystals [7, 13].

The layout of the acquisition system is shown in Fig. 1. Only fluorescent or white light images are collected, the system has no provision for polarizers to collect birefringence images. Excitation light is supplied by an ultra-bright LED (Mightex Systems, Toronto, Canada), with excitation, dichroic, and emission filters of the XF30 set (Omega Optical, Brattleboro, VT) housed in filter cube custom made by a local machine shop. The light is focused and the fluorescence imaged by 35 mm imaging lense (Edmund Optics, cat. #59872). Image acquisition is done by a color camera (UI-5580SE, IDS Imaging, Germany) connected to the computer through an ethernet cable. Stepper motors are interfaced through drivers (Weeder Technologies) connected to a serial port of the connected PC. The motors control the position of the stages (Velmex Inc., Bloomfield, NY) in the X and Y axis and of the camera (focus) in the Z axis. The crystallization plate is placed manually in the plate holder. Six types of plates have been programmed to date Greiner 3 well (Greiner Bio-one, Monroe, NC); Art Robbins Intelliplate 2 well (Art Robbins, Sunnyvale, CA); Corning Crystal EX 3 well, and 1 well (Corning, Acton, MA); Microlytic CFHT (Microlytic, Burlington, MA); and Matrical 1536 well (Matrical, Spokane, WA). Additional plate designs can be readily accommodated. Stage positioning parameters are read in for the selected plate type from a text file, the input parameters include the spacing in steps between wells, the offsets of the crystal growing drop positions within each well, and additional offsets to account for the plate not being perfectly square and perpendicular with respect to the optical system. At the start of the data collection process, the camera gain settings are also read in for the fluorescent probe. Different fluorescent probes can be accommodated by changing the excitation source and the filters in the cube. The X, Y and Z movement of the stepper motor allows the camera to be positioned to exact well position. The configuration of the well plates is maintained in a file.

The basic flow of the image acquisition is shown in Fig. 2. The protein crystallization screening plate is manually loaded into the assembled microscopy system. First, the probe configuration is loaded and the camera is initialized with proper settings. The plate configuration is then loaded to seek the co-ordinate of each well in the plate. At the start, the camera is positioned to the top-left corner of the well-plate. For each well, the camera is positioned above the well and the image is captured and saved in the repository. This process is repeated until all the wells are scanned. It takes around 30 minutes to collect images from a 3-celled 96-well plate.

IMAGE CATEGORIES

Hampton Research defines a scoring system having a range of 9 outcomes for a crystallization trial. In this paper, the crystallization trial images are classified into 3 categories: non-crystals, likely leads, and crystals. The mapping of these phases into our image categories with respect to Hampton's research is shown in Table 1. Higher class value implies better crystals or higher probability for crystallization.

Non-crystals - This category consists of images under the following phases: clear drop (initial state of the crystallization process), phase separation, or regular amorphous precipitates. This category indicates that the conditions corresponding to these images do not favor crystallization. Therefore, these can be discarded in further experiments. Figure 3 shows some sample images under this category.

Likely leads—This category consists of images corresponding to likely lead conditions, and hence, can be a good starting point for optimizing the crystallization conditions. Birefringent precipitate or microcrystals fall under this category. Some images have bright regions, however, it is not clear whether the bright regions indicate crystals or not. The images could be affected by conditions such as improper focusing, camera lighting, etc. Since high intensity might indicate the presence of crystals, these images should be reviewed by an expert. Hence we include such images under likely leads category. Figure 4 shows some sample images in this category.

Crystals—This category includes images having clear crystals. The crystals can have different shapes and sizes like needle, spherulites, plates, or 3D crystals. Figure 5 shows some sample images in this category.

IMAGE PROCESSING AND FEATURE EXTRACTION

Expert-classified images are used to obtain a decision model for the classification of new images. We focus on fast and effective image processing techniques so that the time for processing an image is less than the time between collecting two images. The steps of the image processing and feature extraction are explained below.

Consider an image I of size $H \times W$. Let $I(x,y)$ represent the pixel at location (x,y) where $x \in [0, H-1]$ and $y \in [0, W-1]$. In a color image, each pixel consists of red (R), green (G), and blue (B) components and can be described as a 3-tuple (R, G, B) . The red, green and blue intensity values of a pixel at $I(x,y)$ are represented as $I_R(x, y)$, $I_G(x, y)$, and $I_B(x, y)$, respectively.

