Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on Publication Date: 14-17 Oct. 2007 Page(s): 286-293 ISBN: 978-1-4244-1509-0 Digital Object Identifier: 10.1109/BIBE.2007.4375578 Posted online: 2007-11-05 14:20:32.0

Rapid Detection and Classification of Bacterial Contamination Using Grid Computing

Wamiq M. Ahmed[1], Bulent Bayraktar[1], Arun K. Bhunia[2], E. Dan Hirleman[3], J. Paul Robinson[4,5], and Bartek Rajwa[5]
[1]School of Electrical and Computer Engineering, [2]Molecular Food Microbiology Laboratory,
[3]School of Mechanical Engineering, [4]Weldon School of Biomedical Engineering,
[5]Bindley Bioscience Center, Purdue University, West Lafayette, IN 47907

***Abstract*—Bacterial contamination of food products is a serious public health problem that creates high costs for the food-processing industry. Rapid detection of bacterial pathogens is the key to avoiding disease outbreaks and costly product recalls associated with food-borne pathogens. Automated identification of pathogens using scatter patterns of bacterial colonies is a promising technique that uses image processing and machine learning approaches to extract features from forward-scatter patterns produced by irradiating bacterial colonies with red laser light. The feature vector used for this approach can consist of hundreds of features, and a sufficiently large number of training images is required for accurate classification. As most feature extraction algorithms have high computational cost, the feature extraction step becomes the bottleneck in the whole processing pipeline. Computational grid technologies provide a promising and economical solution to this problem. In this work we report the implementation of the laser-scatter-analysis technique on a computational grid. A set of more than 2000 images was used for training of classifiers. The invariant form of Zernike moments up to order 20, radial Chebyshev moments, and Haralick features were extracted. Linear discriminant analysis and support vector machine classifiers were used for classification. We report speed-ups achieved and the scalability of this approach for large sets of images and for higher-order moments. Laser-scatter-analysis technique combined with computational grid technology offers a feasible and economic solution for rapid and accurate detection and classification of bacterial contamination.**

Keywords-Bacterial contamination; grid computing; feature extraction; classification

I. INTRODUCTION

Food products contaminated by bacteria are a serious risk for the public and are responsible for various disease outbreaks and health hazards. Contaminated products also generate serious costs for the food-processing industry because of product recalls. Fast and accurate identification of pathogens present in the contaminated products is extremely important in order to avoid the harmful effects of such contamination. Numerous analysis techniques have been proposed for this purpose [1, 2]. Most current methods utilize expensive biochemical or molecular biology–based technologies and require complex sample preparation for accurate pathogen detection and recognition. Analysis and classification of microorganisms using forward-scatter patterns is a newly proposed, inexpensive, label-free technique [3, 4]. This approach requires a laser to illuminate bacterial colonies grown on agar plates, and a digital camera connected to a computer to collect information about forward-scattered light patterns. A number of different features, including Zernike moments, Chebyshev moments, and Haralick texture features, are extracted from the resultant patterns, providing the means for automated, rapid classification. This new technique provides reproducible results and does not require any special chemical treatments or sample preparation. However, accurate classification requires that the classifiers be trained and optimized using large training sets, even if just a few bacterial classes are detected. Extracting higher-order shape moments and texture features from large sets of patterns is extremely time consuming and becomes the bottleneck for classifier optimization. Hence the speed of the feature extraction step determines the speed of training. For industrial application, where thousands of samples may need to be processed rapidly, feature extraction slows down the testing phase as well. Computational grid technologies provide a cost-effective solution to this problem [5]. Computational grids harness the computing power of commodity, heterogeneous resources that are not subject to centralized control, and make these resources available to applications that need them. Grid technologies provide an economical solution to meet the high-throughput computing needs of researchers by integrating diverse computing resources. This integration of resources has been made possible by grid toolkits like Globus and Condor [6, 7]. Condor is a distributed software system that manages diverse, heterogeneous computing resources and makes them available for compute-intensive jobs as a single resource. Globus is an open-source toolkit used for building grid systems. It provides tools for resource management, communication, process creation, and data access. In recent years computational grid technologies have been used for many applications in bioinformatics, such as the analysis of protein folding and biological sequence alignment [8-11].

In this paper, we describe the implementation of the forward scatter–based bacterial identification system on a computational grid. The processing pipeline of the serial implementation is shown in Fig. 1. The scatter patterns produced by scattering of laser light from bacterial colonies are first preprocessed to remove experimental artifacts. This involves image centering and adaptive histogram equalization. Image features are then extracted from these images. The output of the feature extraction step is a feature vector that

contains many hundred features. The most discriminative features are then selected and the classifier is trained on these features. The trained classifier can then be used for classifying test scatter patterns of bacterial colonies. The feature extraction is the most computationally expensive processing phase. Hence significant improvement in processing time can be achieved by speeding up this step of the processing pipeline. We implemented the feature extraction process on the grid maintained by our university. The image set is processed in parallel on the grid for feature extraction and these features are then used for classification.

