

An Overview of Volume Rendering Techniques for Medical Imaging

<https://doi.org/10.3991/ijoe.v16i06.13627>

Amr S. Mady
Halan Inc., Cairo, Egypt
amrsaedmady@gmail.com

M. Samir Abou El Seoud
The British University, Cairo, Egypt

Abstract—One of the most valuable medical imaging visualizations or computer-aided diagnosis is Volume rendering (VR). This survey's objective is reviewing and comparing between several methods and techniques of VR, for a better and more comprehensive reading and learning of both pros and cons of each method, and their use cases.

Keywords—Volume rendering, Ray-Casting, Image Splatting, Texture Slicing, Cell-Projection (CP), Shear-Warp, Marching Cubes, Projected Tetrahedra, Maximum Intensity Projection, Direct Volume Rendering, Indirect Volume Rendering.

1 Introduction

Several groups and associations consisting of scientists, researchers, and developers have worked on developing numerous VR techniques enhancements, acceleration techniques, and applications. All that work on tiresome is because of the crucial importance of VR in many industrial fields and especially in computer games and medical imaging visualization software. But most of the globally known techniques of VR falls under three famous types, each has its own advantages and disadvantages. These three types of volume rendering techniques will be discussed thoroughly in the next section, namely:

1. Direct Volume Rendering (DVR) technique. It includes Shear Warp, Image Splatting, Texture Slicing, Cell Projection, and Ray-Casting,
2. Indirect Volume Rendering (IVR) technique. It includes marching cubes, and Tetrahedral techniques.
3. Maximum Intensity Projection (MIP) technique.

2 Methods

2.1 Direct volume rendering (DVR) techniques

Image splatting: Shear Warp algorithm in volume rendering is one of the fastest classic methods used in volume rendering [1, 2]. It finds the most aligned samples' axes with the grid of the voxels [3]. Thereafter, it samples each voxel in the data just one time, see (Figure 1). Consequently, it has a very fast computational run time than other techniques. The disadvantages of this technique are:

- I. Not the most accurate method in volume rendering voxel's sampling.
- II. Quality is low.

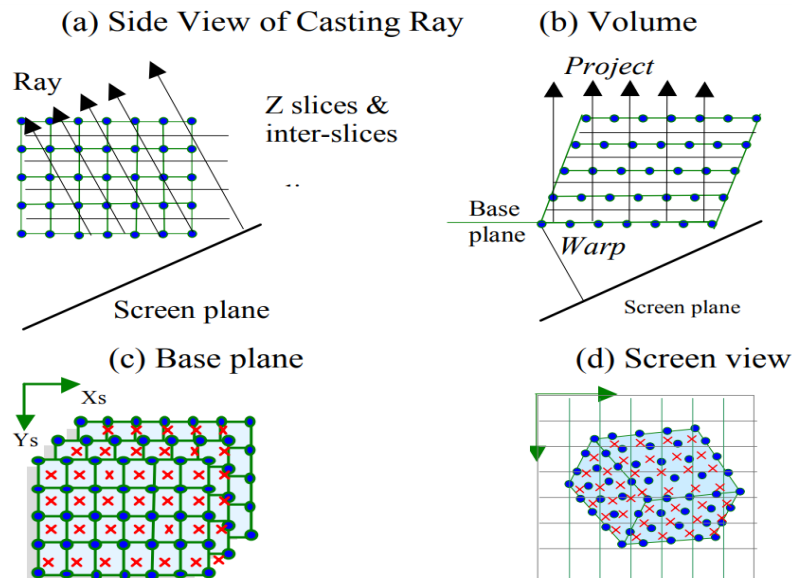


Fig. 1. Shear-Warp illustration, (a) Side View of casting Ray, (b) Volume, (c) Base Plane, (d) Screen View [2].

Image splatting: Image splatting is a very well-known method. It was first discovered by Lee Westover in 1989 [4]. In this method, voxels are rendered by overlapping functions such as Gaussian kernel. In Image Splatting, the volume is a field of 3D interpolation kernel. Each grid voxel has its own kernel, the images produce 2D footprints on the screen, and each 2D footprint is a result of a kernel (see Figure 2). This method is also fast as it neglects the empty volume spaces, and only stores volumetric points from the dataset. However, being doing so, this implementation results in aliasing, blurring, and color bleeding [5-7].