Figure 6 shows the components of our system. Firstly, we downsample the image and apply median filter for noise removal. Next, we generate binary images by three thresholding techniques. Then, we extract image intensity features by combining the binary image and median filtered image. Likewise, we generate blobs from the binary images and extract features related to the shape or size of the individual objects. Details on the feature extraction process are explained below.

Image down-sampling—A high resolution image may keep unnecessary details for image classification, especially, if the image has significant noise. In addition, processing a high resolution image increases the computation time significantly. Therefore, the images are down-sampled before further processing. Suppose image $\mathbf{I}$ ($H \times W$) is to be down-sampled by k times. Then the resulting image $\downarrow \mathbf{I}$ is of size $h \times w$ where $h = H/k$ and $w = W/k$. In our experiments, the original size of the images is 2560×1920 pixels. By down-sampling it by 8-fold, image size is reduced to 320×240 pixels. Our analysis shows that the down-sampled images contain sufficient detail for feature extraction.

Noise removal—The down-sampled image $\downarrow \mathbf{I}$ is passed through a median filter to remove random scattered noise pixels. Among different filters, a median filter provided the best results for noise removal of our datasets. To apply the median filter, a neighborhood window of size $(2p+1) \times (2q+1)$ around a point (x, y) is selected. Suppose ${}^x_p R_q^y$ represents a region in the original image centered around (x, y) with top left coordinate $(x-p, y-q)$. F maps a 2D data into 1D set and $\text{median}({}^x_p R_q^y)$ provides the median value in the selected neighborhood around (x, y) . The red component in the resulting region (image) is denoted by $M_r(x, y)$ and is given by Equation (1):

$$M_r(x, y) = \text{median}(F({}^x_p \downarrow I_r^y_q)) \quad (1)$$

Similarly, the components for green, $M_g(x, y)$, and blue, $M_b(x, y)$, are calculated.

Grayscale conversion—The result from the median image $\mathbf{M}$ is a color image with RGB values for each pixel. From this image, a grayscale image $\mathbf{G}$ is derived which consists of a single intensity value for each pixel. The gray-level intensity at each pixel is calculated as the average of the color values for red, green, and blue components in $\mathbf{M}$. The conversion can be expressed in the form of Equation (2):

$$\mathbf{G}(x, y) = [\mathbf{M}_r(x, y) + \mathbf{M}_g(x, y) + \mathbf{M}_b(x, y)] / 3 \quad (2)$$

Thresholding—Thresholding is applied to create a binary (black and white) image from a color or grayscale image. Essentially, the objective is to classify all the image pixels as a foreground (object) or a background pixel. In basic thresholding, a threshold value $\tau \in [0, 255]$ is selected. The set of pixels with gray-level intensity below the threshold τ are considered as background pixels and the remaining are considered as foreground pixels. A pixel in the binary image, $B(x, y) \in \{0, 1\}$, is defined as in (3):

$$B = \tau_{\rightarrow} (G) = \begin{cases} B(x, y) = 0 & \text{if } G(x, y) < \tau \\ B(x, y) = 1 & \text{otherwise} \end{cases} \quad (3)$$

If the threshold changes based on the content of an image, such thresholding is called as dynamic thresholding. Images vary depending on crystallization techniques and imaging devices. This makes it difficult to use a fixed threshold for binarization. Therefore, dynamic thresholding methods are preferred. Different thresholding techniques provide good results for different images. Hence, combining the results from multiple thresholding techniques is helpful. We apply three dynamic thresholding methods – Otsu's thresholding [9], 90th percentile green intensity threshold, and maximum green intensity thresholding. The implementation and results for each of these techniques are described next.

Otsu's thresholding—Otsu's method (Otsu 1979) [9] iterates through all possible threshold values and calculates a measure of spread of the pixel levels in foreground or background region. The threshold value (τ_0) for which the sum of foreground and background spreads is minimal is selected and binary image ($B_{\text{otsu}} = \tau_{0\rightarrow} (G)$) is constructed applying this threshold. Down-sampled images and corresponding Otsu thresholded images followed by the median filter is given in Fig. 7.