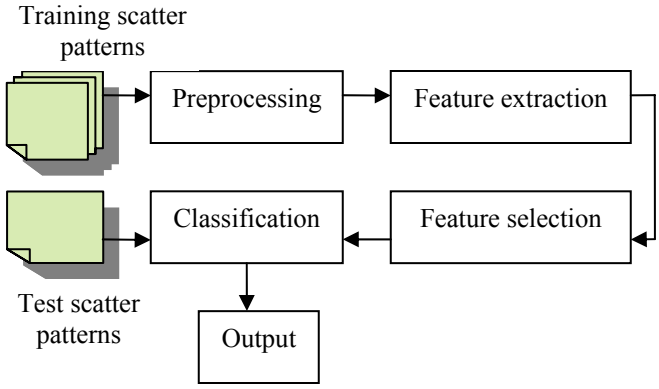

Figure 1. Processing pipeline for laser scatter technique.

The main contribution of this paper is a fast implementation of light-scatter technique that makes this technique feasible and economic for rapid and accurate classification, and for widespread use. Feature extraction during the training of classifiers is computationally the most expensive step and it benefits greatly from grid computing. In industrial settings where hundreds or thousands of potential cases of contamination are to be analyzed, grid implementation can significantly help the testing phase as well. The fast implementation also makes it possible to train the classifiers on much larger data sets than is possible with sequential implementation. These larger training sets improve the classification accuracy of classifiers. We also explore the computational cost of higher-order shape moments, which can potentially improve classifier performance. Grid computing as the enabling technology paves the way for developing a large database of scatter patterns for numerous species of bacteria. Classifiers trained on such a database would be able to quickly detect and classify a wide variety of bacterial contaminations.

The organization of the paper is as follows. Section II describes the forward scatter–based bacterial identification technique. Section III discusses image features and classification algorithms used for colony identification. Implementation of feature extraction algorithms on the grid is explained in Section IV. Section V presents the results of experiments and Section VI concludes the paper.

II. Bacterial Cultures and the Laser Scatterometer

The bacterial cultures, including various species and strains of *Listeria*, *E. coli*, *Salmonella*, *Staphylococcus*, and *Vibrio*, were serially diluted in sterile 20 mM phosphate buffered saline (PBS), pH 7.4, so that the dilutions would produce about 30-50 colonies per plate. The diluents were evenly distributed on the surface of brain heart infusion (BHI) agar plates in duplicate and were incubated at 37°C for 18-36 h or until the colony reached 1.8 to 2 mm in diameter. The thickness of the colony (along the optical axis) was measured from the surface profile data obtained by a laser triangulation probe (Microtrak II Laser Displacement Sensor System, MTI instruments Inc., Albany, NY), and was typically around 0.3 to 0.4 mm. The laser scatterometer has been described in detail in [3, 4]. Briefly, scatter patterns from bacterial colonies with a diameter of approximately 1.8 to 1.9 mm and a thickness of around 0.3 to 0.4 mm were collected using the laser scatterometer system. The laser generated a collimated beam of light on the order of 1 mm in diameter (at the $1/e^2$ irradiance points) that was directed through the center of the bacterial colony and the substrate of bacterial agar medium. The forward-scattered light and the transmitted light formed the scatter patterns on the detector. The resultant images (640x480 pixels) were cropped to 300x300 pixels by keeping the center of the circularly shaped scatter patterns in the geometric center of the image and selecting a 300x300 rectangle around it.

III. Processing Algorithms for Bacterial Colony Identification

A. Feature extraction

Feature extraction is the identification of particular characteristics of an object of interest in an image. The proper selection of these characteristics is the key to the success of many recognition and analysis tasks. The features we used for our analysis include Zernike and Chebyshev moments, and Haralick texture. Since their introduction by Hu [12], moments have been utilized in numerous applications ranging from optical character recognition and face recognition to image registration. Such features capture intrinsic information about the image and do not require objects with closed boundaries. For a 2D continuous function f(x,y), moments have the general form of $M_{pq} = \iint f(x,y) h_{pq}(x,y) dxdy$, where $h_{pq}(x,y)$ is a polynomial in x and y with powers p and q, respectively. The different polynomials lead to different types of moments. If f(x,y) is a digital image, then $M_{pq} = \sum_x \sum_y f(x,y) h_{pq}(x,y)$.

One important aspect of feature extraction is to find suitable features for a specific application. Owing to the circular nature of scatter patterns, we used features with radial properties (Zernike and radial Chebyshev (Tchebichef) moment invariants). Additionally, as bacterial scatter patterns exhibit specific textures, we used Haralick texture features as another input to the classifier.

B. Zernike polynomials and moments

In (r, θ) polar coordinates, the Zernike radial polynomials $R_{nm}(r)$ are defined as [13]

$$R_{nm}(r) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!} r^{n-2s}, \quad (1)$$

where n is a non-negative integer, and m is a non-zero integer subject to the following constrains: n-|m| is even, and n ≥ |m|. The (n,m) order of the Zernike basis function $V_{nm}(r, \theta)$, defined over the unit disk is

$$V_{nm}(r,\theta) = R_{nm}(r)\exp(jm\theta) \quad (2)$$

The Zernike moment of an image is then defined as

$$Z_{nm} = \frac{n+1}{\pi} \iint_{\text{unit disk}} V^*_{nm}(r,\theta) f(r,\theta), \quad (3)$$

where V^*_{nm} is a complex conjugate of V_{nm}.

