Journal of Computer and Communications, 2015, 3, 54-62
Published Online May 2015 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2015.35007

Scientific Research Publishing

Parallel Cascade Correlation Neural Network Methods for 3D Facial Recognition: A Preliminary Study

Sokyna M. Al-Qatawneh, Khalid Mohammad Jaber

Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman, Jordan
Email: S.Qatawneh@zuj.edu.jo, drkjaber@yahoo.com/k.jaber@zuj.edu.jo

Received March 2015

Abstract

This paper explores the possibility of using multi-core programming model that implements the Cascade correlation neural networks technique (CCNNs), to enhance the classification phase of 3D facial recognition system, after extracting robust and distinguishable features. This research provides a comprehensive summary of the 3D facial recognition systems, as well as the state-of-the-art for the Parallel Cascade Correlation Neural Networks methods (PCCNNs). Moreover, it highlights the lack of literature that combined between distributed and shared memory model which leads to novel possibility of taking advantage of the strengths of both approaches in order to construct an efficient parallel computing system for 3D facial recognition.

Keywords

Parallel Computing, CCNNs, 3D Facial Recognition, MPI, GPGPU, Multicore

1. Introduction

The simultaneous execution of the same task on multi-processors in order to obtain results quickly is known as parallel computing [1]. The advantage of this method is that a task is divided into many sub-tasks and processed by more than one processor simultaneously, resulting in a significant reduction in the response time. Parallel computing plays a very critical role in many application areas, especially those which require processing large amount of data, such as image processing, filtering, data visualization, and segmentation [2]. On the other hand, automatic 3D pattern recognition is considered as a very hard to solve problem due to its non-linearity [3]. In particular, it is presented as a template matching challenge, where recognition should be performed in a high-dimensional space [3]. Therefore more computation is needed to find a match, which can be sorted by using a dimensional reduction technique to project the problem into a lower dimensionality space. Particularly, facial recognition based on 3D information is relatively new in terms of literature, algorithms, commercial applications, and datasets used for experimentation [3]. Besides that, comparing different 3D facial recognition techniques is very challenging for a number of reasons. Firstly, there are very few standardized 3D facial databases which are used for benchmarking purposes. Secondly, there are differences in the experimental setups and in the metrics, which are used to evaluate the performances of pattern recognition techniques [3]. The Cascade Corre-

How to cite this paper: Al-Qatawneh, S.M. and Jaber, K.M. (2015) Parallel Cascade Correlation Neural Network Methods for 3D Facial Recognition: A Preliminary Study. *Journal of Computer and Communications*, **3**, 54-62.
http://dx.doi.org/10.4236/jcc.2015.35007

lation Neural Networks (CCNNs) can be considered as a good solution to the pattern recognition computation problems; one of its advantages is that they can reduce misclassifications among the neighborhood classes.

The rest of the paper is organized as follows: Section 2 looks at the state of the art and background of 3D features extraction and facial recognition. In Section 3 we present the approaches of parallel processing systems and its related work with facial recognition systems. A detailed literature on cascade correlation neural networks and its application with parallel systems are introduced in section 4. Section 5 presents the summary of this research with future directions for our work.

2. 3D Facial Recognition Studies

According to Al-Qatawneh [3], there is a limit in 3D facial recognition researches in comparison with the wealth of 2D facial recognition researches, regardless of that there are a number of investigations which have demonstrated how geometric pattern structure could be used to aid recognition [3]. For instance, 3D facial recognition has attracted more attention in recent years due to two major factors. Firstly, the inherent problems with 2D facial recognition systems which appear to be very sensitive to facial pose variation, variant facial expressions, lighting and illumination [3]. On the other hand, for example, Xu *et al.* [4] compared 2D intensity image against depth images and concluded that depth map give a more robust face representation, because intensity images are significantly affected by changes in illumination [3]. Secondly, the recent developments in 3D acquisition techniques, such as 3D scanners, use of infrared and other technologies which have made obtaining 3D data much easier than it was before. For 3D face recognition applications, two main requirements have to be met. The first requirement is to provide a powerful representation modelling technique for 3D facial image. The second is to provide a matching algorithm or criterion to recognize and distinguish between these models. While the second requirement has been subject to extensive investigation and research, the first requirement is still considered an open research area [3]. In a paper presented by Bowyer *et al.* [5] covered this topic in detail by presenting a comparative survey of 3D face recognition algorithms. They concluded that 3D face recognition has the potential to overcome limitations of its 2D counterpart. In particular, 3D shape data of a face could be used to correct the corresponding 2D facial image, taken with a non-standard pose, to a standard pose [3].