From the original and binary images in Fig. 7, we can observe that a single technique may not yield good results for all images. In the binary images 7(b), and 7(d), the objects and background are distinguished well. However, in the binary image 7(f) for the original image 7(e), objects and the background are not well separated. Hence, the result is not as desired. If the protein solution drop is also illuminated, crystals are not distinguishable in the thresholded image. This causes difficulty in extracting correct features from the image.

90th percentile green intensity threshold (G_{90})—For the fluorescence based acquisition with green light as the excitation source, we could observe that the intensity of the green pixel component is higher than the red and blue components in the crystal regions. We deploy this feature in our algorithm for thresholding. The threshold intensity (τ_{g90}) is computed as the 90th percentile intensity of the green component in all pixels. This means that the number of pixels in the image with the green component intensity below this

intensity constitute 90% of the pixels. After the threshold intensity is computed, Equation (3) is applied using $\tau = \tau_{g90}$ to generate the binary image ($B_{g90} = \frac{\tau_{g90}}{\tau_{g90}} \rightarrow (M_g)$).

Maximum green intensity threshold (G_{100})—This is similar to the earlier thresholding technique except maximum intensity of green component (τ_{g100}) is used as threshold to generate the binary image ($B_{g100} = \frac{\tau_{g100}}{\tau_{g100}} \rightarrow (M_g)$). The foreground (object) region in the binary image from this method is usually smaller than the binary image from the 90th percentile threshold method.

Figure 8 shows image **M** after noise removal and the results of applying three thresholding methods on this image. For this particular image, the binary image from the maximum green intensity threshold produces the best result.

Binary images from all of the three thresholding methods are used for feature extraction. The extracted features can be divided into two general categories – intensity features and region features. More details on the extraction of these features are described below.

Intensity features—Binary images obtained from the above steps are used as a mask to differentiate foreground and background region. Figure 9 (a) shows an original image. The image in Fig. 9(b) is the binary image of image in Fig. 9(a) obtained by applying a thresholding technique. Figure 9(c) shows the image with background pixels from the original (median-filtered) image and foreground pixels in black. Similarly, Fig. 9(d) shows the foreground pixels in the original image with background pixels in black.

We then extract the following statistical features related to intensity.

1. Threshold intensity (τ) for the corresponding thresholding technique
2. No of white pixels in the binary image (N_f)

$$N_f = \sum_{i=1}^h \sum_{j=1}^w B(i, j) \quad (4)$$

3. Average image intensity in the foreground region (μ_f)

$$\mu_f = \frac{1}{N_f} \sum_{i=1}^h \sum_{j=1}^w G(i, j) \cdot B(i, j) \quad (5)$$

4. Standard deviation of intensity in the foreground region (σ_f)

$$\sigma_f = \sqrt{\frac{1}{N_f} \sum_{i=1}^h \sum_{j=1}^w ((\mu_f - G(i, j)) \cdot B(i, j))^2} \quad (6)$$

5. Average image intensity in the background region (μ_b)

$$\mu_b = \frac{1}{h * w - N_f} \sum_{i=1}^h \sum_{j=1}^w G(i, j) \cdot (1 - B(i, j)) \quad (7)$$

6. Standard deviation of intensity in the background region (σ_b)

$$\sigma_b = \sqrt{\frac{1}{h * w - N_f} \sum_{i=1}^m \sum_{\substack{j=1, \\ T(i,j)=0}}^n ((\mu_b - G(i,j)) \cdot (1 - B(i,j)))^2} \quad (8)$$

Region features—After thresholding, we essentially expect crystals to be distinguished as objects. More information about the crystals is obtained by extracting shape features like uniformity, symmetry, size, etc. These features are important to describe the features of objects that are not crystals. Such objects are obtained when the overall thresholding does not yield a good result.

Region segmentation—We apply connected component labeling [16] on binary images to extract high intensity regions or blobs. The binary image could be obtained from any of the thresholding methods. Let **O** be the set of the blobs in a binary image **B**, and **B** consists of **n** number of blobs. The i^{th} largest blob is represented by O_i where $0 < i < n$ and $area(O_i) > area(O_{i+1})$. Each blob O_i is enclosed by a minimum boundary rectangle (MBR) centered at $m_{x,y}^i$ or $m^i(x, y)$ having width (w_i), and height (h_i). In our implementation, we define the

minimum size of the blob to be 3×3 pixels. The MBR of O_i is represented as $\frac{m_x^i}{2} R_{\frac{w_i}{2}}^{\frac{m_y^i}{2}} \frac{h_i}{2}$.