To compute the Zernike moments of a given image, the center of the image is taken as the origin and pixel coordinates are mapped to the range of the unit circle. Under rotation, the orientation angles of the Zernike moments change but their magnitude remains unchanged. Therefore, the magnitudes of Zernike moments $|Z_{nm}|$ can be used as rotation-invariant features.

C. Chebyshev polynomials and radial moments

Chebyshev moments, unlike Zernike, belong to the class of discrete orthogonal moments. Therefore the implementation does not involve any numerical approximations. The scaled Chebyshev polynomials t_n for an image of size NxN are defined according to the following recursive relations [14, 15] :

$$t_0(x) = 1, \quad t_1(x) = (2x - N + 1)/N$$

$$t_p(x) = \frac{(2p-1)t_1(x)t_{p-1}(x) - (p-1)\left\{1 - \frac{(p1-)^2}{N^2}\right\}t_{p-2}(x)}{p}, \; p > 1$$

$$\text{where} \quad \rho(n,N) = \frac{N(N^2-1)(N^2-2^2)...(N^2-n^2)}{2n+1}. \quad (4)$$

The radial Chebyshev moments of order p and repetition q are defined as

$$S_{pq} = \frac{1}{2\pi\rho(p,m)} \sum_{r=0}^{m-1} \sum_{\theta=0}^{2\pi} t_p(r)\exp(-jq\theta) f(r,\theta)$$

$$m = (N/2) + 1 \quad (5)$$

In the above equation, both r and θ take integer values. The mapping between (r,θ) and image coordinates x, y is given by

$$x = \frac{rN}{2(m-1)}\cos\theta + \frac{N}{2} \qquad y = \frac{rN}{2(m-1)}\sin\theta + \frac{N}{2} \quad (6)$$

The m and n constants can be selected to suit the desired sampling frequency. Typically m has a value which is at least N/2, and n is 360 when the image is sampled at one-degree intervals. As with Zernike moments, it can be shown that magnitudes of radial Chebyshev ($|S_{pq}|$) moments are invariant to rotation.

D. Texture analysis and Haralick texture features

A primary tool for quantifying texture is a gray-level co-occurrence matrix. These matrices are used to quantify the number of occurrences at various distances and angles of pixel intensity values with respect to each other. The so-called Haralick texture features are then employed to extract 14 low- and high-frequency (depending on the distance from each other of pixels used in the co-occurrence matrix) texture properties [16, 17]. We used the mean and the range of 12 of these 14 features, which constitutes 24 features per image.

Formally, let image I have N_x pixels in the horizontal direction and N_y pixels in the vertical direction. Suppose also that there are N_g distinct gray-tone levels in the quantized (digital) image. Let $L_x = 1, 2, \ldots, N_x$ be the horizontal spatial domain, $L_y = 1, 2, \ldots, N_y$ be the vertical spatial domain, and $G = 1, 2, \ldots, N_g$ be the set of N_g distinct gray levels (tones). The texture-context information in image I is contained in the overall or "average" spatial relationship which the gray tones in image I have with one another. More specifically, this texture-context information is adequately specified by the matrix of relative frequencies P_{ij} with which two neighboring pixels separated by a distance d occur on the image, one with gray level i and the other with gray level j. Such matrices of gray-tone spatial-dependence frequencies are a function of the distance between them. A pixel has (excluding the borders) eight nearest-neighbor pixels (north, south, east, west, northwest, northeast, southwest, southeast).

After the number of pixel pairs R used in computing a particular gray-tone spatial-dependence matrix is obtained, the matrix can be normalized by dividing each entry in the matrix by R. Note that only the distinct gray levels are used to build the P matrices. If the gray levels are in the range [0, 255] and all are used in the image, then the P matrix will be a 256-by-256 matrix. Using the co-occurrence matrix, we can quantify texture using features such as angular second moment, contrast, sum average, sum variance, inverse difference moment, sum of squares (variance), entropy (a measure of randomness), sum entropy, difference entropy, difference variance, information measure of correlation, and maximal correlation coefficient [16, 17].

E. Feature selection

Feature selection deals with the problem of selecting the most relevant subset of features [18, 19]. Feature selection is an important analysis step for applications that have to deal with a large number of features. Among the benefits of feature selection are reduced measurement, storage, and processing time; faster and more accurate classifiers; and an improved understanding of the underlying data-generation process [19]. Feature subset selection methods are generally divided into filter and wrapper methods [20]. Filter approaches select a subset of features based on their discriminative power, whereas

wrapper approaches select a subset of features by wrapping the feature selection process around the classifier. The performance of the feature selection process, in the wrapper approach, is judged by the performance of the classifier. Variable ranking is a filter approach that ranks variables according to their capacity to discriminate between classes found in the training set. Many different criteria can be used for ranking features. For our experiments we used Fisher's criterion, which selects features based on the ratio of inter-class variance to intra-class variance [21].