As it explained in our previous work [3] Nagamine *et al.* [6] tackled face recognition by exploring facial profiles. They used horizontal section (extracted as an intersection of a face surface with a plane parallel to X-Z plane), vertical section (extracted as an intersection of a face surface with a plane parallel to Y-Z plane) and circular cross section (extracted as an intersection of a face surface with a cylinder (axis on Y-Z plane and parallel to Z-axis)). They extracted five feature points, and used them to standardize the face pose. For comparison between faces, Euclidean distance matching between feature vectors of different faces was used. It was concluded that vertical profiles that pass through the central region of the face give better recognition rates, circular sections which cross near the eyes and part of the nose also show some distinctiveness, while the distinctiveness of the horizontal profiles are not remarkable in themselves [3].

In addition, Hasher *et al.* [7] [8] used PCA and Independent Component Analysis (ICA) to analyze range images in a similar way to 2D intensity images and estimated probability models for the coefficients. For registration, and pose standardization they used the nose tip and the nose bridge [3]. They used a database of 37 individuals with images of 6 different facial expressions for each [7].

Elyan and Ugail [9] presented a method to determine the symmetry profile of the face. They computed the intersection between the symmetry plane and the facial mesh and then computed a few feature points along the symmetry profile in order to allocate the central region of the face and extract a set of profiles from that region. In this approach, they assume that the symmetry profile passes through the tip of the nose. To locate the tip of the nose they fit a bilinear blended Coon's surface patch. Coon's patch is simply a parametric surface defined by four given boundary curves [9]. These four boundaries of the Coon's patch are determined based on a boundary curve that encloses an approximated central region of interest, which is simply the region of the face that contains or is likely to contain the nose area [3].

Considering all this prior work which has been done, [3] asserted that there still remain a number of areas that 3D face recognition research needs to address. For registration, automatic landmark localization, artefact removal, scaling, and elimination of errors due to occlusions, glasses, beard, etc. need to be worked out. Additionally, methods of deforming the face without the loss of discriminative information would also be beneficial It is likely that information fusion is the future of 3D face recognition [3]. There are many ways of representing and

combining texture and shape information. Publicly available 3D datasets are necessary to encourage further research on these topics [3].

Table 1 gives a comparison of selected elements of algorithms that use 3D facial data to recognize faces.

3. Parallel Processing Studies

The meaning of parallel computing is when in order to obtain results quickly there is a simultaneous execution of the same task on multiple processors [1]. The benefit of this method is that a task is divided into many subtasks and processed by more than one processor simultaneously, resulting in a significant reduction in the response time. Parallel computing plays a very critical role in many application areas, especially those which require processing large amount of data, such as weather forecast, data visualization, biology and engineering [29].

Parallel processing computers can be classified in many perspectives; such as implicit and explicit parallelism. When referring to implicit parallelism we mean that it is a built-in programming approach that is incorporated within parallel language and parallelizing compilers, it does not specify or control scheduling of calculations. Whereas the responsibility of explicit parallelism is on the programmer, with tasks such as task decomposition, synchronization, communication and so on. As another classification is based on instruction streams and data streams that were proposed by Flynn in 1996. Flynn proposed a four-way classification of parallel computers that are SISD (Single Instruction Single Data), SIMD (Single Instruction multiple Data), MISD (Multiple Instruction Single Data), and MIMD (Multiple Instruction Multiple Data) [1].

