

Efficient Solution of EFIE via Low-Rank Compression of Multilevel Predetermined Interactions

Dipanjan Gope, *Student Member, IEEE*, and Vikram Jandhyala, *Senior Member, IEEE*

Abstract—This paper describes the predetermined interaction list oct-tree (PILOT) algorithm and its application in expediting the solution of full-wave electric field integral equation (EFIE)-based scattering problems for three-dimensional arbitrarily shaped conductors. PILOT combines features of the fast multipole method (FMM) and QR decomposition-based matrix compression techniques to optimize setup times, solve times, and memory requirements. The method is kernel independent and stable for electrically small structures unlike traditional FMM. The novel features of the algorithm, namely the mixed potential compression scheme and the hierarchical multilevel predetermined matrix structure are explained in detail. A complexity estimate is presented to demonstrate the scaling in time and memory requirements. Examples exhibiting the accuracy and the time and memory performances are also presented. Finally, a quantitative study is included to address the expected but gradual degradation of QR-based compression techniques for electrically large structures.

Index Terms—Fast solver, integral equations, low-rank compression, radar cross section (RCS) computations.

I. INTRODUCTION

EFFICIENT modeling and simulation of electromagnetic (EM) wave scattering by conducting structures continues to be an active topic of research. The applications include, but are not limited to radar cross section (RCS) computations, antenna analysis, remote sensing, biomedicine, electromagnetic interference (EMI), and electromagnetic compatibility (EMC). Method of Moments (MoM) [1]-based electric field integral equations [2], magnetic field integral equations [3] and combined field integral equations [4] are among the common techniques employed in the solution of such problems. In this paper, we focus on electric field integral equation (EFIE)-based MoM solvers, though the PILOT algorithm is general enough to be applicable to all the above formulations.

The traditional EFIE-MoM solver leads to the generation of a dense matrix with large number of unknowns, the solution of which presents a significant time and memory bottleneck. To alleviate this problem, fast iterative solution algorithms

have been developed which compress the dense system by exploiting the basic physics of Green's function interactions. The most popular compression techniques include the fast multipole method (FMM) [5]–[9], the adaptive integral method (AIM) [10], the precorrected fast fourier transform (FFT) method [11], [12], the QR-based methods [13]–[20] and the multilevel matrix decomposition algorithm (MLMDA) [21], [22]. Of the QR-based methods, IE-QR [17], [18] is based on single level matrix decomposition while others are multilevel approaches. The memory requirement and the setup time is reduced from $O(N^2)$ to $O(N \log N)$ for multilevel techniques and the solve time is reduced from $O(N^3)$ as in Gaussian elimination or $O(N^2) \times p \times r$ as in a regular iterative solver to $O(N \log N) \times p \times r$, where N is the number of basis functions, p is the number of iterations for convergence per right-hand side (RHS), and r is the number of RHS vectors. Though similar in essence, the compression schemes differ in performance. The FFT-based methods are simpler to parallelize, but degrade in efficiency for strongly nonuniform distributions of basis-functions due to the inherent necessity of a uniform global grid. The FMM and the QR-based algorithms on the other hand are ideally suited for application to arbitrarily shaped structures. However, they tend to have different relative setup and solve time performances, with FMMs having lower setup times owing to well-defined and predetermined interaction regions and reuse of multipole and local expansion information across levels. On the other hand, some QR methods, like IES³, require increased setup times due to expensive binary tree splits and merges. These methods, however, tend to require less memory and produce faster matrix-vector products owing to superior customized compaction in the setup phase. Due to these differences, the applicability of the algorithms becomes problem specific. QR-based methods are ideally suited for problems with a large number of RHS vectors which involve a single setup but several matrix-vector products in multiple iterative solutions; the FMM is better applicable otherwise, i.e., for smaller number of RHSs.

In the presented paper, QR compression on a predetermined interaction list reduces the setup-time as compared to binary tree-based methods like IES³ [13]–[15] and the UV-method [19], without compromising on the memory or solve-time and thereby yields a high efficiency solver irrespective of the number of RHSs. This paper describes the PILOT algorithm and its application to scattering problems. The novel features in the presented algorithm are twofold.

Manuscript received October 19, 2004; revised February 23, 2005. This work was supported in part by the U.S. DARPA/MTO NeoCAD program by Grant N66001-01-1-8920, NSF Career Award Grant ECS-0093102, and NSF/SRC Joint Initiative on Mixed-Signal Electronic Technologies.

D. Gope was with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: dips@u.washington.edu). He is now with Intel Corporation, Santa Clara, CA 95052 USA.

V. Jandhyala is with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: jandhyala@ee.washington.edu).
Digital Object Identifier 10.1109/TAP.2005.856350

- i) Mixed potential compression: The compression efficiencies of QR-based EFIE solvers depend on the methodology adopted to compress the scalar and vector potentials. Different schemes are studied (Section V) and the technique adopted in PILOT is demonstrated to be more efficient than the ones employed by IES³ [13]–[15] or the single level IE-QR [17], [18].
- ii) Hierarchical form: The multilevel oct-tree structure (common to FMM approaches) and the QR compression efficiency of IES³ are combined and further optimizations in the form of the merged interaction list (MIL) ensure a good all-round performance.

The QR compression is achieved by employing the modified Gram Schmidt (MGS) method [23] on selected rows and columns of submatrices thereby ensuring a $O(N \log N)$ setup time, as discussed in Section III. PILOT, like IES³, is stable for the analysis of electrically small structures unlike traditional FMM that breaks down when the size of the lowest level cube is less than one-fifth of the wavelength [9] due to near-singularities in the translation operator. Like other QR based methods, PILOT is kernel independent and can be applied directly with multilayered dielectric Green's functions. PILOT is a multilevel approach and its $O(N \log N)$ scaling is more favorable than the corresponding $O(N^{1.5})$ scaling of single level approaches like IE-QR [17], [18]. Also, unlike the MLMDA approach which requires the development of equivalent sources and observers, PILOT applies compression directly on the matrix entries and the interactions do not occur across levels.

The rest of the paper is organized as follows. Section II introduces the EFIE-MoM-based formulation. In Section III, the PILOT-QR algorithm is discussed in detail. In Section IV, a complexity estimate for the presented algorithm is derived. The relative advantages of PILOT over existing QR-based methods are discussed in Section V. Results demonstrating the accuracy, time, and memory efficiency of the algorithm are presented in Section VI. In Section VII, a quantitative study is performed on the gradual degradation of the QR based algorithms for electrically massive structures. Section VIII discusses conclusions.