F. Machine learning

Machine learning deals with tools and algorithms that learn through experience. Machine learning algorithms are generally divided into different categories that include supervised and unsupervised algorithms. In the supervised case the learning algorithm is provided with labeled examples which it uses for learning, whereas in the case of unsupervised learning such examples are not provided. Numerous algorithms have been proposed for supervised and unsupervised learning. These include decision trees, maximum likelihood classifiers, neural networks, and support vector machine (SVM) classifiers [22]. For our application we used an SVM classifier. SVM algorithms attempt to maximize the margin between different classes [23]. SVM algorithms achieve this by mapping the input to a higher dimensional space and constructing the separating hyperplane. In our case, extracted features were selected based on Fisher's criterion, and the SVM classifier was first trained on these examples and then used for classifying previously unseen scatter patterns.

IV. IMPLEMENTATION OF FORWARD SCATTER–BASED BACTERIAL IDENTIFICATION ON THE GRID

Grid computing provides high-throughput computing resources. Applications that have extensive parallelism and do not require significant message passing can best utilize the grid resources. As all the images used for feature extraction are processed independently, this application is ideally suited for grid computing. A block diagram of the sequential processing steps is shown in Fig 1. The images are first centered and adaptive histogram equalization is performed. Image features, including Zernike moments, Chebyshev moments, and Haralick texture features are extracted and the classifier is trained on these features. The run times on a single processor for preprocessing, feature extraction, and classification for a set of 20 images are shown in Table I. As is obvious, the feature extraction step is the most compute-intensive and slows down the bacterial identification process. With this observation in mind, we implemented the feature extraction step on the grid. The pipeline for parallel implementation is shown in Fig. 2. Scatter images are first preprocessed in a serial manner and then parallel jobs for feature extraction are submitted to Condor. The task distribution module does load balancing and assigns jobs evenly to available processors, making sure that the workloads of any two processors do not differ by more than one image. The feature vectors produced by these parallel jobs are then integrated, and input to the classifier. The classifier is trained on these training data and is used for classifying test scatter images.

TABLE I. TIME TAKEN BY DIFFERENT PROCESSING STEPS. A TOTAL OF 20 IMAGES WERE USED

Processing step	Time (sec)
Preprocessing	< 10
Feature extraction	3853
Classification	< 10

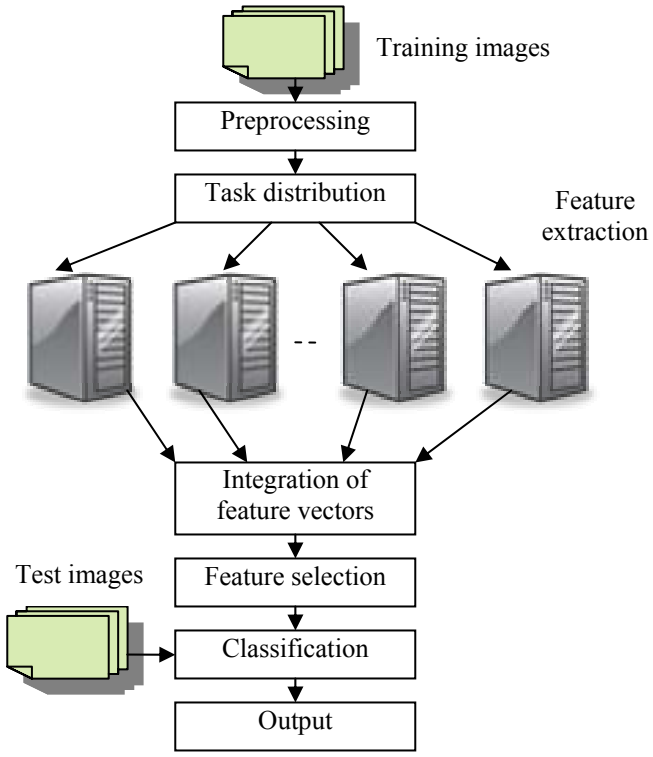

Figure 2. Processing pipeline for implementation of laser-scatter analysis technique on the grid.

The grid environment at our university consists of several high-performance clusters and a network of commodity computers as shown in Fig. 3. These clusters are owned by different research groups. When processors in these clusters are idle, they are made available to the Condor job scheduler. Cluster A consists of Dell X86_64 EM64T systems. It has 512 nodes that have 3.2 GHz Pentium IV dual processors. Each node has 4 GB of memory and the system has a total disk space of 20.48 TB. The nodes are connected using Gigabit Ethernet. Cluster B consists of Dell IA-32 Pentium IV systems with processor speed of either 3.06 GHz or 3.2 GHz. It has 308 nodes with two processors on each node. The system has a total disk space of 11.088 TB. The nodes of this cluster are interconnected through Gigabit Ethernet and in some cases with InfiniBand. Cluster C has 132 nodes that consist of HP X86_64 AMD64 systems. It has a total disk space of 6.72 TB. The nodes of this cluster are connected with either Gigabit Ethernet or InfiniBand. Each node of cluster D has two 900 MHz Itanium IA-64 I2 processors. Each node has 8 GB of memory. This system has a total disk space of 372 GB. The nodes in this system are connected with Gigabit Ethernet. Cluster E consists of machines that have been phased out by instructional labs and other departments. Processors in this

cluster have various processing speeds and memories and are interconnected through 100MB or Gigabit Ethernet.