In 1988 E. E. Johnson proposed a new taxonomy which was based on memory structure such as shared/global memory or distributed memory [1]. Communications are totally involved between processors in parallel computation and the mechanism which are used for communication/synchronization and is referred to as message passing. Fortunately, many message-passing libraries have been developed to provide routines to initiate and configure the messaging environment as well as to send/receive packets of data between processors. The two most popular message-passing libraries are Parallel Virtual Machine (PVM) [30] and Message Passing Interface (MPI) [30], while the most popular routines as shared address space paradigms are the POSIX Thread [30] and OpenMP [30] as illustrated in **Figure 1**.

Recently, researchers tried to use General Purpose computation on Graphics Processing Units (GPGPU) as parallel programming approach. GPGPU are techniques to program GPU chips using application programming interface (API) functions such as OpenGL, Direct3D and CUDA [31] and are used in order to obtain results quickly. However, Graphics Processing Units (GPUs) are highly threaded streaming multiprocessors of very high computation and data throughput [32]. In 2006, CUDA (Compute Unified Device Architecture) was created by NVIDIA which is a parallel computing platform and programming model and implemented by the graphics processing units (GPUs). CUDA has been widely deployed through thousands of applications and published research papers such as astronomy, biology, chemistry, physics, and data mining. Supported by an installed base of over 300 million CUDA-enabled GPUs in notebooks, workstations, compute clusters and super

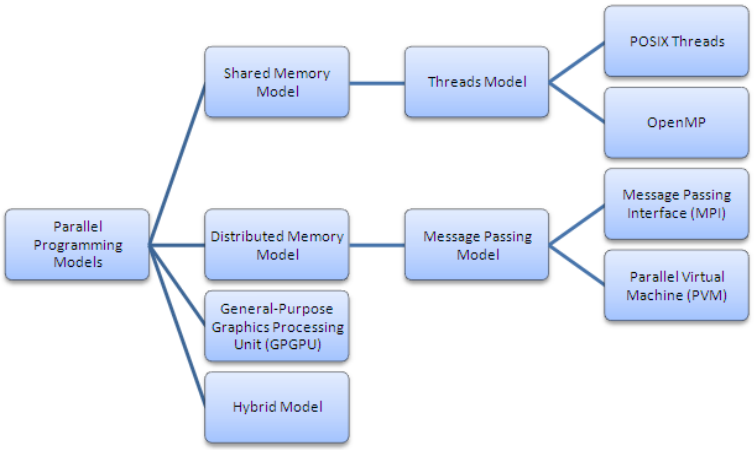

Figure 1. Parallel programming models.

Table 1. Summary of recognition algorithms using 3D facial data.

Reference	Database Size	Data Representation	Algorithm	Reported Performance
Cartoux *et al.* [10], 1989	18	Profiles, surface	Curvature based Nearest Neighbor	100%
Lee *et al.* [11], 1990	6	EGI	Correlation	N/A
Gordon, [12], 1992	26	curvatures	Euclidean nearest neighbor	100%
Nagamine *et al.* [6], 1992	160	Multiple profiles	Euclidean nearest neighbor	100%
Tanaka *et al.* [13], 1998	37	curvatures based EGI	Fisher's spherical Correlation	100%
Zhao and Chellappa [14], 2000	N/A	3D model + Texture	Produce a prototype images	81%
Beumier *et al.* [15], 2001	120	2D and 3D vertical profiles	Minimum distance, fusion	1.4% EER
Wang *et al.* [16], 2002	300	Feature vector point signature Gabor features	PCA + SVM	>90%
Chang *et al.* [17], 2003	278	Texture + Range image	PCA	99% 3D + 2D, 93% 3D only
Hesher *et al.* [8], 2003	222	Range image	ICA or PCA, nearest neighbor	97%
Moreno *et al.* [18], 2003	420	Curvature, line, region features	Euclidean nearest neighbor	78%
Pan et al [19], 2003	360	Point set, range image	Hausdorff and PCA	3% - 5% EER
Chang *et al.* [20], 2003	278	Texture+ Range image	PCA	99% 3D + 2D, 93% 3D only
Tsalakanidou, [21] 2003	80	Range image	PCA	99%
Xu *et al.* [4], 2004	30 &120	Point set + feature vector	Minimum distance	96% on 30, 72% on 120
Lu *et al.* [22], 2005	196	Surface mesh	ICP, TPS	89%
Lee *et al.*[23], 2005	70	depth map	Feature extraction, nearest neighbor	94%
Bronstein *et al.*[24], 2005	220	Point set	Canonical forms	100%
Lee *et al.* [25], 2005	200	Feature vector	SVM	96%
Pan *et al.* [26], 2005	720	Range image	PCA	95%
Zhang *et al.* [27], 2006	32	Range images	mean curvature	96.9%
Qatawnah *et al.* [28], 2008	56	Range images	Feature extraction, CCNN, SVM and KNN	100%
Elyan and Ugail [19], 2009	144	triangle mesh	Coon's surface patch	86.9%