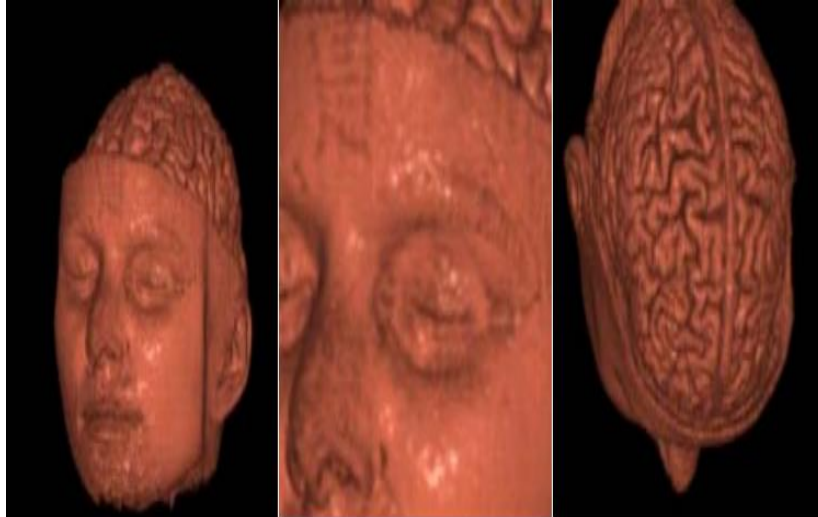


Fig. 2. Examples of image splatting on a full head dataset [3].

Texture slicing: Texture Slicing technique produces high quality volume rendering [8, 9], but at the expense on a much higher computational time. It works as follows:

1. It makes viewport-aligned slices parallel to the plane of the image (see Figure 3)
2. Every time the view of the matrix is updated
3. The whole viewport-aligned slices are recomputed.

Compositing process in this technique is made by Back-To-Front compositing. Consequently, the polygonal slices textures are blended.

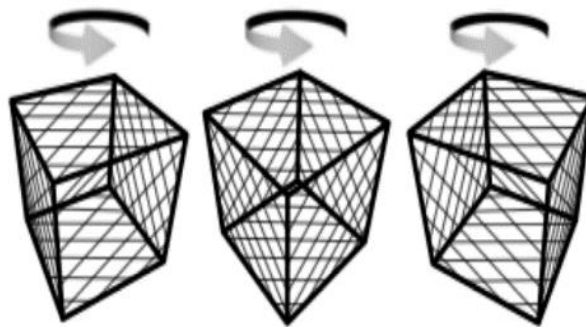


Fig. 3. Texture slicing sampling generating view-aligned slices parallel to the image plane [3].

Cell projection: In the cell-projection (CP) scheme [10], unstructured grid volume information is rendered via the following 3 steps (see Figure 4) as explained in the 3 steps below. For straightforwardness of dialog, we expect the cell to be a tetrahedral:

- I. **Projection Phase:** It's a given three-dimensional data (cell) into a two-dimensional screen and discover the projection area (R) for each data (cell).
- II. **Scan Conversion Phase:** It performs scan conversion for each projected cell. More precisely, for every pixel $P(x, y)$ in the projection area (R) on the screen:

Figure the cell's contribution (color (RGB) and opacity (α)) to pixel $P(x, y)$ and depth values (z_{front} and z_{back}) for both the front and back intersection points where the ray corresponding to pixel $P(x, y)$ intersects the cell.

In this study, the data structure comprising of these four parameters (RGB, α , z_{front} and z_{back}) is referred as ray segment. As a result of the scan conversion of a cell, ray segments for every pixel in the projection area (R) are computed. For every cell, we compute its ray segments. After that, for every pixel on the screen, gather the ray-segments corresponding to the pixel and make a depth-sorted rundown of these ray segments (Figure 5).

- III. **Composition Phase:** Given the ray-segment lists for all pixels on the screen, compute the pixel value (color) by compositing the depth-sorted ray segments in the ray-segment list from front to back by alpha blending using Porter-Duff's over operation. Finally, we get the volume rendered picture (Figure 6).

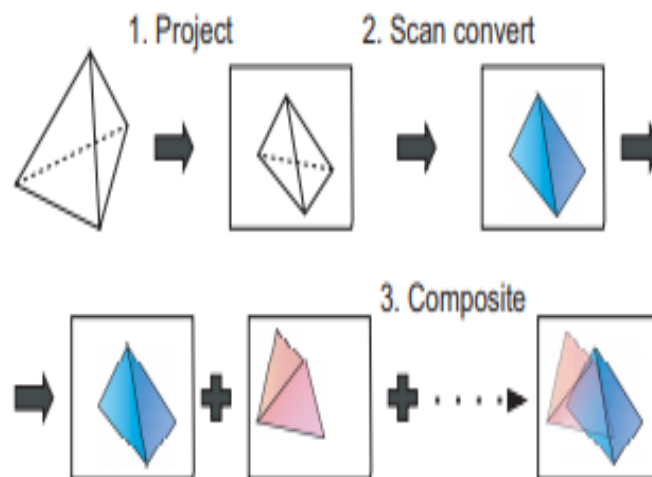


Fig. 4. Cell Projection [10].