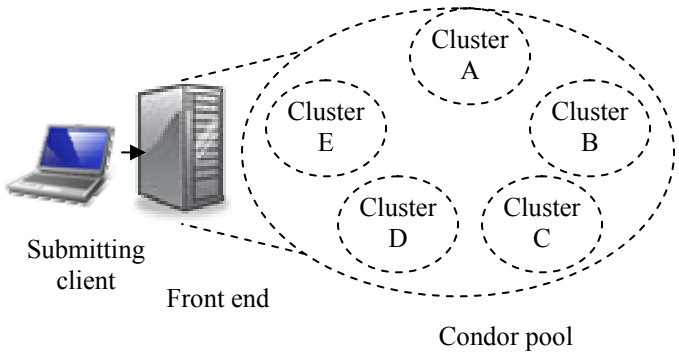

Figure 3. The Condor pool consists of 4 tightly coupled clusters and one cluster of commodity processors.

V. EXPERIMENTS AND RESULTS

For our experiments we used a set of 2234 scatter patterns produced by bacterial colonies. Each pattern had a size of 300x300 pixels. The bacterial species used included *E. coli*, *Listeria*, *Salmonella*, *Staphylococcus*, and *Vibrio*. A representative set of scatter patterns measured for *Staphylococcus* and *Salmonella* colonies is shown in Fig. 4. All of the feature extraction algorithms were implemented in Matlab. Since Condor runs at a lower priority on clusters compared to batch jobs submitted through portable batch system (PBS), a Condor job is evicted if a higher priority job is assigned to the same node. The Matlab jobs are not check-pointed in Condor, and therefore job eviction causes the jobs to restart from the beginning. In order to minimize the chances of job eviction we used the processors with maximum processing power (using kflops and rank features in Condor script). A maximum of 16 processors were used for our experiments.

A. Speed-up and scalability for feature extraction

In order to evaluate the speed-ups achieved by scaling the number of processors and the number of images, we used a set of 128 scatter patterns. The number of processors was varied from 1 to 16. The execution times are shown in Table II. The resultant speed-ups are linear with respect to the number of processors as shown in Fig. 5, suggesting that this approach is easily scalable. In the subsequent experiment, we fixed the number of processors at 16 and scaled the number of scatter patterns from 32 to 512. The resultant execution times are shown in Fig. 6. We observed that the runtimes are linear with respect to the number of scatter patterns and thus scaling to higher number of images is possible.

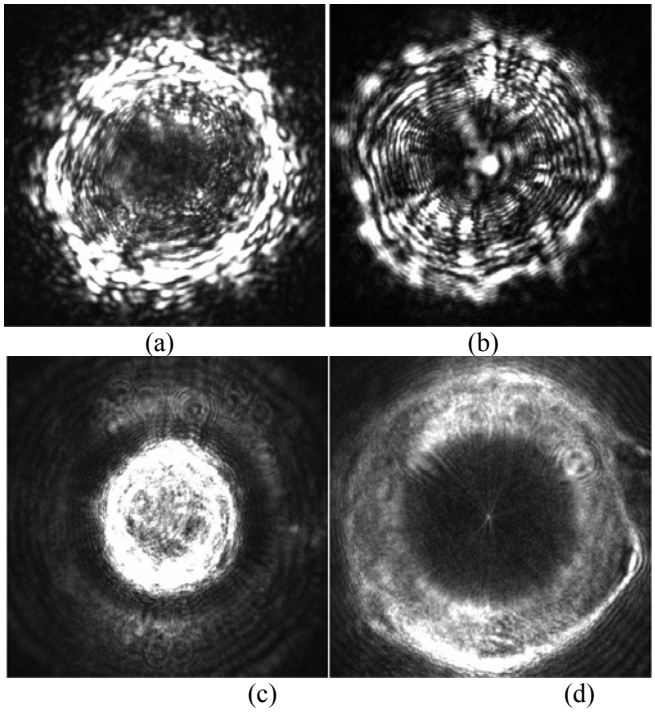

Figure 4. Representative images of scatter patterns formed by (a) *Salmonella copenhagen* (b) *Salmonella enteriditis 13096* (c) *Staph. aureus S41* (d) *Staph. epi ATCC35547*.

TABLE II. RUN TIME AND SPEED-UP FOR DIFFERENT NUMBER OF PROCESSORS. A SET OF 128 IMAGES WAS USED

Number of processors	Time (sec)	Speed-up
1	22640	1
2	11382	1.99
4	6176	3.67
8	3098	7.31
16	1547	14.63

B. Scalability for order of moments

The objective of the next set of experiments was to analyze the scaling behavior of higher-order moments. We used 16 processors and a set of 64 scatter images. Zernike moments of order 10 through 30 were calculated in steps of 5. The resulting run times are shown in Fig. 7. We observed a sharp increase in the run time after order 25, suggesting that calculation of higher-order moments requires increased processing power. In another experiment, radial Chebyshev moments of order 10 through 50 were calculated in steps of 10 using 16 processors and a fixed set of 64 scatter patterns. The run times are shown in Fig. 8. We noted good scaling behavior of Chebyshev moments up to order 40, after which a sharp increase in the processing requirements occurred.