computers [33].

However, there are several mechanisms to parallelize large amounts of data; function decomposition, data decomposition or both. Data decomposition consists of horizontal distribution and vertical distribution or both. Moreover, hybrid parallelism merges both data parallelism with horizontal or vertical distribution and function decomposition [1] as shown in **Figure 2**. The one that is probably the most important programming language in computer science used to analyze data, develop algorithms, and create models and applications is a MATLAB [34]. Over recent times, MALTAB has gained much popularity and has been applied in many fields including image processing, bioinformatics, engineering, medical, signal processing, communications and parallel computing. Additionally, MATLAB has parallel computing built-in functions that lets the researcher to solve computationally and data-intensive problems using multi-core processors, GPUs, and computer clusters [34].

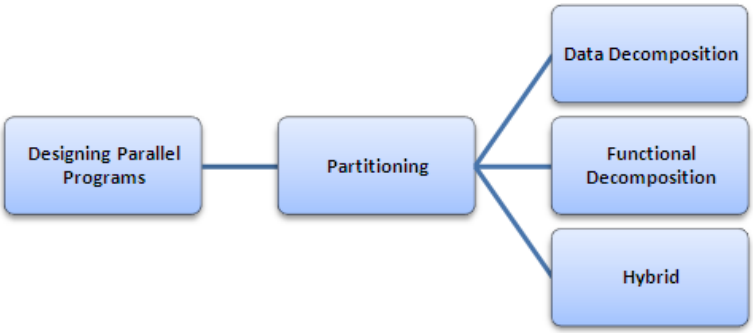

Figure 2. A mechanism to parallelize large amounts of data.

MATLAB multicore programming is implemented based on threads. Threads are lightweight processes; and it is very much easier to write threaded programs because the applications are threaded and run on a single machine, can also run on multiple machines without changes. This ability of migrating programs between different platforms is of great benefit to threaded APIs. Additionally, threads which run on the same processor reduce the latency of accessing memory, I/O and communications.

This can be done by a different number of ways such as it can measure the parallel performance of a given application such as Amdahl's law, speedup (S), Efficiency (E) and Overhead. The time of execution is defined by the serial runtime of a program and is the time elapsed between the beginning and the end of its execution on a sequential machine (Ts). The Parallel run time (Tp) is the time which elapses from the moment a parallel computation starts to the moment the last processor finishes execution. Overhead function (To) or total overhead of a parallel system is the total time collectively spent by all processing elements over and above that required by the fastest known sequential algorithm for solving the same problem on a single processing element; as given by Equation (1):

$$\text{Overhead}(To) = p*T\ p\text{-}Ts \tag{1}$$

The measure that captures the relative benefit of solving the problem in parallel is called Speedup (S). It is defined as the ratio of the time taken to solve a problem on a single processing element to the time required to solve the same problem on a parallel computer with p identical processing elements; given by Equation (2):

$$\text{Speedup}(S) = Ts/T\ p \tag{2}$$

The Efficiency (E) is a measure of the fraction of time for which a processing element is usefully employed; it is defined as the ratio of speedup to the number of processing elements; as given by Equation (3):