II. EFIE

Conducting three-dimensional (3-D) arbitrary shaped objects illuminated by incident fields can be treated by applying the EFIE, wherein the boundary condition on the surface S of the object is given by

$$(\mathbf{E}^s(\mathbf{J}) + \mathbf{E}^i)_{\tan} = 0 \quad (2.1)$$

where \mathbf{E}^s is the scattered electric field produced by the surface current \mathbf{J} , \mathbf{E}^i is the incident electric field, and subscript \tan denotes the tangential components on S . In terms of potentials, the scattered electric field can be written as

$$\mathbf{E}^s(\mathbf{J}) = -j\omega\mathbf{A} - \nabla\Phi \quad (2.2)$$

where ω is the angular frequency, and the vector potential \mathbf{A} is defined as

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi} \int_S \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \mathbf{J}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \quad (2.3)$$

the scalar potential Φ is defined as

$$\Phi(\mathbf{r}) = \frac{1}{4\pi\epsilon} \int_S \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \quad (2.4)$$

where \mathbf{r} and \mathbf{r}' are observation and source locations and μ , ϵ , k are the permeability, permittivity, and wavenumber, respectively. \mathbf{J} and ρ represent the surface current density and surface charge density, respectively, which are related by the continuity equation:

$$\nabla_s \cdot \mathbf{J}(\mathbf{r}) + j\omega\rho(\mathbf{r}) = 0, \quad \forall \mathbf{r} \in S \quad (2.5)$$

where ∇_s represents surface divergence.

Using triangular tessellations, Rao–Wilton–Glisson (RWG) basis functions \mathbf{f} [2] and Galerkin testing, the EFIE system can be written in the form of a matrix equation:

$$\bar{\mathbf{Z}}_{N_e \times N_e} \mathbf{x}_{N_e \times 1} = \mathbf{b}_{N_e \times 1} \quad (2.6)$$

where N_e is the number of RWG edges, \mathbf{x} comprises the unknowns namely the coefficients of RWG bases, and the matrix and RHS entries are given by

$$\begin{aligned} \bar{Z}_{ij} = & \frac{j\omega\mu}{4\pi} \int_{T_i^{+,-}} \int_{T_j^{+,-}} \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \mathbf{f}_j(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \cdot \mathbf{f}_i(\mathbf{r}) ds \\ & + \frac{1}{4j\omega\pi\epsilon} \int_{T_i^{+,-}} \int_{T_j^{+,-}} \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \nabla' \cdot \mathbf{f}_j(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \nabla \\ & \cdot \mathbf{f}_i(\mathbf{r}) ds \end{aligned} \quad (2.7a)$$

$$\mathbf{b}_i = \int_{T_i^{+,-}} f_i \cdot \mathbf{E}^i ds \quad i, j = 1, 2, \dots, N_e \quad (2.7b)$$

where $T^{+,-}$ indicates the domain of definition for the RWG basis function namely the positive and the negative triangles associated with the RWG edge, such that the current is assumed to flow from the positive to the negative triangle. The resultant matrix is dense which creates a time and memory bottleneck that is addressed by the PILOT algorithm described in Section III.

III. PILOT EFIE ALGORITHM

The predetermined interaction list oct-tree QR algorithm efficiently compresses the method of moments' matrix in a multilevel scheme. This leads to reduced memory and fast matrix vector products that can be exploited by a Krylov-subspace iterative solver such as GMRES [24]. In this section, three main issues pertaining to the PILOT algorithm are discussed, namely the mixed potential compression scheme, the predetermined matrix structure for fast setup and the QR formation technique using only selected column and row samples.

A. Mixed Potential Scheme

The key idea behind the compression is that the interaction submatrix between groups of well-separated basis-functions is low-ranked with a given tolerance level. If N_e^s and N_e^o are the number of RWG edges in the well-separated source and the observer groups respectively and N_p^s and N_p^o are the corresponding number of patches, then the EFIE interaction subma-

trix is given by $\bar{\mathbf{Z}}_{N_e^o \times N_e^s}^{\text{sub}}$, which can be further represented into its components as

$$\begin{aligned} \bar{\mathbf{Z}}_{N_e^o \times N_e^s}^{\text{sub}} &= \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{++} + \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{--} + \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{+-} + \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{-+} \\ &\quad + \bar{\mathbf{P}}_{N_e^o \times N_e^s}^{++} + \bar{\mathbf{P}}_{N_e^o \times N_e^s}^{--} \\ &\quad + \bar{\mathbf{P}}_{N_e^o \times N_e^s}^{+-} + \bar{\mathbf{P}}_{N_e^o \times N_e^s}^{-+} \end{aligned} \quad (3.1a)$$

where the component entries are given as in

$$\begin{aligned} \bar{\mathbf{L}}_{ij}^{+-} &= \frac{j\omega\mu}{4\pi} \int_{T_i^+} \int_{T_j^-} \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \mathbf{f}_j^{T_j^-}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \cdot \mathbf{f}_i^{T_i^+}(\mathbf{r}) ds \\ i &= 1, 2, \dots, N_e^o; \quad j = 1, 2, \dots, N_e^s \end{aligned} \quad (3.1b)$$

and

$$\begin{aligned} \bar{\mathbf{P}}_{ij}^{+-} &= \frac{1}{4j\omega\pi\epsilon} \int_{T_i^+} \int_{T_j^-} \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \nabla' \cdot \mathbf{f}_j^{T_j^-}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \nabla \\ &\quad \cdot \mathbf{f}_i^{T_i^+}(\mathbf{r}) ds \\ i &= 1, 2, \dots, N_e^o; \quad j = 1, 2, \dots, N_e^s \end{aligned} \quad (3.1c)$$

where $\mathbf{f}_j^{T_j^\pm}$ denotes the half of RWG function $\mathbf{f}_j^{T_j^{+-}}$ which is defined on the positive or negative triangle as indicated in the superscript.

In PILOT compression is performed on the component submatrices. $\bar{\mathbf{Z}}_{N_e^o \times N_e^s}^{\text{sub}}$ is represented as

$$\bar{\mathbf{Z}}_{N_e^o \times N_e^s}^{\text{sub}} = \bar{\mathbf{L}}_{N_e^o \times N_e^s} + \bar{\mathbf{B}}_{O_{N_e^o \times N_e^o} \times N_e^o} \bar{\mathbf{P}}_{N_e^o \times N_e^p} \bar{\mathbf{B}}_{S_{N_e^p \times N_e^s}^T} \quad (3.2a)$$

where

$$\bar{\mathbf{L}}_{N_e^o \times N_e^s} = \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{++} + \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{--} + \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{+-} + \bar{\mathbf{L}}_{N_e^o \times N_e^s}^{-+} \quad (3.2b)$$

$$\begin{aligned} \bar{\mathbf{B}}_{ij} &= \begin{cases} 1 & \text{Patch } j \text{ is the positive patch of Edge } i \\ -1 & \text{Patch } j \text{ is the negative patch of Edge } i \\ 0 & \text{Patch } j \text{ does not belong to Edge } i \end{cases} \\ i &= 1, 2, \dots, N_e; \quad j = 1, 2, \dots, N_p \end{aligned} \quad (3.2c)$$

$$\bar{\mathbf{P}}_{ij} = \frac{1}{4j\omega\pi\epsilon a_i a_j} \int_{T_i} \int_{T_j} \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} ds' ds \quad (3.2d)$$

where a_i and a_j are the areas of the observer and source triangles, respectively. The pre- and postmultiplier $\bar{\mathbf{B}}$ matrices are inherently sparse while the edge-by-edge $\bar{\mathbf{L}}$ matrix and the patch-by-patch $\bar{\mathbf{P}}$ matrix are compressed by applying QR decomposition on their subblocks $\bar{\mathbf{M}}_{m \times n}$ representing the corresponding interactions between well-separated m source basis functions and n observer basis functions

$$\bar{\mathbf{M}}_{m \times n} = \bar{\mathbf{Q}}_{m \times r} \bar{\mathbf{R}}_{r \times n} \quad (3.3)$$

where r is the rank of $\bar{\mathbf{M}}_{m \times n}$ under the given tolerance ϵ and will henceforth be referred to as epsilon-rank. The storage cost required for the compressed representation is $(m+n)r$ in comparison to mn required for the uncompressed subblock of size $m \times n$. The choice of this particular mixed scheme is aimed at achieving the most efficient time and memory performance per submatrix, as is demonstrated in Section V.