Figure 10(b) is a binary image of the original image in Fig. 10(a) which consists of 4 blobs. Extraction of the individual blobs is given in Fig 10(c).

For each O_i , we apply skeletonization ($\Omega_i = skel(O_i)$) to get the boundaries of the blob. The skeletonization is a ‘hit and miss’ morphological operation with the structuring element given by matrix *S*.

$$S = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Each point in a binary image of O_i where the pixel’s neighborhood matches the structuring element is a hit and the corresponding pixel in the output is zero; otherwise it remains the same. The resulting image consists of objects converted to single pixel thickness. Figure 10(d) shows the skeletonization of the blobs in Fig 10(c).

Area of the blob (a_i)—Let B^i represent the binary image of the minimum bounding rectangle of O_i . The area or the number of white pixels in O_i can be calculated as

$$a_i = \sum_{x=1}^{h_i} \sum_{y=1}^{w_i} B^i(x, y).$$

Measure of fullness (f_i)—Measure of fullness indicates whether the blob completely covers its MBR or not. It is calculated as the ratio of area of the blob to the area of its MBR

$$\text{i.e., } f_i = \frac{a_i}{(w_i * h_i)}$$

Boundary pixel count (N_b)—The skeleton image (Ω_i) is used to compute the number of pixels in the boundary of the blob. This is calculated using $N_b^i = \sum_{x=1}^{h_i} \sum_{y=1}^{w_i} \Omega_i(x, y)$

Measure of boundary uniformity (u_1, u_2)—A measure of boundary smoothness is calculated by comparing the distance of each boundary pixel from the center of the MBR to the assumed radius ($r = \frac{w_i + h_i}{2}$) as shown in Fig. 11(a). Let P_i be the set of points on the perimeter of the blob O_i , i.e., the skeleton Ω_i . We define two measures u_1^i and u_2^i related to boundary uniformity defined by Equations (9) and (10) respectively.

$$u_1^i = \frac{1}{N_b^i} \left(\sum_{p \in P_i} |dist(p, m^i(x, y)) - r| \right) \quad (9)$$

$$u_2^i = 1 - \frac{1}{N_b^i} \sum_{p \in P_i} z_{x,y}^i \quad (10)$$

where $z_{x,y}^i$ is defined such that

$$z_{x,y}^i = \begin{cases} 0 & \text{if } |dist(p, m^i(x, y)) - r| \leq \varepsilon \\ 1 & \text{otherwise} \end{cases}$$

Here, ε is the allowable difference set as $(\varepsilon) = 0.15 * \frac{w_i}{2} * \frac{h_i}{2}$.

Measure of symmetry (ξ)—Symmetry can be a useful measure especially in distinguishing irregular objects. We calculate the measure of left-right symmetry (symmetry along Y-axis) as shown in Fig. 11(b). Each blob is scanned row-wise. Let p_k be the k^{th} boundary pixel. Then the measure of symmetry (ξ_i) corresponding to the blob O_i is calculated as in equation (11).

$$\xi_i = 1 - \frac{1}{N_b^i} \sum_{p \in P_i} z_{x,y}^i \quad (11)$$

where $z_{x,y}^i$ is defined such that

$$z_{x,y}^i = \begin{cases} 0 & \text{if } |dist(p_k, m_{x,y}^i) - dist(p_{k+1}, m_{x,y}^i)| \leq \varepsilon \\ 1 & \text{otherwise} \end{cases}$$

Using the above measures, we calculate 9 blob related features which are listed below.

1. No of blobs (η)
2. Area of the largest blob (a_1) = $\sum_{i=1}^m \sum_{j=1}^n B_{x,y}^1$
3. The largest blob fullness (f_1) = $\frac{a_1}{(w_1 * h_1)}$
4. The largest blob boundary pixel count (N_b^1) = $\sum_{i=1}^m \sum_{j=1}^n \Omega_{x,y}^1$
5. The largest blob boundary uniformity measure (u_1^1) as defined in (9).