$$\text{Efficiency}(E) = S/P \tag{3}$$

Parallel Neural Networks for Facial Recognition

The equipment and ability to use automatic systems for face recognition research has emerged over the last few decades. One reason for growing interest in this topic is the wide range of possible applications for face recognition systems. There is the possibility of training neural networks but it is time consuming task. Therefore, substantial efforts are dedicated to cope with the training time in different fields on different hardware architectures. Such as, Altaf *et al.* [35] proposed two techniques of parallelizing back propagation neural networks using Multicore programming based on multithreading and general-purpose computation on graphics processing units (GPGPU). While Xavier *et al.* [36] presented the parallel training of a back-propagation neural network using CUDA on GPU. The implementation was tested with two standard benchmark data sets which provided by PROBEN1. The two parallelization strategies on a cluster computer; training example and node parallelism using MPI approach was presented by Pethick *et al.* [37].

Lyle *et al.* [38] proposed the scalable massively parallel artificial neural networks for pattern recognition application. In this approach, the MPI used to parallelize the C++ code. However, each layer is distributed equally over all processors, which mean they used the data decomposition approach to parallelize the algorithm. There are further versions of parallel neural network which uses the task decomposition paradigm for cluster system where they duplicates the full neural network at each cluster node was presented by Dahl *et al.* [39]. Their sys-

tem was implemented using MPI library. Schuessler *et al.* [40] proposed a method for parallelization of neural network training based on the backpropagation algorithm and implemented it using two different multithreading techniques (OpenMP and POSIX threads) applicable to the current and next generation of multithreaded and multi-core CPUs.

4. Cascade Correlation Neural Networks

The training of back-propagation neural networks is considered to be a slow process because of the step-size and moving target problems [3] [41]. To overcome these problems cascade correlation neural networks were developed. These are "self organizing" networks [3] [41] with topologies which are not fixed. The supervised training begins with a minimal network topology and new hidden nodes are incrementally added to create a multi-layer construction. The new hidden nodes are added to make the most of the correlation between the new node's output and the remaining error signal that the system is being adjusted to eliminate. The weights of a new hidden node is fixed and not changed later, hence making it a permanent feature detector in the network. This feature detector can then be used to generate outputs or to create other more complex feature detectors [3] [41].

In a CCNN, the number of input nodes is determined by the input features, while the number of output nodes is determined by the number of different output classes [3]. The training of a CCNN starts with no hidden nodes. The direct input-output connections are trained using the entire training set with the aid of the back propagation learning algorithm [3]. Hidden nodes are then added gradually and every new node is connected to every input node and to every pre-existing hidden node. The goal of this adjustment is to maximize S, the sum overall output units o of the magnitude of the correlation1 between V, the candidate unit's value, and E_o, the residual output error observed at unit o. S can be defined as:

$$S = \sum_o \left| \sum_p \left(V_p - \overline{V} \right) \left(E_{p,o} - \overline{E_o} \right) \right| \tag{4}$$

where *o* is the network output at which the error is measured and p is the training pattern. The quantities $\overline{V}$ and $\overline{E_o}$ are the values of V and E_o averaged over all patterns.

Training is carried out using the training vector and the weights of the new hidden nodes are adjusted after each pass [41]. Cascade correlation networks have a number of attractive features including a very fast training time, often a hundred times faster than a perceptron network [41]. This makes cascade correlation networks suitable for use with large training sets.

Depending on the application and number of input nodes, cascade correlation networks are fairly small, often having fewer than a dozen neurons in the hidden layer [42] [43]. This can be contrasted with probabilistic neural networks which require a hidden-layer neuron for each training case. Also, the training of CCNNs is quite robust, and good results can usually be obtained with little or no adjustment of parameters [41].

Parallel Cascade Correlation Neural Networks

As mentioned before, Fahlman *et al.* [41] claim that Cascade Correlation algorithm is attractive for parallel implementation because the candidate units do not need to interact which means it can be trained independently. Few previous efforts have been made to enhance cascading correlation neural networks. Few research efforts have reported parallel cascading correlation neural network in different fields. David German [31] proposed a project proposal titled computing hardware for accelerated training of cascade-correlation neural networks by parallelization of multiply-accumulate operations implemented on a field-programmable gate array (FPGA) in communication with a host PC. However, the author was not clear enough and did not mention the parallel technique in details. Moreover, the authors did not mention their benchmark to evaluate their method such as speedup, overhead,..etc.