B. Hierarchical Predetermined Matrix-Block Structure

As aforementioned, the efficient compression of a single EFIE submatrix using the mixed potential scheme is discussed. The next step is the identification of a deterministic matrix block structure, or in other words, identification of the set of all EFIE submatrices to be compressed in a multilevel scheme so as to ensure the lowest memory requirement. This is directly dependent on the hierarchy employed to subdivide the geometry and the designation of the source-observer interaction groups. The PILOT algorithm is based on an oct-tree hierarchy which is common to the FMM approach, followed by the creation of a MIL as discussed later, to maximize compression.

1) *Oct-Tree Geometry Decomposition in 3-D*: The starting cell c_0^0 is the smallest cube that encloses the entire geometry. Each cube c_i^l , which is the i th cube at level l , is decomposed by spatially balanced splits along each coordinate, e.g., x , y , and z in 3-D. Cube c_j^{l+1} resulting from this decomposition is called a child of c_i^l and the latter is denoted as the parent of c_j^{l+1}

$$P_{c_j^{l+1}} = c_i^l. \quad (3.4)$$

All the child cubes of c_i^l are siblings of each other, such that a sibling set is defined as

$$S_{c_i^l} = \{c_k^{l+1} \mid \forall k | P_{c_k^{l+1}} = c_i^l\}. \quad (3.5)$$

The geometric decomposition is, hence, exactly similar to that of multilevel FMM and, therefore, its interaction scheme can be leveraged.

2) *FMM Multilevel Interaction Shell*: Every cube $c_i^l \forall i$, $l | 0 \leq l \leq l_c; 0 \leq i < n_c^l$, where l_c is the total number of levels and n_c^l is the total number of cubes at level l , has a nearest neighbor shell $K_{c_i^l}$ and an interaction shell $I_{c_i^l}$. The nearest neighbor shell is defined as

$$\begin{aligned} K_{c_i^l} &= \left\{ c_j^l | c_j^l \text{ is in the same level as } c_i^l \text{ and} \right. \\ &\quad \left. \text{has at least one contact point with } c_i^l \right\}. \end{aligned} \quad (3.6)$$

Consequently the interaction shell is defined as

$$I_{c_i^l} = \left\{ c_j^l | P_{c_j^l} \in K_{P_{c_i^l}}; c_j^l \notin K_{c_i^l} \right\}. \quad (3.7)$$

3) *PILOT MIL*: It is possible to compress each submatrix pertaining to the interaction between the observer basis functions residing in cube c_i^l and source basis functions residing in any cube $c_j^l | c_j^l \in I_{c_i^l}$. However, closer inspection reveals a pattern in the interaction shell of siblings that can be exploited to achieve further compression. The interaction shells of siblings as demonstrated in Fig. 1(a), share many common cubes, designated as the common interaction region, denoted by I_s [Fig. 1(b)]

$$I_s = \bigcap I_{c_i^l} \neq \emptyset \quad \forall i | c_i^l \in S_{c_j^l}. \quad (3.8)$$

For visualization purposes, the 2-D interaction shells are illustrated in Fig. 1, though our algorithm is designed for 3-D geometries.

TABLE I
DISTINCT TYPES OF MIL

Type	Name	Number of observer cubes	Number of source cubes	Number of orientations	Max. $\varepsilon = 10^{-3}$ Rank (r_{\max}^p)
1	Outer-wall	8	12	6 (1 on each wall)	8
2	Outer-corner	8	10	8 (1 on each corner)	14
3	Inner-4	4	9	6	14
4	Inner-2	2	3	12	8
5	Inner-1	1	1	8	4

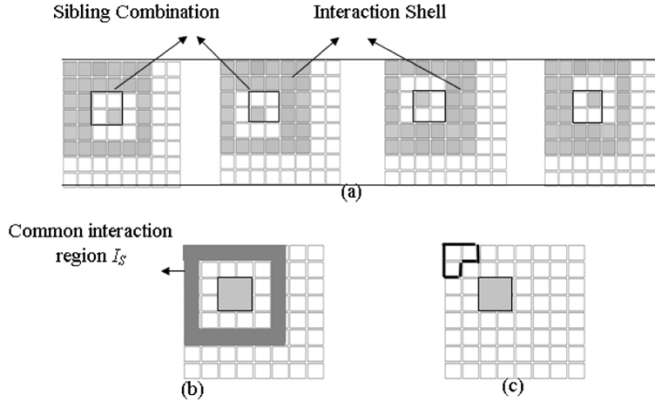


Fig. 1. (a) The individual interaction shells of siblings. (b) Common interaction shell for the sibling combination. (c) A merged interaction.

It is, therefore, possible to group source sibling cubes and common interaction region observer cubes in order to compress larger matrices to low epsilon-ranks and thereby gain in terms of overall compressibility. The common interaction region, as shown in Fig. 1(b), is decomposed into disjointed parts so as to maintain the low rank of such an interaction. Each such disjointed part is called a merged interaction (μ) as shown in Fig. 1(c).

Any given set of siblings S_{c_i} has an associated list of merged interaction entries, collectively called the Merged Interaction List (MIL), which completely spans all same level interactions for observer basis functions residing in cubes $c_j^l | c_j^l \in S_{c_i}$. The features of MIL for a given kernel are as follows:

- The MIL pattern for a given sibling combination is invariant from that of a different sibling combination at the same level.
- The MIL pattern is the same for sibling combinations across levels. A higher-level MIL is just a magnified version of that at the lower level.
- Each MIL entry corresponds to a maximum epsilon-rank (r_{\max}^p) for its $\bar{\mathbf{P}}$ interaction submatrix that is determined as a one-time process for a random distribution of basis functions in the observer and source clusters. The epsilon-rank for that of $\bar{\mathbf{L}}$ is $3 \times r_{\max}^p$ for 3-D structures and $2 \times r_{\max}^p$ for 2-D structures.

For a 3-D case, the MIL consists of 40 entries with five distinct types as shown in Fig. 2. Each one of the 40 entries corresponds to one of the five types, differing only in relative orientations. The cubes in the middle are combination of observer siblings

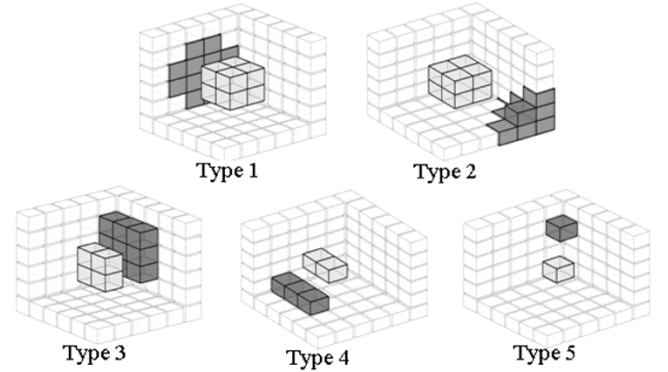


Fig. 2. The types of merged interactions constituting the MIL.

from the same sibling set and the darker cubes constitute the merged common interaction source region.