6. The largest blob uniformity measure (u_2^1) as defined in (10).
7. The largest blob measure of symmetry (ζ_1) as defined in (11)
8. Average area of the top 5 largest blobs excluding largest blob (a_{avg}) = $\frac{1}{k} \sum_{i=2}^k a_i$ and $k = \min(\eta, 6)$ where η is the number of blobs.
9. Average fullness of the top 5 largest blobs excluding largest blob (f_{avg}) = $\frac{1}{k} \sum_{i=2}^k f_i$ and $k = \min(\eta, 6)$

Table 2 provides the 9 blob-related features for the image in Fig. 10(a) with thresholding method G100. Six of these features are related to the largest blob. It should be noted that the blobs may not necessarily represent crystals in an image. In some binary images, the whole drop can appear as a large white region. Thus, the features from the largest area are important not only to identify crystals but also to identify falsely thresholded images. Besides the features from the largest blob, we extract the average area and average fullness from top 5 large sized blobs excluding the largest blob. These features provide aggregated information for the other large blobs and are especially useful to distinguish precipitates where the binary image consists of many blobs with non-uniform shapes.

For each image, we apply 3 thresholding techniques and obtain 3 binary images. From each binary image, we extract 6 intensity related features and 9 blob related features. Therefore, we extract a total of $3 \times (6+9) = 45$ features per image.

CLASSIFICATION METHODS

We use two classification techniques to classify protein crystallization trial images. One of them applies multilayer perceptron neural network on the dataset. The second one, max-class ensemble method, is proposed by us to mitigate the risk of missing images having crystals.

Multilayer perceptron neural network (MLP)—MLP is a widely used classification algorithm in pattern recognition problems [6]. The model consists of one or more hidden layers between input and output layers and weights are associated with connecting nodes. Training is done using back propagation learning algorithm. MLP classifier is applied over a 45 dimension vector obtained using the features from all three thresholding methods ($Otsu$, G_{90} , G_{100}).

Max-class ensemble method—Ensemble methods provide a model for combining predictions from multiple classifiers. Essentially, the goal is to reduce the risk of misclassification. Bagging and boosting are two popular methods of selecting samples for ensemble methods [18]. Most often, majority voting or class-averaging is used to determine the result score from an ensemble classifier. Protein crystallization has a class imbalance problem. Not necessarily all classes are represented at the same amount. All precipitates start with the first state and only successful crystallization process will lead to the last state (crystalline outcome). The number of precipitates that lead to crystals is minority. Typical classifiers are biased towards the crowded classes and try to predict them with high sensitivity. Although overall accuracy is improved, the crystals might be missed. The cost of missing a crystal is significantly high. A majority voting approach that is used by traditional ensemble techniques might fail for these cases. To avoid missing crystals or to assign proper scores to precipitates that are close to the crystallization phase, we propose our *max-class ensemble* method. The *max-class ensemble* method works as follows. Let $M_k^t(p_m)$ denote the class of the precipitate P_m using classifier M_k at time instant t . Then the *max-class*

ensemble method is defined as $\max_{1 \leq k \leq w} (\max_{1 \leq t \leq \tau} M_k^t(p_m))$ where $1 \leq k \leq w$ and $1 \leq t \leq \tau$ assuming w classifiers and τ observations.

Feature extraction depends on the quality and correctness of the binary (or thresholded) images. As mentioned earlier, the comparative performance of thresholding techniques may vary for different images. Therefore, in our ensemble method, we execute 3 MLP classifiers using the features from each thresholding method (*Otsu*, *G₉₀*, *G₁₀₀*). We execute another MLP classifier with all features combined. Each image has now 4 predicted classes which could be the same or different. The result class (or score) is the maximum class (or score) from all these classifiers.

SYSTEM INFORMATION

Our software is developed as a windows form based application in C#. Image feature extraction is implemented using AForge Imaging library (<http://code.google.com/p/aforge/>). Training and classification is implemented using Weka data mining library (<http://www.cs.waikato.ac.nz/ml/weka/>). Visual Studio 2010 is used as the IDE. The user interface consists of a tabbed layout for image acquisition, image scoring, and system settings. The repository of images is maintained inside a directory and the records are maintained in MySQL database. On a Windows 7 Intel Core i7 CPU @2.8 GHz system with 4 GB memory, it takes around 1 hour 40 minutes to process and classify 2250 images. This amounts to less than 3 seconds to process and classify an image, and fits well into the average sample to sample translation time of ~6 seconds for our system.