Moreover, Ingrid *et al.* [44] proposed the parallel training data of recurrent cascade correlation learning architecture (RCC) to recognize Japanese phonemes using a method called time-slicing. However, it was intended by the authors that in parallel RCC there will be a large number of training patterns or the training set should be divided into smaller chunks which are to be trained separately and sequentially, this is done in such a pattern that it goes from the simplest to the most complicated one. Therefore, the authors did not use the standard parallel computing concepts and models such as shard memory, distributed memory. Furthermore, they did not use parallel programming approaches such as multithreading, MPI or GPGPU.

5. Summary and Future Work

The purpose of this paper was to propose a brief assessment for existing 3D facial recognition techniques. As well as, highlight the potential of using parallel CCNNs methods for 3D facial recognition systems. The automatic processing and characterization of 3D scanned images is still considered a challenging problem with a relatively few papers in the public domain addressing it. Semi automatic approaches, where initial assumptions are made about the pose of the face are often encountered to simplify the problem. On the other hand, extracting facial features points is considered to be an essential stage in any facial recognition systems, which can be utilized to allocate the central region of the face and extract a set of effective profiles from that region in order to use them effectively with neural networks algorithms for recognition and classification purpose. The neural networks have been applied in many fields such as face recognition, speech recognition, and pattern recognition. Parallel neural network was implemented using different parallel techniques such as GPGPU [35] [36], MPI [37]-[39] and multithreading [40]. However, only two-research effort have reported parallel cascading correlation neural network in different fields using FPGA. From the review, the authors noted the previous works did not implement the algorithm using stranded parallel approaches. Moreover, there is not any work hybrid between distributed and shared memory model or used both of GPU and CPU methods. Some areas of future work include develop and implement a compact representation of facial data by reconstructing a 3D triangulated human face containing the coordinate and connectivity information to simplify the process of recognition in order to extract robust and distinguishable features. After that we will propose an efficient parallel computing system for 3D facial recognition using a multi-core programming model that implements the cascade correlation neural network technique (CCNN), which is widely recognized as appropriate and efficient validation methods as explained before.

Acknowledgements

The authors would like to thank Al-Zaytoonah University of Jordan for funding this project (Grant Number: 2/25/2014).

References

[1] Jordan, L.E. and Gita, A. (2002) Fundamentals of Parallel Processing. Prentice Hall Professional Technical Reference.

[2] Khalid Mohammad, J., Rosni, A. and Nur'Aini Abdul, R. (2014) Fast Decision Tree-Based Method to Index Large DNA-Protein Sequence Databases Using Hybrid Distributed-Shared Memory Programming Model. *Int. J. Bioinformatics Res. Appl.*, **10**, 321-340.

[3] Al-Qatawneh, S. (2012) 3D Facial Feature Extraction and Recognition. LAP Lambert Academic Publishing.

[4] Xu, C.H., Wang, Y.H., Tan, T.N. and Quan, L. (2004) Depth vs. Intensity: Which Is More Important for Face Recognition. 17*th International Conference on Pattern Recognition*, **1**, 342-345.

[5] Bowyer, K.W., Chang, K. and Flynn, P. (2006) A Survey of Approaches and Challenges in 3D and Multi-Modal 3D + 2D Face Recognition. *Computer Vision and Image Understanding*, **101**, 1-15. http://dx.doi.org/10.1016/j.cviu.2005.05.005

[6] Nagamine, T., Uemura, T. and Masuda, I. (1992) 3D Facial Image Analysis for Human Identification. *International Conference on Pattern Recognition*, 324-327.

[7] Hesher, C., Srivastava, A. and Erlebacher, G. (2003) A Novel Technique for Face Recognition Using Range Imaging. *International Symposium on Signal Processing and Its Applications*. http://dx.doi.org/10.1109/ISSPA.2003.1224850

[8] Hesher, C.A.S. and Erlebacher, G. (2002) PCA of Range Images for Facial Recognition. In: *Proceedings of International Multiconference in Computer Science*, Las Vegas.