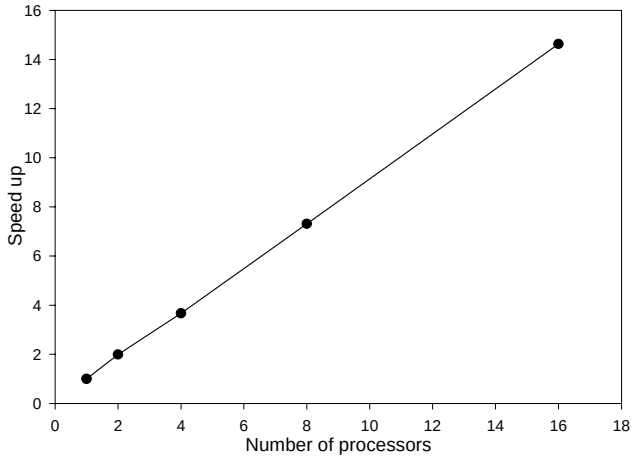

Figure 5. Scaling processors for a set of 128 scatter patterns.

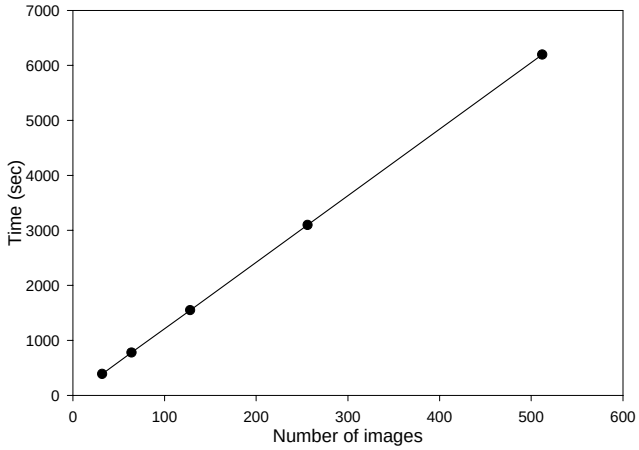

Figure 6. Scaling scatter images for 16 processors.

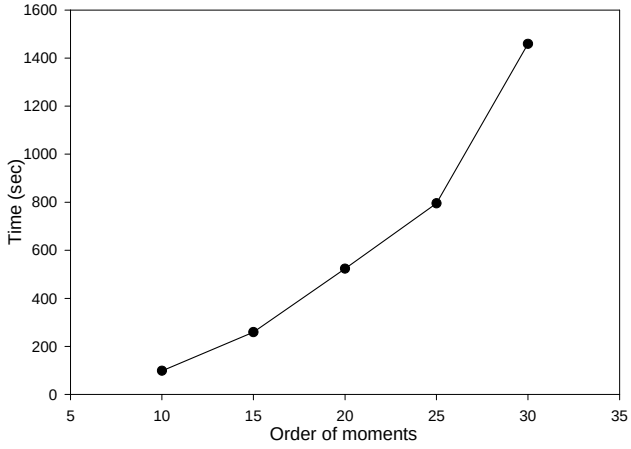

Figure 7. Scaling order of Zernike moments for 16 processors and 64 scatter patterns.

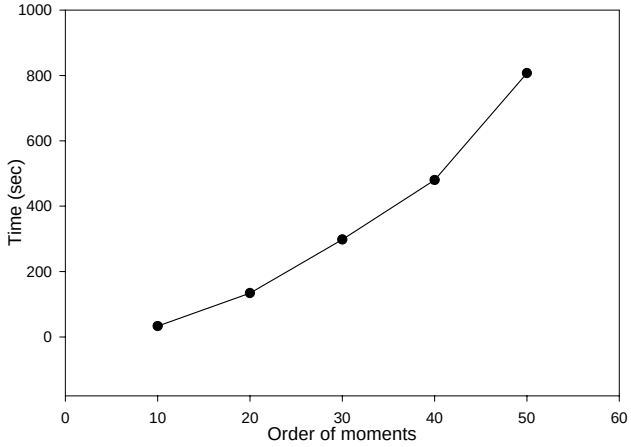

Figure 8. Scaling order of Chebyshev moments for 16 processors and 64 scatter patterns.