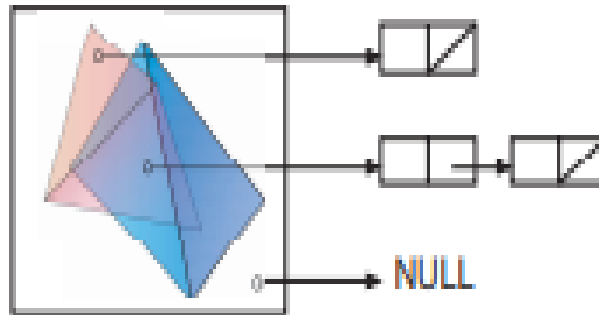


Fig. 5. Ray Segment List [10].

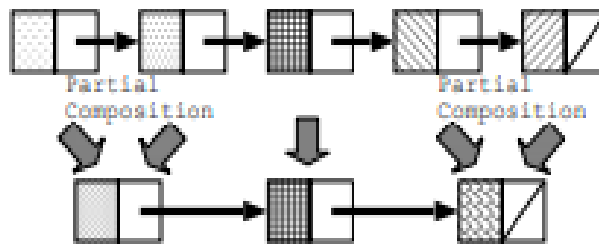


Fig. 6. Partial Composition [10].

Ray casting: Ray casting [1] is considered the best volume rendering technique. It produces the highest rendering quality, and also supports performance optimizations such as space leaping and ray termination in early stages, once it exits the volume. Ray casting works by the basics of the volume rendering pipeline. It first cast visual rays from each pixel through the volume to the end of the view direction in the virtual scene. Thereafter, it starts computing the composition and resampling to accumulate the end value of the pixel once the ray intersects with the volume surface. As mentioned above, the voxels that has zero opacities are stored as transparent pixels, see Figure (7). Briefly, following steps explain how it works:

Step 1:

- Trace from each pixel a ray into object space
- Compute and accumulate color/opacity value along the ray
- Assign the obtained value to the pixel

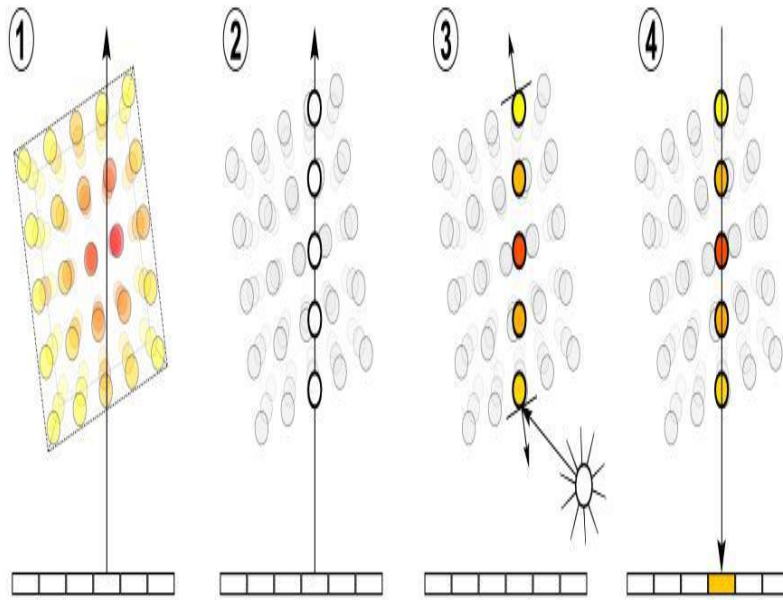


Fig. 7. Ray-marching illustration

Step 2:

Compositing (alpha blending), i.e. the iterative computation of discretized volume integral Figure (8).