RESULTS AND DISCUSSION

Our dataset consists of 2250 images that are manually classified by an expert according to the Hampton Research scale. For this work, we consider classification into 3 categories – non-crystals, likely leads, and crystals. Mapping from the original score to 3 categories is done following Table 1. Most images belong to the non-crystal category. We add more crystal images into the dataset to include all kinds of crystals and to reduce the class imbalance in the training. The distribution of the image categories is given in Table 3.

There is no perfect classification system and each classification system is susceptible to missing crystals. If two classes (crystals and non-crystals) are defined, the expert needs to go over images classified as- a) crystals to verify them, and b) non-crystals to detect missing crystals. This would require the expert to check all images and this type of classification is not helpful for the expert. In our experiments, we have “likely leads” as the third class so that the expert may need to scan in this class to detect missing crystals but not the images in non-crystals. This would save significant effort in terms of manual scanning all images. Testing is done by applying 10-fold cross validation. In this process, the entire training set is first split randomly into 10 equal sized subsamples. Each subsample is used for testing while remaining subsamples are used for training. This process is repeated 10 times with each subsample being used exactly once for testing. The results are combined to get a single estimation for the complete training set.

Using MLP classifier

The classification results using MLP with a single hidden layer, 24 nodes in the hidden layer and 0.3 learning rate is provided in the form of a contingency table in Table 4. The overall accuracy is 90% [(1469+299+262)/2250]. Table 4 also provides the precision-recall using one-vs-all for each category. The non-crystals are fairly well detected (97%). This category corresponds to the crystallization conditions which are discarded from further experiments. Since most of the images belong to this category, the effort for manual review of the

classification results is greatly reduced. Likewise, the recall for the likely leads and crystals categories are 0.74 and 0.79, respectively. The system misses around 2% (6 out of 332) actual crystals. The images classified as likely leads and crystals are to be reviewed by an expert. The misclassification of the non-crystal images to the higher categories leads to 6% $[(42+3) / (405+324)]$ unnecessary checks for the images that surely do not contain crystals. Figure 12(a) shows the precision-recall plot with this approach. The precision and recall for non-crystal category is very high compared to the measures for the other two categories.

Using max-class ensemble

Classifying a crystal as a non-crystal is a more critical problem than classifying a non-crystal as a crystal. Because of the cost of missing a crystal, the critical performance measure for protein crystallization data set is the recall of the crystal category. The recall using MLP classifiers is 0.79, which is not high. To overcome this issue, we have done experiments with our max-class ensemble classification method. The max-class ensemble classifier is biased towards high classes (or scores). The classification results using a max-class ensemble classifier over four MLP classifiers (three MLP classifiers with 15-dimensional feature inputs from Otsu, G90 and G100 thresholding methods, and another MLP classifier with 45-dimensional feature input) is given in Table 5. The max-ensemble class chooses the highest score (or class) among these classifiers.

The overall accuracy with max-class ensemble method is 88% $[(1402+270+312)/2250]$. In comparison to the first approach, the number of false negatives (i.e., classifying an image in a high class to a lower class) is decreased at the cost of increase in false positives (i.e., classifying an image in a low class to a higher class). Figure 12 (b) shows the precision-recall plot. Both the precision and recall for class 0 (non-crystals) are very high. The precision for class 0 (non-crystals) is close to 99%. The recall for class 2 is increased from .79 (first method) to 0.94 (max-class ensemble method). Only 1.2% (4 out of 332) actual crystals are predicted as non-crystals.

Figure 13 shows crystal images that are predicted as non-crystal. The image intensity in the images in Fig. 13(a) and Figure 13(b) are very low. The crystal images missed have either very low image intensity or are blurred. Higher intensity excitation lighting, camera gain settings, and adequate focusing can help eliminate such errors.

CONCLUSION

In this paper, we described the design and implementation of a stand-alone system for protein crystallization image acquisition and crystallization. The image acquisition utilizes a low cost in-house assembled fluorescence microscopy system. Image analysis is carried out on the fluorescence images. The main advantage of this approach is the ability to rapidly identify crystals and potential lead crystallization conditions by analyzing image intensity and high intensity regions. We describe the implementation of an efficient (fast) and effective (with good accuracy) image classification system to automatically classify the images into one of noncrystal, likely leads, or crystal categories. We describe three dynamic thresholding methods and extraction of intensity and region features using the binary images to obtain a 45- dimensional feature vector which is input to the classification algorithm.