C. Classification results

To demonstrate practical application of the forward-scatter analysis technique a set of 2234 scatter images representing various strains of *E. coli*, *Listeria*, *Salmonella*, *Staphylococcus*, and *Vibrio* was used. Feature extraction was performed using 16 processors and the job was completed in less than 8 hours. Zernike moments up to order 20, Chebyshev features up to order 16, and Haralick features with distance 20 were calculated. This resulted in a feature vector of size 496 (240 features for Zernike moments, 136 features for radial Chebyshev moments, and 120 features for 5 different Haralick distances). Feature extraction was followed by feature selection using Fisher's criterion. Sets of 30-240 most discriminative features were selected for construction of an SVM-based classifier. The classification success was estimated using 5x2 cross validation: the data set is permutated randomly, and is divided into 2 disjoint sets (a "training" set and a "test" set). The process in repeated 5 times. In every iteration, the "test" set is used to estimate the classification success (sensitivity and accuracy) of the classifier trained on the "training" set. Although the sensitivity of the method for all the tests species was above 80%, for the sake of brevity we present classification results only for *Salmonella* and *Staphylococcus spp*. Table III shows the classification results for *Salmonella spp*. obtained with the SVM system. The F-score representing the classification success varies from around 0.84 to 0.98. Fig. 9 shows the canonical plot for different strains of *Salmonella*. One may note that *S. enteriditis* PT4 and *S. enteriditis* 13096 are the two classes most difficult to separate by a linear discriminant classifier. Table IV shows the classification results for *Staphylococcus spp*. The F-score varies from around 0.80 to 1.00. *Staph. aureus ATCC13301* showed the lowest classification accuracy while *Staph. aureus PS103* was always correctly recognized. Fig. 10 shows the canonical plot for different species of *Staphylococcus*. We observe that *Staph. aureus ATCC13301* and *Staph. epidermidis ATCC35547* are the two classes most difficult to separate.

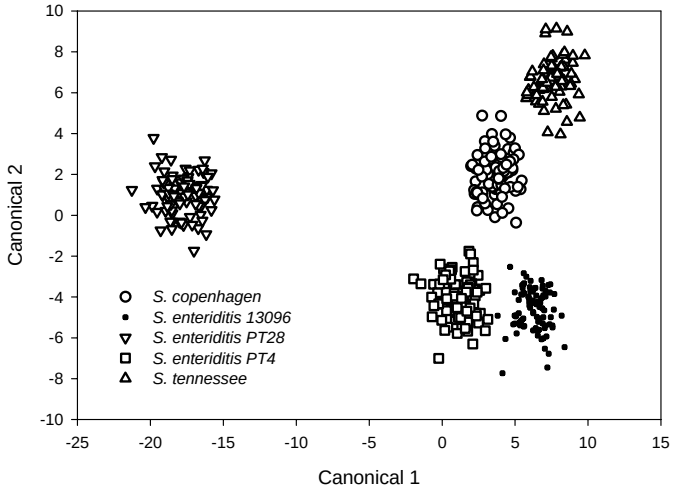

Figure 9. Canonical plot showing separation of classes in linear space. *S. enteriditis PT4* and *S. enteriditis 13096* are the two classes most difficult to separate.

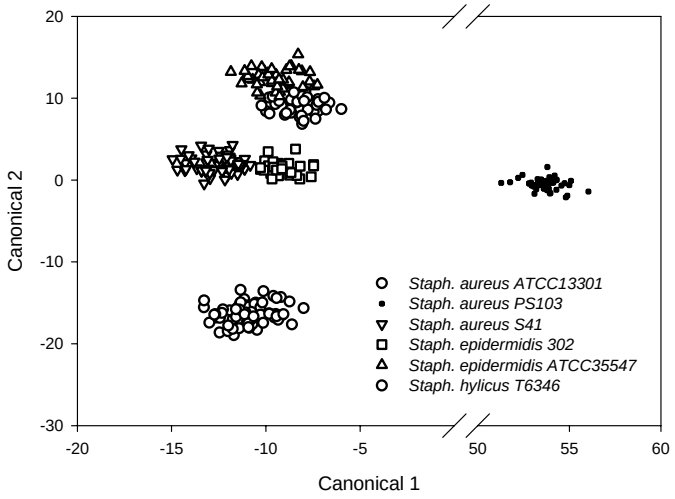

Figure 10. Canonical plot showing separation of classes in linear space. *Staph. aureus ATCC13301* and *Staph. epidermidis ATCC35547* are the two classes most difficult to separate.

VI. CONCLUSION

In this paper we present an application of grid computing for rapid detection and classification of bacterial contamination. We used computational grid technology for meeting the enormous processing requirements of the bacterial scatter detection technology. This new technique for detection and classification of bacterial contamination has a high computational cost for the feature extraction process, which severely impacts the training and optimization procedures. The use of a computational grid provides an efficient and cost-effective solution for the rapid training required for bacterial identification. We report the speed-ups achieved and the scaling behavior of our implementation. A total of 2234 scatter patterns were used for training the classifier. This load was run on 16 computers and the feature extraction was completed in less than 8 hours. The same task took more than 85 hours on a desktop computer having an AMD Athlon 64 processor running at 2.21 GHz with 2 GB of memory. The approach shows nice scalability properties in number of machines and size of the data set, which suggests that a higher number of processors will result in a proportional improvement in execution time. Grid technology is an economic solution for the high computational cost of bacterial scatter detection technique and makes this new technique a feasible and practical approach for rapid detection and classification of bacterial contamination.