[9] Elyan, E. and Ugail, H. (2009) Automatic 3D Face Recognition Using Fourier Descriptors. In: *International Conference on CyberWorlds*, IEEE Computer Society, Bradford, UK.

[10] Cartoux, J.Y., Lapreste, J.T. and Richetin, M. (1989) Face Authentification or Recognition by Profile Extraction from Range Images. *IEEE Computer Society Workshop on Interpretation of* 3*D Scenes*, 194-199.

[11] Lee, J.C. and Milios, E. (1990) Matching Range Images of Human Faces. *IEEE International Conference on Computer Vision*, Osaka, Japan.

[12] Gordon, G. (1992) Face Recognition Based on Depth and Curvature Features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. http://dx.doi.org/10.1109/CVPR.1992.223253

[13] Tanaka, H., Ikeda, M. and Chiaki, H. (1998) Curvature-Based Surface Recognition Using Spherical Correlation Principal Directions for Curved Object Recognition. *Third IEEE International Conference on Automatic Face and Gesture Recognition*. http://dx.doi.org/10.1109/AFGR.1998.670977

[14] Chellappa, R. and Zhao, W.Y. (2000) Illumination-Insensitive Face Recognition Using Symmetric Shape-from-Shading. *IEEE International Conference on Computer Vision*.

[15] Beumier, C. and Acheroy, M. (2001) Face Verification from 3D and Grey Level Clues. *Pattern Recognition Letters*, **22**, 1321-1329. http://dx.doi.org/10.1016/S0167-8655(01)00077-0

[16] Y. Wang, Chua C.-S. and Ho, Y.-K. (2002) Facial Features Detection and Face Recognition from 2D and 3D Images. *Pattern Recognition Letters*, **23**, 1191-1202. http://dx.doi.org/10.1016/S0167-8655(02)00066-1

[17] Chang, K.I., Bowyer, K.W. and Flynn, P.J. (2003) Multimodal 2D and 3D Biometrics for Face Recognition. *IEEE International Workshop on Analysis and Modeling of Face and Gestures*, ed. K.W. Bowyer, 187-194.

[18] Moreno, A., Sanchez, A., Velez, J. and Diaz, F. (2003) Face Recognition Using 3D Surface-Extracted Descriptors. *Irish Machine Vision and Image Processing Conference* (*IMVIP*).

[19] Pan, G., Wu, Z. and Pan, Y. (2003) Automatic 3D Face Verification from Range Data. 193-196.

[20] Chang, K.I., Bowyer, K.W. and Flynn, P.J. (2003) Multimodal 2D and 3D Biometrics for Face Recognition. *IEEE International Workshop on Analysis and Modeling of Face and Gestures*, ed. K.W. Bowyer, 187-194.

[21] Tsalakanidou, F., Tzocaras, D. and Strintzis, M. (2003) Use of Depth and Colour Eigenfaces for Face Recognition. *Pattern Recognition Letters*, **24**, 1427-1435. http://dx.doi.org/10.1016/S0167-8655(02)00383-5

[22] Lu, X. and Jain, A.K. (2005) Intergrating Range and Texture Information for 3D Face Recognition. 7*th IEEE Workshop on Applications of Computer Vision*.

[23] Lee, Y., Song, H., Yang, U., Shin, H. and Sohn, K. (2005) Local Feature Based 3D Face Recognition. *International Conference on Audio- and Video-Based Biometric Person Authentication*.

[24] Bronstein, A., Bronstein, M. and Kimmel, R. (2005) Three-Dimensional Face Recognition. *International Journal of Computer Vision*, **64**, 5-30. http://dx.doi.org/10.1007/s11263-005-1085-y

[25] Lee, Y., Song, H., Yang, U., Shin, H. and Sohn, K. (2005) Local Feature Based 3D Face Recognition. *International Conference on Audio- and Video-based Biometric Person Authentication*.