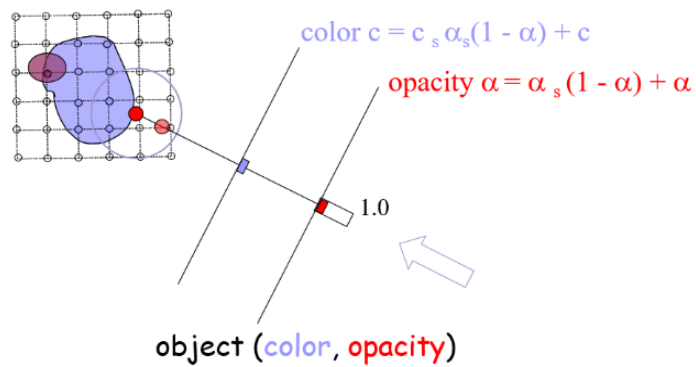


Fig. 8. Compositing of pixels' color/opacity along the ray.

2.2 Indirect volume rendering (IVR) techniques

The goal of the indirect volume rendering techniques or surface rendering (SR) techniques, or iso-surfacing is to create a surface with constant density from a 3D dataset. This will decrease the computation complexity. Different methods were developed for this technique, such as: contour tracing, marching cubes [12], and marching tetrahedra [13]. In the next section, the steps of the marching cubes algorithm will be discussed in more details.

Marching cubes: Marching Cubes (MC) approximates the real Iso-surface of the original dataset used [12]. It approximates surfaces by triangular meshes. The surfaces are found by linear interpolation along cell edges. Afterwards, the normal vectors are used as gradients of the Iso-surface. Briefly, marching cubes technique creates a surface from 3D datasets and the following steps describes this process step by step:

1. Read four slices into memory.
2. Scan two slices and create a cube from four neighbors on one slice and four neighbors on the next slice.
3. Calculate an index for the cube by comparing the eight density values at the cube vertices with the surface constant.
4. Using the index and look up the list of edges from a precalculated table.
5. Using the densities at each edge vertex and find the surface-edge intersection via linear interpolation.
6. Calculate a unit normal at each cube vertex using central differences. Interpolate the normal to each triangle vertex.
7. Output the triangle vertices and vertex normals.

Here is the main drawback of this technique: Because of the numerous numbers of geometric primitives used, the algorithm requires a huge amount of computations to deliver high quality image of the 3D data. On the opposite, a dataset set with small details will deliver low quality rendering, and less accuracy, see Figure (9).

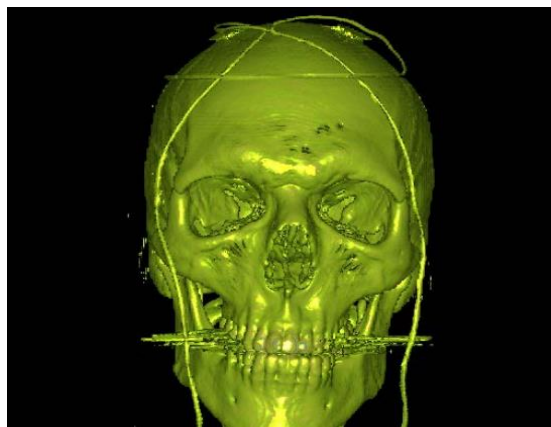


Fig. 9. An Iso-surface surface rendering of a human skull [3].

Projected tetrahedra algorithm: Projected Tetrahedra (PT) [13] is an algorithm that operates with any set of three-dimensional data that has been tetrahedralized and the three-dimensional simple of triangulated data in the plane. Since a large class of data is sampled or computed on a lattice of six-sided cells or cubes, this decomposition is incorporated in the description. The tetrahedra are ultimately described as partially transparent, triangular elements for hardware rendering. Tetrahedra demonstrate the images of high quality that can be produced as well as to oblige volumes more general than rectilinear grids. An example of a liquid oxygen post data-set rendered by the PT technique could be seen in the below figure.

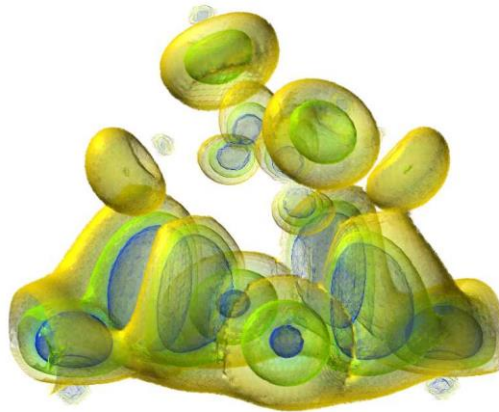


Fig. 10. Liquid oxygen post data-set rendered by PT algorithm [14].