TABLE III. CLASSIFICATION SUCCESS FOR SALMONELLA SPP

Salmonella spp.	Sensitivity	Accuracy	F-score
S. copenhagen	0.8105	0.9145	0.86
S. enteriditis 13096	0.8809	0.9311	0.90
S. enteriditis PT28	1	0.9719	0.98
S. enteriditis PT4	0.9404	0.8451	0.89
S. tennessee	0.859	0.8333	0.84

TABLE IV. CLASSIFICATION RESULTS FOR STAPHYLOCOCCUS SPP

Staphylococcus spp.	Sensitivity	Accuracy	F-score
Staph. aureus S41	0.9522	0.9438	0.95
Staph. hylicus T6346	0.9855	0.9714	0.98
Staph. aureus PS103	1	1	1
Staph. aureus ATCC13301	0.8087	0.7983	0.80
Staph. epi. 302	0.8968	0.9521	0.92
Staph. epi. ATCC35547	0.8222	0.8296	0.82

REFERENCES

[1] G. Corkidi, R. Diaz-Uribe, J. L. Folch-Mallol, J. Nieto-Sotelo, "COVASIAM: an image analysis method that allows detection of confluent microbial colonies and colonies of vaiour sizes for automated counting," Appl Environ Microbiol, 64(4):1400-1404.

[2] I. G. Wilson, "Use of the IUL countermat automatic colony counter for spiral plated total viable counts," Appl. Environ Microbiol, 61(8):3158-3160, 1995.

[3] B. Bayraktar, P.P. Banada, E. D. Hirleman, A. K. Bhunia, J. P. Robinson, B. Rajwa, "Feature extraction from light-scatter patterns of Listeria colonies for identification and classification," J Biomed Opt, 11(3):34006, 2006.

[4] Banada PP, Guo S, Bayraktar B, Bae E, Rajwa B, Robinson JP, Hirleman ED, Bhunia AK. Optical forward-scattering for detection of Listeria monocytogenes and other Listeria species. Biosens Bioelectron. 2007 Mar 15;22(8):1664-71.

[5] I. Foster, C. Kesselman, "The Grid: Blueprint for a new computing infrastructure," Morgan Kaufmann Publishers, San Francisco, California, 1999.

[6] I. Foster, C. Kesselman, "Globus: A metacomputing infrastructure toolkit," International Journal of Supercomputing Applications, 11(2):115-128, 1997.

[7] Condor project homepage, http://www.cs.wisc.edu/condor.

[8] N. Jacq, C. Blanchet, C. Combet, E. Cornillot, L. Duret, K. Kurata, H. Nakamura, T. Silvestre, V. Breton, "Grid as a bioinformatic tool," Parallel Computing, 30:1093-1107, 2004.

[9] A. natrajan, M. Crowley, N. Wilkins-Diehr, M. A. Humphrey, A. D. Fox, A. S. Grimshaw, C. L. Brooks III, "Studying protein folding on the grid: experiences using CHARMM on NPACI resources under legion," 10th IEEE International Symposium on High Performance Distributed Computing, pp. 14-21, 2001.

[10] W. E. Johnston, "Computational and data grids in large-scale science and engineering," Future Generation Computer Systems 18, pp. 1085-1100, 2002.

[11] A. YarKhan, J. J. Dongarra, "Biological sequence alignment on the computational grid using the GrADS framework," Future Generation Computer Systems 21, 980-986, 2005

[12] Hu, M. Visual-Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory 8, 179-& (1962).

[13] A. Kotanzad, Y. H. Hong, "Invariant image recognition by Zernike moments," IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(5):489-497, 1990.

[14] R. Mukundan, S. H. Ong, P. A. Lee, "Image analysis by Tchebichef moments," IEEE Transactions on Image Processing, 10(9):1357-1364, 2001.

[15] R. Mukundan, "A new class of rotational invariants using discrete orthogonal moments," in 6th IASTED Conference on Signal and Image Processing, pp. 80-84, 2004.

[16] R. M. Haralick, K. Shanmuga, I. Dinstein, "Textural features for image classification," IEEE Trans Sys Man Cybern, 3(6):610-621, 1973.

[17] Haralick, R. M. and Shapiro, L. G. Computer and Robot Vision. Vol. I. Addison-Wesley Publishing Company. 1992.

[18] A. Blum, P. Langley, "Selection of relevant feature and examples in machine learing," Artificial Intelligence, 97, 245-271.

[19] I. Guyon, A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research 3 (2003) 1157-1182.

[20] R. Kohavi, G. John, "Wrappers for feature selection," Artificial Intelligence, 97(1-2):273-324, December 1997.

[21] R. A. Fisher, "The use of multiple measurement in taxonomic problems," Annals of Eugenics, Vol. 7, part II, pp. 179-188, 1936.

[22] A. K. Jain, R. P. W. Duin, Jianchang Mao, "Statistical pattern recognition: a review," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, January 2000.

[23] C. Cortes, V. Vapnik, "Support vector networks," Mach Learning, 20(3):273-293, 1995.