A brief description of the distinct types is provided in Table I. The decomposition of the common interaction regions into merged interaction entries could be performed in many different ways. However, based on *a priori* extensive experimentations involving random distributions of basis functions, it has been found that the combination demonstrated above is most efficient in terms of the overall rank-reduction for the entire list. Also, it must be noted that the maximum ranks obtained in Table I correspond to the frequency region where the variation in the kernel is dominated by $1/r$ rather than e^{-jkr} . For higher frequencies, as demonstrated in Section VII, the ranks are expected to increase. This issue is addressed by maintaining a multiplying factor for the ranks, which is constant for all entries in the MIL for a given frequency.

C. QR Compression Using Sampled Rows and Columns

For every MIL entry, two low-epsilon-ranked submatrices $\bar{\mathbf{L}}^{\text{sub}}$ and $\bar{\mathbf{P}}^{\text{sub}}$, corresponding to (3.2a), are compressed using only sampled rows and columns. Thereby the construction of the entire submatrix is avoided and compression is achieved in $O(N \log N)$ time. The procedure of obtaining the samples rows $\bar{\mathbf{S}}_r$ and sampled columns $\bar{\mathbf{S}}_c$, for the given submatrix $\bar{\mathbf{Z}}_{m \times n}^{\text{sub}} | \bar{\mathbf{Z}}^{\text{sub}} \in \{\bar{\mathbf{L}}^{\text{sub}}, \bar{\mathbf{P}}^{\text{sub}}\}$ with m rows and n columns, was suggested by Kapur and Long [13]–[15] and is briefly discussed later.

The first row and the first column are selected by default as the starting set. Consequently, new rows and columns, to be constructed, are selected alternately based on the following algorithm:

Algorithm 1:

Let $I_r = \{a, b, c, \dots\}$ be the indexes of the k rows and $I_c = \{A, B, C, \dots\}$ be the indexes of the k columns, chosen already as shown in Fig. 3(a), such that $1 \leq a, b, c, \dots \leq m$, $1 \leq A, B, C, \dots \leq n$ and $a \neq b \neq c, \dots$, $A \neq B \neq C, \dots$. The $(k+1)$ th column and $(k+1)$ th row, to be constructed are chosen through the following steps.

Step 1: k vectors, $v_1^{i=1,2,\dots,k}$, each of length k are constructed from the already formed columns using the values only at the indexes of the already constructed rows, as in

$$v_1^i = [\bar{\mathbf{Z}}^{\text{sub}}(a, I_c[i]); \bar{\mathbf{Z}}^{\text{sub}}(b, I_c[i]); \bar{\mathbf{Z}}^{\text{sub}}(c, I_c[i]); \dots] \quad (3.9)$$

such that

$$v_1^{i=1} = [\bar{\mathbf{Z}}^{\text{sub}}(a, A); \bar{\mathbf{Z}}^{\text{sub}}(b, A); \bar{\mathbf{Z}}^{\text{sub}}(c, A); \dots]$$

Step 2: $m - k$ vectors, $v_2^{i=1,2,\dots,m-k}$, each of length k are formed from the entries of the nonselected columns, already constructed in the process of row-construction, as indicated by circles in

Fig. 3(a). These vectors are of the form $v_2^j = [\bar{\mathbf{Z}}^{\text{sub}}(a, J); \bar{\mathbf{Z}}^{\text{sub}}(b, J); \bar{\mathbf{Z}}^{\text{sub}}(c, J); \dots]$ where $1 \leq J \leq n$ and $J \neq A \neq B, \dots$.

Step 3: For each vector v_2^j constructed in step 2, a value is obtained corresponding to

$$\text{MinVal}(j) = \min \left\{ \left\| v_1^1 - v_2^j \right\|, \left\| v_1^2 - v_2^j \right\|, \left\| v_1^3 - v_2^j \right\|, \dots \right\} \quad (3.10)$$

Step 4: The maximum entry of the *MinVal* vector obtained in the previous step is determined and the column corresponding to this maximum value is chosen as the $(k+1)$ th column, marked *D* in **Fig. 3(b)**.

The corresponding $(k+1)$ th row is constructed using similar steps with row vectors instead of column vectors. The total number of sampled rows and columns constructed is typically twice the expected maximum rank of the given interaction. Once the sampled rows and columns are formed, the following steps enable the representation of $\bar{\mathbf{Z}}^{\text{sub}}$ as

$$\bar{\mathbf{Z}}_{m \times n}^{\text{sub}} = \bar{\mathbf{Q}}_{m \times r} \bar{\mathbf{R}}_{r \times n}. \quad (3.11)$$

First, $\bar{\mathbf{Q}}_{m \times r}$ is formed by employing MGS decomposition on $\bar{\mathbf{S}}_c$

$$\bar{\mathbf{S}}_{c \times s} = \bar{\mathbf{Q}}_{m \times r} \bar{\mathbf{R}}'_{r \times s} \quad (3.12)$$

where $\bar{\mathbf{Q}}_{m \times r}$ is a unitary matrix, $\bar{\mathbf{R}}'_{r \times s}$ is upper triangular and s is the number of samples chosen (usually twice the MIL interaction epsilon-rank). Matrix $\bar{\mathbf{Q}}_{s \times r}$ is formed by taking rows of

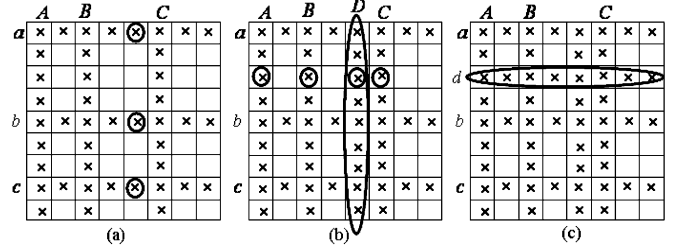


Fig. 3. (a) Selection of new column. (b) Construction of new column and selection of new row. (c) Construction of new row.

$\bar{\mathbf{Q}}_{m \times r}$, such that the indexes of those rows are the same as the ones used to construct $\bar{\mathbf{S}}_r$ from $\bar{\mathbf{Z}}^{\text{sub}}$. Under such conditions

$$\bar{\mathbf{S}}_{r \times n} = \bar{\mathbf{Q}}_{s \times r} \bar{\mathbf{R}}_{r \times n}. \quad (3.13)$$

To solve for $\bar{\mathbf{R}}_{r \times n}$ from (3.13), $\bar{\mathbf{Q}}_{s \times r}$ is decomposed using MGS into a unitary matrix $\bar{\mathbf{Q}}'_{s \times r}$ and an upper triangular square matrix $\bar{\mathbf{R}}_{r \times r}$

$$\bar{\mathbf{Q}}_{s \times r} = \bar{\mathbf{Q}}'_{s \times r} \bar{\mathbf{R}}_{r \times r}. \quad (3.14)$$

Using (3.14) and the properties of $\bar{\mathbf{Q}}'_{s \times r}$, (3.11) can be written as

$$\bar{\mathbf{Q}}_{s \times r}'^T \bar{\mathbf{S}}_{r \times n} = \bar{\mathbf{R}}_{r \times r} \bar{\mathbf{R}}_{r \times n}. \quad (3.15)$$

From (3.15), $\bar{\mathbf{R}}_{r \times n}$ can be extracted by backsubstitution since $\bar{\mathbf{R}}_{r \times r}$ is a square upper triangular matrix.