[26] Pan, G., Han, S., Wu, Z. and Wang, Y. (2005) 3D Face Recognition Using Mapped Depth Images. 175.

[27] Zhang, L., Razdan, A., Farin, G., Femiani, J., Bae, M. and Lockwood, C. (2006) 3D Face Authentication and Recognition Based on Bilateral Symmetry Analysis. *The Visual Computer*, **22**, 43-55. http://dx.doi.org/10.1007/s00371-005-0352-9

[28] Qatawnah, S., Ipson, S., Qahwaji, R. and Ugail, H. (2008) 3D Face Recognition Based on Machine Learning. In: *IASTED International Conference on Visualization, Imaging and Image Processing* (*VIIP* 2008), Palma de Mallorca, Spain.

[29] Khalid Mohammad, J., Rosni, A. and Nur'Aini Abdul, R. (2014) Fast Decision Tree-Based Method to Index Large DNA-Protein Sequence Databases Using Hybrid Distributed-Shared Memory Programming Model. *Int. J. Bioinformatics Res. Appl.*, **10**, 321-340.

[30] Claudia, L. (2001) Parallel and Distributed Computing: A Survey of Models, Paradigms and Approaches. John Wiley \\& Sons, Inc.

[31] German, D. (2007) Engineering 90 Project Proposal. Computing Hardware for Accelerated Training of Cascade-Correlation Neural Networks. Dec. 4.

[32] GPGPU. http://gpgpu.org/

[33] Nvidia. http://www.nvidia.com

[34] Sharma, G. and Martin, J. (2009) MATLAB¦®: A Language for Parallel Computing. *International Journal of Parallel Programming*, 3-36. http://dx.doi.org/10.1007/s10766-008-0082-5

[35] Huqqani, A.A., Schikuta, E., Ye, S. and Chen, P. (2013) Multicore and GPU Parallelization of Neural Networks for Face Recognition. *Procedia Computer Science*, **18**, 349-358.

[36] Xavier, S.-C., Francisco, M.-R. and Victor, U.-C. (2010) Parallel Training of a Back-Propagation Neural Network Using CUDA. Book Parallel Training of a Back-Propagation Neural Network Using CUDA, Series Parallel Training of a Back-Propagation Neural Network Using CUDA, ed., IEEE Computer Society.

[37] Mark Pethick, M.L., Werstein, P. and Huang, Z.Y. (2003) Parallelization of a Backpropagation Neural Network on a Cluster Computer. *Proc. the Fifteenth IASTED International Conference on Parallel and Distributed Computing and Systems*, November, 574-582.

[38] Long, L.N. and Gupta, A. (2008) Scalable Massively Parallel Artificial Neural Networks. *Journal of Aerospace Computing, Information, and Communication*, **5**, 1-11.

[39] George, D., Alan, M. and Tia, N. (2008) Parallelizing Neural Network Training for Cluster Systems. In: *Book Parallelizing Neural Network Training for Cluster Systems, Series Parallelizing Neural Network Training for Cluster Systems*, ACTA Press.

[40] Schuessler, O. and Loyola, D. (2011) Parallel Training of Artificial Neural Networks Using Multithreaded and Multicore CPUs. *Adaptive and Natural Computing Algorithms, Lecture Notes in Computer Science* 6593, Springer, Berlin, Heidelberg, 70-79.

[41] Fahlmann, S.E. and Lebiere, C. (1989) Advances in Neural Information Processing System 2(NIPS-2). In: Touretzky, D.S., Ed., Morgan Kaufmann, Denver, 524.

[42] DTREG. http://www.dtreg.com/cascade.htm

[43] Shet, R.N., Lai, K.H., Edirisingh, E. and Chung, P.W.H. (2005) Pattren Recognition and Image Analysis. In: Marques, J.S., de la Pe'rez, B.N. and Pina, P., Eds., *Lecture Notes in Computer Science*, Springer, Berlin.

[44] Ingrid, K., Masafumi, K., Jun-ichi, A. and Hideto, T. (1996) The Time-Sliced Paradigm & Mdash;a Connectionist Method for Continuous Speech Recognition. Inf. Sci., 133-158.