2.3 Maximum Intensity Projection (MIP) technique

Maximum Intensity Projection (MIP) [15] is a technique that renders values which are observed throughout every pixels' highest intensity. In point of fact, it could be considered as a less complicated version of the Direct Volume Rendering technique, where unambiguous classification of transfer functions is not mandatory. Scalar field values are accompanied with an intensity in the MIP technique.

Each maximum intensity observed through each pixel is projected to the pixel corresponding to it. MIP could be considered as a search algorithm more than a color/opacity volume reconstruction algorithm. It is usually used in medical visualization, especially for circulatory systems. The data is a lot of slices, where most regions are dim. However, vessels will in general be more luminous. This set is crumpled into a solitary image by carrying out a projection through the set that allots the brightest voxel over all slices to every pixel in the projection (see Figure 11). Opposite to the DVR algorithm, MIP algorithm does not require the tiresome formation of color and opacity transfer functions. However not at all like surface SR, it shows information of data's

density in all cases. Moreover, since datasets as a rule contains a ton of noise, threshold esteems for SR, which permit extraction of vascular structures that are hard to discover. One of the extensions of MIP is Depth-shaded Maximum Intensity Projection, this method modules data values by their depth to attain depth shading. Figure 12 below shows the renders of human feet dataset using different types of approaches.

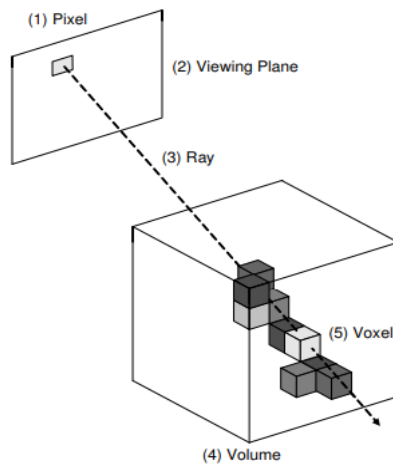


Fig. 11. Maximum Intensity Projection through ray casting illustration [15].

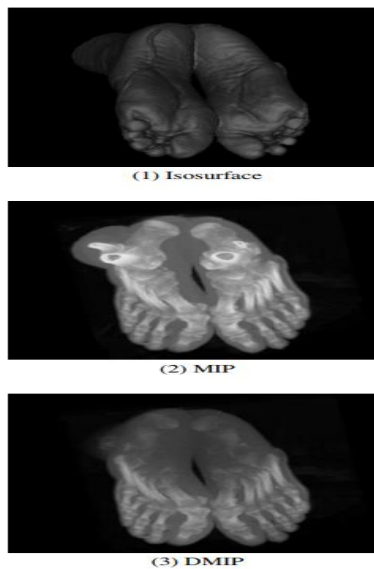


Fig. 12. A human feet dataset rendered using an iso-surface ray cast function, MIP and DMIP - the problem of missing depth information in MIP is clearly visible [15].

3 Conclusion

In this study, numerous volume rendering techniques has been reviewed comprehensively, and showing / discussing the benefits and downsides of them. The significance of this work is to giving a clear understanding intensively to the audience and readers of the importance and benefits of each of the volume rendering techniques, and their applications.

4 Future Work

This study provides a robust foundation for future work in volume rendering technologies. One extent of our future work is to combine the new developments in machine learning with volume rendering. The goal is to develop software that would render medical imaging data, which recognize real anatomical structure, transforming grayscale data to RGB, and detecting abnormalities. Moreover, our intent is to investigate more volume rendering techniques and compare them with the above-mentioned ones using benchmarks and statistics.

5 References

- [1] Pfister, H., et al. The VolumePro real-time ray-casting system. in Proceedings of the 26th annual conference on Computer graphics and interactive 161 techniques. 1999. New York, New York, USA: ACM Press/Addison-Wesley Publishing Co. <https://doi.org/10.1145/311535.311563>
- [2] Wu, Y., et al. Shear-image order ray casting volume rendering. in Proceedings of the 2003 symposium on Interactive 3D graphics. 2003. New York, New York, USA: ACM <https://doi.org/10.1145/641480.641510>
- [3] Noon, Christian John, "A Volume Rendering Engine for Desktops, Laptops, Mobile Devices and Immersive Virtual Reality Systems using GPU-Based Volume Raycasting" (2012). Graduate Theses and Dissertations. 12419.
- [4] Westover, L., Footprint evaluation for volume rendering. ACM SIGGRAPH Computer Graphics, 1990. 24: p. 367-376. <https://doi.org/10.1145/97880.97919>
- [5] Botsch, M. and L. Kobbelt, High-quality point-based rendering on modern GPUs. 11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings., 2003: p. 335-343
- [6] Zwicker, M. and H. Pfister, Hardware-Accelerated Adaptive EWA Volume Splatting. IEEE Visualization 2004, 2004: p. 67-74.
- [7] Neophytou, N. and K. Mueller. GPU accelerated image aligned splatting. in Fourth International Workshop on Volume Graphics, 2005. 2005. IEEE. <https://doi.org/10.1109/VG.2005.194115>
- [8] Rezk-Salama, C., et al. Interactive volume on standard PC graphics hardware using multi-textures and multi-stage rasterization. in Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware. 2000. New York, New York, USA: ACM. <https://doi.org/10.1145/346876.348238>