IV. COMPLEXITY ANALYSIS

The multilevel PILOT is a $O(N \log N)$ complexity algorithm in time and memory. In this section, the complexity estimate of the algorithm is obtained with regards to the number of elements that need to be stored in a compressed representation. A uniform 3D distribution of basis functions is assumed for the surface of the entire geometry where the total number of RWG edges is N_e . The storage of the compressed vector potential matrix $\bar{\mathbf{L}}$ is rigorously investigated below. At first, the cost of the QR matrix storage for a given MIL is computed. This storage depends heavily on the epsilon-ranks of each interaction. Usually for most distributions the computed epsilon-ranks are much smaller than the maximum epsilon-ranks (Table I). However, in this estimate we obtain the higher cost bound by considering the maximum epsilon-ranks for all computations.

At a given level l , the total number of cubes is 8^l and the number of nonempty cubes, for a surface only distribution, is 4^l . The corresponding number of edges per cube is $n(l) = (N_e/4^l)$. The cost of QR storage for each of the five types of MIL is given by

$$3r_p^{\text{max}} n_{\text{type}} \left(\frac{n_{\text{obs}} + n_{\text{src}}}{2^l} \right) n(l) \quad (4.1)$$

where n_{obs} and n_{src} are the number of observer and source cubes respectively, in an MIL entry, $3r_p^{\text{max}}$ is the maximum epsilon-rank for vector potentials, n_{type} is the number of orientations for the MIL type and $1/2^l$ is the fraction of cubes that are

TABLE II
THE QR STORAGE COST FOR A SINGLE MIL

MIL Type	Total Storage Cost
Outer-wall	$3840 n(l) / 2^l$
Outer-Corner	$4536 n(l) / 2^l$
Inner-4	$3276 n(l) / 2^l$
Inner-2	$1440 n(l) / 2^l$
Inner-1	$192 n(l) / 2^l$

nonempty. Based on (4.1) the cost of storage of the 5 different types of MIL entries are tabulated in Table II.

The storage cost for a complete MIL is hence computed as $13284n(l)/2^l$. Considering that the total number of MILs in a single level is 4^{l-1} , the cost for a complete QR storage across all levels is $415N_e L$, where L is the total number of levels. On the other hand the cost of storing the dense matrices for neighbor interactions, which are restricted to the deepest level only, is $27N_e^2/2^{3L}$. The optimized value of L that minimizes the storage cost is given by

$$L = 0.48 \ln(0.135N_e). \quad (4.2)$$

Using (4.2) the total QR storage cost is obtained as $200N_e \ln(0.135N_e)$ while the dense neighbor interaction storage cost is $200N_e$.

Similarly, if N_p is the total number of patches, the storage cost for the scalar potential part can be computed as $66N_p \ln(0.406N_p) + 66N_p$. The memory requirement for the algorithm can be obtained by multiplying the element-by-element storage cost by the number of bytes required for each element. The number of operations for a single matrix-vector product is also directly related to the cost obtained above.

V. PILOT: RELATIVE ADVANTAGES

A. Algorithm Issues

Though the single level IE-QR [17], [18] scales as $O(N^{1.5})$, both PILOT and IES³ algorithms exhibit $O(N \log N)$ scaling in time and memory owing to the multilevel structure. However, IES³ suffers from a high setup time due to its ill-predictable matrix structure. IES³, being based on the binary tree geometry decomposition, gives rise to source and observer boxes of unequal shapes with variable separations, as demonstrated in Fig. 4.

Therefore, predicting the optimum matrix structure would necessitate infinite rank-map estimations. To counter this problem, IES³ resorts to splits and merges thereby climbing up and down the binary tree to maximize the compression. Thus the setup time is largely increased by the variability of the tree structure and the resulting backtracking and refinement. On the other hand the PILOT algorithm through the MIL, as discussed in

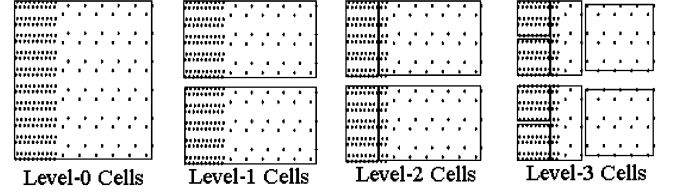


Fig. 4. Levels of binary tree decomposition.

TABLE III
EPSILON-RANK FOR INTERACTION MATRICES ENTRIES

Matrix	$\bar{\mathbf{P}}$	$\bar{\mathbf{H}}$	$\bar{\mathbf{L}}^{++} / \bar{\mathbf{L}}^{+-}$ $/ \bar{\mathbf{L}}^{-+} / \bar{\mathbf{L}}^{--}$	$\bar{\mathbf{L}}^{++} + \bar{\mathbf{L}}^{+-}$ $+ \bar{\mathbf{L}}^{-+} + \bar{\mathbf{L}}^{--}$	$\bar{\mathbf{P}}^{++} + \bar{\mathbf{P}}^{+-}$ $+ \bar{\mathbf{P}}^{-+} + \bar{\mathbf{P}}^{--}$	$\bar{\mathbf{Z}}$
Rank	3	9	9	9	10	12

Section III, enables a predetermined matrix setup procedure. The regular geometrical pattern, as encountered in the oct-tree subdivision scheme, aids in *a priori* indication of an exhaustive list of source-observer interactions that needs to be compressed.

B. Single Submatrix Compression Issues

In IE-QR [17], the entire submatrix is compressed whereas in IES³ [15], the compression is applied on components as in:

$$\bar{\mathbf{Z}}_{N_e^o \times N_e^s}^{\text{sub}} = \bar{\mathbf{A}}_{O_{N_e^o \times 3N_p^o}} \bar{\mathbf{H}}_{3N_p^o \times 3N_p^s} \bar{\mathbf{A}}_{S_{3N_p^s \times N_e^s}}^T + \bar{\mathbf{B}}_{O_{N_e^o \times N_p^o}} \bar{\mathbf{P}}_{N_p^o \times N_p^s} \bar{\mathbf{B}}_{S_{N_p^s \times N_e^s}}^T \quad (5.1a)$$

where

$$\bar{\mathbf{H}}_{ij} = \frac{j\omega\mu}{4\pi} \int_{T_{h_i}} \int_{T_{h_j}} \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|} \mathbf{h}_j(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} ds' \cdot \mathbf{h}_i(\mathbf{r}) ds$$

$$i = 1, 2, \dots, 3 \times N_p^o; \quad j = 1, 2, \dots, 3 \times N_p^s. \quad (5.1b)$$

\mathbf{h} is the positive half RWG (H-RWG) function, i.e., the part of the RWG function defined only on the positive triangle. Also the sparse post and pre-multipliers are defined as shown in (5.1c) at the bottom of the page. In IES³, the $\bar{\mathbf{H}}$ matrix is compressed using MGS. In order to perform a comparison of the storage required for the different schemes a simple rank-estimate is performed. A random distribution of RWG functions is created in a source sphere and an observer sphere that are separated by a distance four times the radius of each sphere. Interaction matrices between them are obtained and the epsilon-ranks are computed using $\varepsilon = 10^{-3}$, as shown in Table III.