We described the classification into 3-categories and compared our max-class ensemble classifier with a (single) MLP classifier. By using a single MLP classifier, we obtained around 90% accuracy. However, the sensitivity for crystal category was low. We introduced max-class ensemble method to reduce the risk of error. By this method, the percentage of missing crystals as non-crystals is around 1.2%. This means that the system only misses 1.2% of the crystals. Our system exhibits high accuracy for the non-crystal category. This

minimizes the unnecessary reviews (images in non-crystal category) by the expert. Therefore, the effort for manual review is greatly reduced. Image processing for a 96-well plate with 3 cells in each well takes less than 15 minutes. This time is less than the time for acquisition which takes around 30 minutes to scan through the whole plate. This allows the image acquisition and classification to be executed in parallel.

Even though the correct classification of non-crystal images is very high, the system does not distinguish between the likely-leads and crystal categories very well. This makes the manual review essential for the two categories. As future work, we intend to improve this accuracy by extracting other relevant image features. We are also working towards classifying the crystals according to morphology such as needles, plates, and 3D crystals. Likewise, we plan to track temporal evolution of the crystals. Images at multiple timestamps can be used to determine size evolution over time, another indication of a crystalline object.

Acknowledgments

This research was supported by National Institutes of Health (GM-090453) grant.

References

1. Bern M, et al. Automatic classification of protein crystallization images using a curve-tracking algorithm. *J Appl Cryst.* 2004; 37:279–287.
2. Berry IM, et al. SPINE high-throughput crystallization, crystal imaging and recognition techniques: current state, performance analysis, new technologies and future aspects. *Acta Cryst.* 2006; D62:1137–1149.
3. Cumbaa CA, et al. Automatic classification of sub-microlitre protein-crystallization trials in 1536-well plates. *Acta Cryst.* 2003; D59:1619–1627.
4. Cumbaa CA, Jurisica I. Automatic classification and pattern discovery in high-throughput protein crystallization trials. *J Struct Funct Gen.* 2005; 6:195–202.
5. Cumbaa CA, Jurisica I. Protein Crystallization Analysis on the World Community Grid. *J Struct Funct Genomics.* 2010; 11:61–69. [PubMed: 20072819]
6. Duda, RO.; Hart, PE.; Stork, DG. *Pattern Classification.* 2. Wiley; 2000.
7. Forsythe E, Achari A, Pusey Marc L. Trace fluorescent labeling for high-throughput crystallography. *Acta Crystallogr D Biol Crystallogr.* 2006; 62(Pt 3):339–346. [PubMed: 16510981]
8. Liu R, Freund Y, Spraggon G. Image-based crystal detection: a machine-learning approach. *Acta Crystallogr D Biol Crystallogr.* 2008; 64:1187–1195. [PubMed: 19018095]
9. Otsu N. A threshold selection method from gray level histograms. *IEEE Trans Syst Man Cybern.* 1979; SMC-9:62–66.
10. Pan S, et al. Automated classification of protein crystallization images using support vector machines with scale-invariant texture and Gabor features. *Acta Crystallogr D.* 2006; 62:271–279. [PubMed: 16510974]
11. Po MJ, Laine AF. Leveraging Genetic Algorithm and Neural Network in Automated Protein Crystal Recognition. *EMBS.* 2008:1926–1929.
12. Pusey ML, et al. Life in the fast lane for protein crystallization and X-ray crystallography. *Progress in Biophysics and Molecular Biology.* 2005; 88(3):359–386. [PubMed: 15652250]
13. Pusey ML, et al. Fluorescence Approaches to Growing Macromolecule Crystals. *Methods in Molecular Biology.* 2008; 426:377–385. [PubMed: 18542877]
14. Saitoh K, et al. Evaluation of Protein Crystallization States Based on Texture Information. *IROS.* 2004:2725–2730.
15. Saitoh K, Kawabata K, Asama H. Design of Classifier to Automate the Evaluation of Protein Crystallization States. *Robotics and Automation ICRA.* 2006:1800–1805.
16. Shapiro, L.; Stockman, G. *Computer Vision.* Prentice Hall; 2002. p. 69-73.