- [9] Engel, K., M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. in Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware - HWWS '01. 2001. New York, New York, USA: ACM Press <https://doi.org/10.1145/383507.383515>
- [10] Takayama, M., Mori, S., Shinomoto, Y., Goshima, M., Nakashima, Y. and Tomita, S. CELL-PROJECTION PARALLEL VOLUME RENDERING WITH EARLY RAY TERMINATION. Proceedings of the Annual Conference of the Institute of Image Electronics Engineers of Japan (2004).
- [11] Roettger, S., et al. Smart hardware-accelerated volume rendering. in Proceedings of the symposium on Data visualisation 2003. 2003. Eurographics Association.
- [12] Lorensen, W.E. and H.E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm. ACM Siggraph Computer Graphics, 1987. 21: p. 163-169. <https://doi.org/10.1145/37402.37422>
- [13] Shirley, P. and Tuchman, A. 1990. A polygonal approximation to direct scalar volume rendering. Proceedings of the 1990 workshop on Volume visualization - VVS '90 (1990). <https://doi.org/10.1145/99307.99322>
- [14] Marroquim, Ricardo & Maximo, Andre & Farias, Ricardo & Esperança, Claudio. (2008). Volume and Isosurface Rendering with GPU - Accelerated Cell Projection. Computer Graphics Forum. 27. 24 - 35. 10.1111/j.1467-8659.2007.01038.x. <https://doi.org/10.1111/j.1467-8659.2007.01038.x>
- [15] Bruckner, Stefan. Performing Maximum Intensity Projection with the Visualization Toolkit. Seminar Paper, The Institute of Computer Graphics and Algorithms, Vienna University of Technology (2002).

6 Authors

Amr S. Mady received his B.Sc. degree in Computer Science from The British University in Egypt (BUE) in 2018. His research interests are: Computer Graphics (CG), Computer Vision (CV), Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), Machine Learning (ML), Mobile Applications, and Medical Application. He worked as a software engineer at a Cairo based software solutions services company at 2017, currently he is a software engineer for a Cairo based technological startup (Halan Inc.). He published 3 research papers in conference proceedings and reputable international journals. E-mail: amrsaeedmady@gmail.com

Samir A. El-Seoud received his B.Sc. degree in Physics, Electronics and Mathematics from Cairo University in 1967, his Higher Diploma in Computing from the Technical University of Darmstadt (TUD) - Germany in 1975 and his Doctor of Science from the same University (TUD) in 1979. His research interest is focused among others on: Parallel Numerical Algorithms, Scientific Computations, Numerical Techniques for Solving Nonlinear Problems, Collaborative Learning, Computer Aided Learning, and Mobile Applications. He held different academic positions at TUD Germany. He has been a Full-Professor since 1987. Outside Germany, he spent several years as a Full-Professor of Computer Science at SQU – Oman, Qatar University, and PSUT-Jordan and acted as a Head of Computer Science for many years. With industrial institutions, he worked as Scientific Advisor and Consultant for the GTZ in Germany and was re-

sponsible for establishing a postgraduate program leading to M.Sc. degree in Computations at Colombo University, Sri-Lanka (2001 – 2003). He also worked as an Application Consultant at Automatic Data Processing Inc., Division Network Services in Frankfurt/Germany (1979 – 1980). Currently, Professor El-Seoud is with the Faculty of Informatics and Computer Science of the British University in Egypt (BUE). He published over 90 research papers in conference proceedings and reputable international journals. E-mail: samir.elseoud@bue.edu.eg

Article submitted 2020-02-06. Resubmitted 2020-03-14. Final acceptance 2020-03-16. Final version published as submitted by the authors.