Based on these findings, the storage, setup and matrix-vector product time (matvec-time) required by the compressed form of the interaction submatrix ($\bar{\mathbf{Z}}_{N_e^o \times N_e^s}^{\text{sub}}$) pertaining to Table III, has been demonstrated in Table IV. For clarity of demonstration, the number of patches in the source and the observer groups has

$$\bar{\mathbf{A}}_{ij} = \begin{cases} 1 & \text{H-RWG } j \text{ belongs to positive patch of Edge } i \\ -1 & \text{H-RWG } j \text{ belongs to negative patch of Edge } i \\ 0 & \text{H-RWG } j \text{ does not belong to any patch of Edge } i. \end{cases}$$

$$i = 1, 2, \dots, N_e; \quad j = 1, 2, \dots, 3 \times N_p. \quad (5.1c)$$

TABLE IV
EFFICIENCY COMPARISON ON THE SINGLE SUBMATRIX

Algorithm	Memory	Setup Time	Matrix Vector Time
IES ³	$K_1 60 N_p$	$K_2 504 N_p$	$K_3 60 N_p$
IE-QR	$K_1 36 N_p$	$K_2 432 N_p$	$K_3 36 N_p$
PILOT	$K_1 33 N_p$	$K_2 261 N_p$	$K_3 33 N_p$

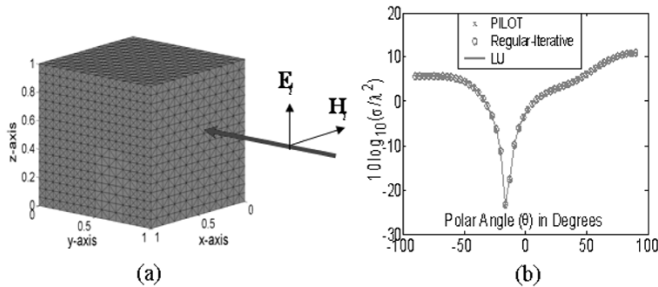


Fig. 5. (a) Surface mesh on cube with size 1 m. (b) The bistatic E-Plane RCS.

been assumed to be both equal to N_p and N_e is assumed to be 1.5 times N_p as in all closed structures.

K_1 , K_2 and K_3 are constants used for simplifying the expressions. As can be seen from Table IV, the PILOT mixed potential compression scheme is the cheapest form of storing individual submatrices. Similar experiments performed with different spacing between the source and observer groups reveal the same trend as seen above. Also, when applying EFIE with asymptotic waveform relaxation (AWE) [25] for fast frequency sweeps or with loop-star for low-frequency conditioning [26], storing the scalar and vector potentials separately as in IES³ and PILOT is necessary for efficient performance.

VI. RESULTS

In this section, simulation results are presented to demonstrate the accuracy, time, and memory efficiency of the PILOT algorithm. Both the QR decomposition tolerance and the absolute residual for the Krylov subspace iterative solution is set to 10^{-3} . All tests were run on a processor with 4 GB RAM and 1.2 GHz CPU speed.

A. Test Case 1

The first example is a canonical structure of a simple cube with each side length $w = 1$ m [Fig. 5(a)]. The cube is discretized using 5268 edges. The E-plane ($\phi = 90^\circ$) RCS is obtained for a frequency of 0.226 34 GHz ($w = 0.755\lambda$) using three methods: the regular LU method, the regular iterative solver and the PILOT iterative solver. The results are demonstrated in Fig. 5(b).

The results are in excellent agreement with each other and with the theoretical RCS results [27]. The backscattering corresponds to $\theta = 90^\circ$. The relative time and memory requirements for the problem are shown in Table V.

It can be seen that the PILOT has superior total-time and memory performance for the problem. The higher setup time of

TABLE V
TIME AND MEMORY REQUIREMENTS FOR THE CUBE

	Memory	Setup Time (sec)	Time per matvec (sec)	Solve Time (sec)
Regular LU	444MB	71	-	10212
Regular Iterative	444MB	71	8.05	602
PILOT Iterative	113MB	157	2.6	215

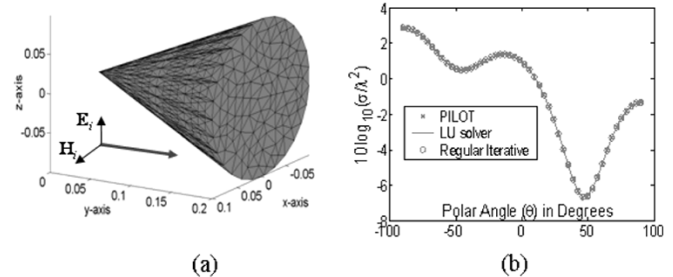


Fig. 6. (a) Conducting cone and incident plane wave. (b) The bistatic E-plane RCS.

PILOT is attributed to the overhead in compressing the submatrices. However, if the number of basis functions is increased, the setup time of PILOT will perform better than the other algorithms. This is because of the $O(N \log N)$ scaling of PILOT as compared to the quadratic scaling of the other methods, as shown in the next example.

B. Test Case 2

A closed conducting cone of height $h = 20$ cm and diameter $d = 20$ cm is considered [Fig. 6(a)]. The structure is meshed with 6105 edges. The bi-static RCS of the structure is computed for a 1 GHz plane-wave as shown in Fig. 6(a) at $\phi = 90^\circ$ using all 3 methods and the results are demonstrated in Fig. 6(b).

The mesh density is then gradually increased (Laplace scaling) and the performance scaling of the algorithm is obtained. The memory requirements, the matrix setup time, the time required for each matrix vector product and the total solve time are compared in Fig. 7. The setup time is the time required for constructing the selected MoM entries and for compressing the MIL submatrices. The total time is the time required between reading in the mesh and producing the solution. It includes the setup time and the solve time which involves multiple matrix vector products required for convergence. It can be seen from the plots that the PILOT algorithm scales as $O(N \log N)$ in all the aspects, while the scaling of the regular iterative solver is quadratic. The regular LU solver scales quadratically in terms of memory and is cubic in terms of solve-time.

C. Test Case 3

This example deals with an airborne drone structure [Fig. 8(a)]. The surface of the structure is discretized with 10 185 edges. The bistatic RCS of the structure at 30 MHz

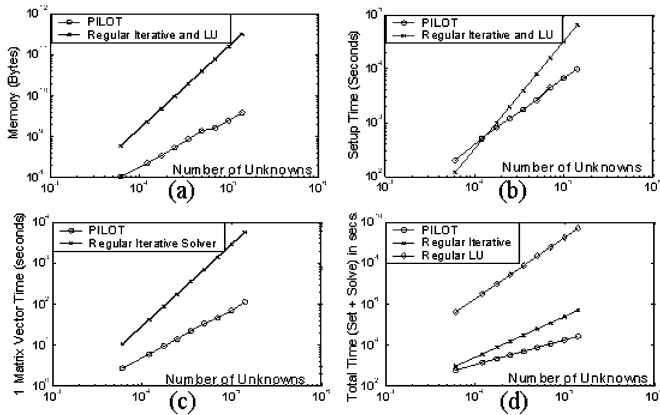


Fig. 7. (a) Memory requirement. (b) Matrix setup time. (c) Matrix vector product time. (d) Total time required.