17. Spraggon G, et al. Computational analysis of crystallization trials. *Acta Cryst.* 2002; D58:1915–1923.
18. Tan P, Steinbach M, Kumar V. *Introduction to Data Mining.* 2005
19. Yang, X., et al. Image-Based Classification for Automating Protein Crystal Identification. *Lecture Notes in Con. Inf. Sciences*; Berlin / Heidelberg: Springer; 2006. p. 932-937.
20. Zhu, Xiaoqing; Sun, Shaohua; Bern, M. Classification of Protein Crystallization Imagery. *EMBS* '04. 2004
21. Zuk WM, Ward KB. Methods of analysis of protein crystal images. *J of Crystal Growth.* 1991; 110:148–155.

SYNOPSIS

We describe a stand-alone real time system for protein crystallization image acquisition using LED based fluorescence microscopy system and crystallization images classification into non-crystal, likely leads, or crystal categories. Our image classification is efficient (fast) and effective (with good accuracy). Our system only misses 1.2% of the crystals (classified as noncrystals) and has an overall accuracy of 88%.

Figure 1.
Layout of assembled microscopy system

Figure 2.
Basic flowchart of protein image acquisition process

Figure 3.
Sample images under non-crystal category

Figure 4.
Sample images under likely leads category

Figure 5.
Sample images under crystals category

Figure 6.
Flowchart of the integrated algorithm for feature extraction

Figure 7.
a, c, and e: median-filtered images; b, d, and f: the Otsu thresholded images for a, c, and e, respectively

Figure 8.

Results of showing the application of the 3 thresholding techniques for a sample image. a) Original b) Otsu's threshold c) 90th per threshold d) Max green threshold

Figure 9. Separating background and foreground region. a) M (Image after noise removal) b) Binary image c) Background pixels d) Foreground pixels

Figure 10.

a) M (image after noise removal) b) Binary image, B c) O: Objects (regions) using connected component labeling d) Ω : the skeletonization of object

Figure 11.
a) Boundary uniformity b) Measure of symmetry

Figure 12. Precision-recall plot (NC: non-crystal; LL: likely leads; C: crystal, O: Overall) a) Using MLP classifier with all features b) Using max-class ensemble method

Figure 13.
Crystals classified as non-crystal using max-class ensemble classifier

Table 1

Mapping of image categories and Hampton's categories

Image category	Hampton's category	
Non-crystals (Class 0)	1	Clear Drop
	2	Phase Separation
	3	Regular Granular Precipitate
Likely leads (Class 1)	4	Birefringent Precipitate or Microcrystals
	*	Unclear bright regions (Not present in Hampton's category)
Crystals (Class 2)	5	Posettes or Spherulites
	6	Needles (1D Growth)
	7	Plates (2D Growth)
	8	Single Crystals (3D Growth < 0.2 mm)
	9	Single Crystals (3D Growth > 0.2 mm)

Table 2

Table showing the blobs and blob features for the image in Fig 10(a)

#	Features	Feature values

1	No of blobs (η)			4	
2	Area (a)	1065	1197	553	598
3	Fullness (f)	0.65	0.78	0.71	0.79
4	Boundary pixel count (N_b)	158	156	108	106
5	Boundary uniformity (u_1)	0.49	0.39	0.87	0.65
6	Boundary uniformity (u_2)	3.58	3.37	1.27	2.12
7	Measure of symmetry ($\hat{\xi}$)	0.62	0.80	0.71	0.74
8	Average area (a_{avg})	583.33			
9	Average fullness (f_{avg})	0.50			

Table 3

Distribution of images into different categories

Category	No of images	Percentage
Non Crystals	1514	67.3%
Likely Leads	404	18.0%
Crystals	332	14.8%
Total images	2250	

Table 4

Classification results using MLP

Actual\Observed	0	1	2	Actual Total	Recall
0	1469	42	3	1514	0.97
1	46	299	59	404	0.74
2	6	64	262	332	0.79
Observed Total	1521	405	324	2250	0.83
Precision	0.97	0.74	0.81	0.84	

Table 5

Results using max-class ensemble classifier

Actual\Observed	0	1	2	Actual Total	Recall
0	1402	97	15	1514	0.93
1	8	270	126	404	0.67
2	4	16	312	332	0.94
Observed Total	1414	383	453	2250	0.84
Precision	0.99	0.70	0.69	0.80	