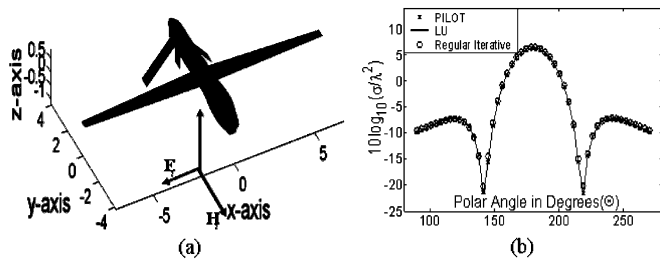


Fig. 8. (a) Surface mesh for airborne drone. (b) The bistatic RCS of the drone.

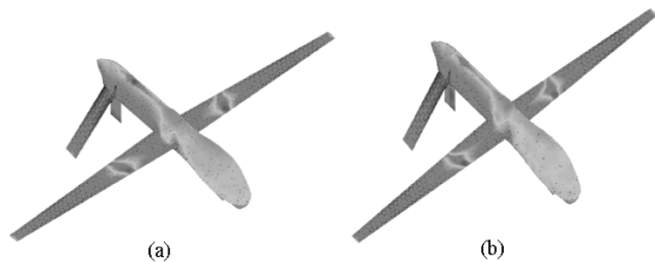


Fig. 9. (a) The current density with LU. (b) The current density with PILOT.

as obtained for $\phi = 0^\circ$ using the three methods is plotted in Fig. 8(b). The backscattering angle corresponds to $(\theta = 180^\circ)$. As seen in the figure, there is excellent agreement between the methods.

The current density on the structure at 30 MHz, obtained using the regular LU solver and PILOT is plotted on a log scale in Fig. 9(a) and (b), respectively.

The time and memory requirements for the solution process are demonstrated in Table VI.

The largest problem size addressed using PILOT consists of 217912 RWG edges on a test chip [Fig. 10(a)] comprising of multiple meander lines, coplanar waveguides, inductors, and capacitors on which S-parameter computations are performed. The solution process requires 3.5 GB memory, 98 min setup time, and 93.4 s per matrix vector product.

VII. FREQUENCY ANALYSIS

The QR-based algorithms degrade in compression efficiency as the structure size becomes much larger compared to the wavelength. More specifically, for significant compression

TABLE VI
TIME AND MEMORY REQUIREMENTS

	Memory	Setup Time (sec)	Time per matvec (sec)	Solve Time (sec)
Regular LU	1659MB	257	-	81696
Regular Iterative	1659MB	257	29	7132
PILOT Iterative	219MB	206	5.21	1334

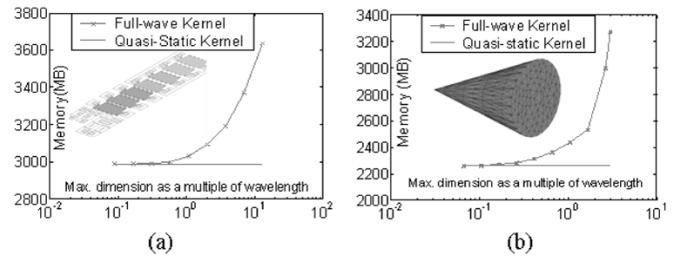


Fig. 10. (a) Memory required by PILOT for a 2-D structure. The corresponding MoM memory requirement is 540 GB. (b) Memory required by PILOT for a 3-D structure. The corresponding MoM memory requirement is 150 GB.

efficiency, the variation in the kernel should be primarily due to the $1/|\mathbf{r} - \mathbf{r}'|$ decay rather than the $e^{-jk|\mathbf{r} - \mathbf{r}'|}$ oscillation [14]. If in a given interaction, the source and observer groups are much larger compared to the wavelength, the rank of such an interaction matrix increases. The rate of increase of the rank depends on the type of the structure under consideration. For a predominantly 1-D structure, there is no increase in rank and hence no drop in efficiency. Three-dimensional source and observer groups exhibit a faster rate of increase in their interaction rank as compared to their 2-D counterparts.

The degradation of overall performance is demonstrated for 2-D and 3-D structures in Fig. 10. The 2-D structure is a single-layer test chip [shown as an inset in Fig. 10(a)] for which S-parameters are computed using an EFIE formulation. The structure is discretized using 191 390 unknowns and the memory requirement is plotted as a function of the electrical dimension of the structure in Fig. 10(a). The 3-D structure under consideration is the test structure 2 of the previous section. The memory requirement for the structure is plotted with increasing frequency in Fig. 10(b).

The memory requirement of the uncompressed matrix for the 2-D structure [Fig. 10(a)] is 540 GB and that for the 3-D structure is 150 GB. Hence, in spite of the apparent gradual degradation in performance, the algorithm is still a viable method of compression for structures with dimensions of multiple wavelengths. For structures described by complex surface meshes involving multiple length scales, such as those encountered in adaptive mesh refinement, the “electrically small” instability of classical FMMs may be exhibited for some interactions even if the overall structure is not electrically small. Therefore, for such problems, PILOT is a preferable alternative that will obviate any low-frequency instability. It should be noted for completeness that, owing to the high frequency behavior as depicted

in Fig. 10(a) and (b), PILOT or IES³ multilevel-QR methods will not exhibit asymptotic $O(N \log N)$ complexity for surface meshes that are scaled using Helmholtz (rather than Laplace) scaling. Also, a single-level QR method such as [18] with a purported complexity of $O(N^{1.5})$ will degrade to an asymptotic quadratic cost owing to the reasons discussed earlier.

VIII. CONCLUSION

In this paper, a multilevel scheme for QR-based matrix compression and subsequent fast iterative solution is presented. The method is based on exploiting the regular geometry structure of oct-tree decomposition along with merged-interactions to achieve a predetermined block-matrix structure optimized for maximum compression. Since the algorithm obviates the necessity for splits and merges akin to the binary tree-based QR methods, the setup time is minimized. The upper bound complexity estimate is derived and the algorithm is shown to scale as $O(N \log N)$ with the number of unknowns for electrically small structures. It should also be noted here that a key reason behind the compaction efficiency of QR-based algorithms like PILOT is the ability for customized storage of each submatrix. Each interaction submatrix, upon QR compression, yields a specific epsilon-rank, which is in most cases much smaller than the maximum epsilon-ranks (Table I). Therefore, it can be expected that the actual cost for PILOT is much smaller than the upper bound complexity estimate presented here. This is in contrast to FMM approaches where the number of wave expansions is fixed for all cubes independent of the distribution of basis functions within. The tradeoff is that the same multipole or plane-wave expansions are used for all intra-level interactions whereas PILOT compresses interactions on a pair-wise group basis.

The method is demonstrated as being amenable to fast full-wave solution for moderately sized scatterers of arbitrary mesh complexity and shape. PILOT is kernel independent and hence can potentially be applied to solve multilayered dielectric problems. Since it deals with a predetermined interaction list and obviates the necessity of tree traversals during the matrix setup and solve, PILOT demonstrates reduced communication overheads and is, therefore, readily amenable to parallelization unlike IES³ or fast multipole based methods.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. Burke and Dr. J. Rockway of Lawrence Livermore National Laboratory and Dr. J. W. Rockway of the Space and Naval Warfare Systems Center, San Diego, CA, for their inspiration behind this work and for the generation of meshes.

REFERENCES

- [1] R. F. Harrington, *Field Computation by Moment Methods*. New York: IEEE, 1991.
- [2] S. M. Rao, D. R. Wilton, and A. W. Glisson, "Electromagnetic scattering by surfaces of arbitrary shape," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 3, pp. 409–418, May 1982.
- [3] R. E. Hodges and Y. Rahmat-Samii, "The evaluation of MFIE integrals with the use of vector triangle basis functions," *Microw. Opt. Tech. Lett.*, vol. 14, no. 1, pp. 9–14, Jan. 1997.
- [4] A. F. Peterson, S. L. Ray, and R. Mittra, *Computational Methods for Electromagnetics*, ser. IEEE/OUP Series, 1998.
- [5] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 10, pp. 1447–1459, Nov. 1991.
- [6] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Trans. Antennas Propag. Mag.*, vol. 35, no. 6, pp. 7–12, Jun. 1993.
- [7] W. C. Chew, J. M. Jin, E. Michielssen, and J. Song, *Fast Efficient Algorithms in Computational Electromagnetics*. Boston, MA: Artech House, 2001.
- [8] Y. C. Pan, W. C. Chew, and L. X. Wan, "A fast multipole-method-based calculation of the capacitance matrix for multiple conductors above stratified dielectric media," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 3, pp. 480–490, Mar. 2001.
- [9] L. J. Jiang and W. C. Chew, "Modified fast inhomogeneous plane wave algorithm from low frequency to microwave frequency," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, vol. 2, Jun. 2003, pp. 22–27.
- [10] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems," *Radio Sci.*, vol. 31, pp. 1225–1251, Sep.–Oct. 1996.
- [11] J. R. Phillips and J. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, pp. 1059–1072, Oct. 1997.
- [12] N. Yuan, T. S. Yeo, X. C. Nie, and L. W. Li, "A fast analysis of scattering and radiation of large microstrip antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2218–2226, Sep. 2003.
- [13] S. Kapur and D. Long, "IES³: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 448–455.
- [14] S. Kapur and D. E. Long, "IES³: Efficient electrostatic and electromagnetic solution," *IEEE Comput. Sci. Eng.*, vol. 5, no. 4, pp. 60–67, Oct.–Dec. 1998.
- [15] S. Kapur, D. Long, and J. Zhao, "Efficient fullwave simulation in layered lossy medium," in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1998, pp. 211–214.
- [16] D. Gope, S. Chakraborty, and V. Jandhyala, "Enhanced efficiency, hybrid FMM-QR fast parasitic extractor for conductors and dielectrics," in *Proc. IEEE/ACM Design and Automation Conf.*, 2004, pp. 794–799.
- [17] S. M. Seo and J. F. Lee, "A single-level low rank IE-QR algorithm for pec scattering problems using EFIE formulation," *IEEE Trans. Antennas Propag.*, vol. 52, no. 8, pp. 2141–2146, Aug. 2004.
- [18] R. J. Burkholder and J. F. Lee, "Fast dual-MGS block-factorization algorithm for dense MoM matrices," *IEEE Trans. Antennas Propag.*, vol. 52, no. 7, pp. 1693–1699, Jul. 2004.
- [19] L. Tsang and Q. Li, "Wave scattering with UV multilevel partitioning method for volume scattering by discrete scatterers," *Microw. Opt. Technol. Lett.*, vol. 41, no. 5, pp. 354–361, Jun. 2004.
- [20] A. Breuer, P. Borderies, and J. L. Poirier, "A multilevel implementation of the QR compression for method of moments," *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2520–2522, Sep. 2003.
- [21] E. Michielssen and A. Boag, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures," *IEEE Trans. Antennas Propag.*, vol. 44, no. 8, pp. 1086–1093, Aug. 1996.
- [22] J. Parron, J. M. Rius, and J. R. Mosig, "Application of the multilevel matrix decomposition algorithm to the frequency analysis of large microstrip antenna arrays," *IEEE Trans. Magn.*, pt. 1, vol. 38, no. 2, pp. 721–724, Mar. 2002.
- [23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1989.
- [24] Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856–869, 1986.
- [25] C. J. Reddy, M. D. Deshpande, C. R. Cockrell, and F. B. Beck, "Fast RCS computation over a frequency band using method of moments in conjunction with asymptotic waveform evaluation technique," *IEEE Trans. Antennas Propag.*, vol. 46, no. 8, pp. 1229–1233, Aug. 1998.
- [26] J. S. Zhao and W. C. Chew, "Integral equation solution of Maxwell's equations from zero frequency to microwave frequencies," *IEEE Trans. Antennas Propag.*, vol. 48, no. 10, pp. 1635–1645, Oct. 2000.
- [27] R. P. Penno, G. A. Thiele, and K. M. Pasala, "Scattering from a perfectly conducting cube," *Proc. IEEE*, vol. 77, no. 9, pp. 815–823, 1989.



Dipanjan Gope (S'00) received the B.Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology (IIT), Kharagpur, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 2003 and 2005, respectively.

He has been a Research Assistant with the Applied Computational Electromagnetics Laboratory, University of Washington, and has held a summer internship position at the IBM T. J. Watson Research Center. Currently, he is with the Design and Technology Solutions Department, Intel Corporation. His research interests include fast solver algorithms, integral equation formulations, computational electromagnetic solvers for circuit applications, and signal integrity analysis.



Vikram Jandhyala (M'00–SM'03) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT), Delhi, in 1993, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign, in 1995 and 1998, respectively. As part of his graduate work, he codeveloped the steepest descent fast-multipole method for rapid simulation of a large class of EM problems.

From 1998 to 2000, he was a Research and Development Engineer with the Ansoft Corporation, Pittsburgh, PA. He was involved in the acceleration of Ansoft's integral-equation solvers, and codeveloped a fast multipole-based extraction tool in Ansoft's Q3D versions released in 1999 and 2000. Since 2000, he has been an Assistant Professor with the electrical Engineering Department, University of Washington, Seattle. He directs the Applied Computational Electromagnetics Laboratory, with research interests and projects in several areas of computational electromagnetics, including fast solvers and integral-equation formulations in the frequency and time domains, high-speed circuits and devices, coupled multiphysics simulation, novel materials, and propagation. He has visiting research status with the Lawrence Livermore National Laboratories. He has authored or coauthored more than 70 journal and conference papers.

Dr. Jandhyala is a full elected member of the International Scientific Radio Union (URSI) Commission B. He has served as a reviewer for several IEEE journals and conferences and national and international proposal panels. He is on the Technical Program Committee of the IEEE Design Automation Conference and the IEEE Antennas and Propagation Society (IEEE AP-S) Symposium. He was a recipient of the 2001 National Science Foundation (NSF) CAREER grant, a 1998 Outstanding Graduate Research Award presented by the University of Illinois, and a 1996–1997 IEEE Microwave Graduate Fellowship.