Memory-Efficient Multilevel Physical Optics Algorithm for Fast Computation of Scattering From Three-Dimensional Complex Targets

Alp Manyas [1,2] and Levent Gürel [1,2]

[1]*Department of Electrical and Electronics Engineering*
[2]*Computational Electromagnetics Research Center (BiLCEM)*
Bilkent University, TR-06800, Bilkent, Ankara, Turkey
{alp, lgurel}@ee.bilkent.edu.tr

***Abstract*— Multilevel physical optics (MLPO) algorithm provides a speed-up for computing the physical-optics integral over complex bodies for a range of aspect angles and frequencies. On the other hand, when computation of the RCS pattern as a function of θ, ϕ, and frequency is desired, the $O\left(N^3\right)$ memory complexity of the algorithm may prevent the solution of electrically large problems. In this paper, we propose an improved version of the MLPO algorithm, for which the memory complexity is reduced to $O\left(N^2 \log N\right)$. The algorithm is based on the aggregation of only some portion of the scattering patterns at each aggregation step. This way, memory growth in each step is prevented, and a significant amount of saving is achieved.**

I. Introduction

For the computation of electromagnetic scattering from electrically large targets, physical-optics (PO) technique can provide very fast solutions. On the other hand, in real-life radar applications, where the computation of the scattering pattern over a range of frequencies and/or angles with sufficient number of samples is desired, further acceleration may be needed. Multilevel physical optics (MLPO) algorithm [1], [2] can be used for such applications, so that a remarkable speed-up can be achieved by evaluating the PO integral in a multilevel fashion. One of the drawbacks of this algorithm is its rapidly growing memory requirement for three-dimensional (3-D) applications. In this paper, a memory-efficient multidimensional MLPO algorithm, with which the scattering pattern of a 3-D target can be evaluated as function of elevation angle (θ), azimuth angle (ϕ), and frequency with a lower memory complexity, is presented. In this algorithm, memory complexity of $O\left(N^3\right)$ is reduced to $O\left(N^2 \log N\right)$.

II. MLPO Algorithm

A. Computation Time

When computing the PO scattering pattern with sufficient number of samples as a function of θ, ϕ, and frequency, sampling rate in each dimension should be proportional to the electrical size of the target. Let R be the smallest radius of a sphere that can contain the target and $N = kR$, where k is the wavenumber. Then, the required number of samples in θ, ϕ, and frequency are $O\left(N\right)$ each and the total number of required samples is $O\left(N^3\right)$. If the target surface is modeled with a triangular mesh, there will be $O\left(N^2\right)$ triangles as the number of triangles will be proportional to the surface area. Hence, the computational complexity of evaluating PO integral analytically [3] on the triangular mesh for each θ, ϕ, and frequency turns out to be $O\left(N^5\right)$.

Since the origin of the each subdomain will be different from the global origin, subdomain patterns will oscillate at higher rates because of the phase shift. Therefore, prior to interpolation, subdomain patterns should be shifted to the origin and restored after interpolation. In this scheme, PO operator Ψ that computes the scattering pattern of any arbitrary surface S can be written as

$$\Psi S = \sum\nolimits_{q=1}^{Q} E\left[\bar{\mathbf{r}}_q\right] \mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f} O\left[\bar{\mathbf{r}}_q\right] \Psi \bar{S}_q. \tag{1}$$

Here, $\bar{\mathbf{r}}_q$ is the center of the smallest sphere that can include the qth subdomain. $O\left[\bar{\mathbf{r}}_q\right]$ is the operator that shifts the origin of the qth subdomain to the global origin in order to remove the phase oscillations. The $\mathbf{I}_{\bar{N}_\theta \bar{N}_\phi \bar{N}_f}^{N_\theta N_\phi N_f}$ matrix is the interpolation matrix that increases the number of samples from $\bar{N}_\theta \times \bar{N}_\phi \times \bar{N}_f$ points to $N_\theta \times N_\phi \times N_f$ points, and $E\left[\bar{\mathbf{r}}_q\right]$ is the operator that shifts the origin of the qth subdomain back to its location after the interpolation.

In the MLPO algorithm, each subdomain is recursively subdivided into smaller subdomains and the scattering patterns of these subdomains are also calculated via MLPO. When the subdomain size is in the order of λ (wavelength), the subdivision process can be stopped and the scattering patterns of the lowest-level subdomains can be evaluated with the PO integral.

As the electrical size of the bottom-level subdomains will be bounded, the required number of θ, ϕ, and frequency samples will be fixed for each subdomain at this level. There will be $O\left(N^2\right)$ filled subdomains at this level and therefore calculating the PO patterns of all subdomains analytically at this level will require $O\left(N^2\right)$ operations. At each aggregation step, local interpolations transforming the scattering patterns form a coarse grid of θ, ϕ, and frequency to a finer grid will require $O\left(N^3\right)$ operations. As there will be $O\left(\log N\right)$ levels, total computational cost of aggregations will be $O\left(N^3 \log N\right)$.

978-1-4244-1605-9/07/$25.00 ©2007 IEEE

Therefore, the overall complexity of the MLPO algorithm is $O\left(N^2\right)+O\left(N^3\log N\right)=O\left(N^3\log N\right)$. This complexity is far less than the $O\left(N^5\right)$ complexity of the conventional PO integration.

B. Memory Requirement

The MLPO algorithm requires $O\left(N^3\right)$ memory to store the radiation patterns for all three dimensions when aggregating from lower levels to the upper levels, since the sampling rates in each of the three dimensions will be doubled. Therefore, the required memory for each cluster will grow by a factor of 8. Since PO current is only on the surface of the target, we can assume that the number of filled clusters is $O\left(N^2\right)$ at the bottom level. We can also assume that the number of filled clusters is reduced by a factor of 4 at each higher level. Therefore, the memory required at each aggregation step will increase by a factor of $8/4=2$. Hence, at the uppermost level, the memory requirement will be $O\left(N^2 2^{\log N}\right)$, which is $O\left(N^3\right)$. As will be shown in Section IV, this memory requirement may prevent the solution of larger problems using the MLPO algorithm. In the next section, we present a memory-efficient implementation that reduces the memory complexity from $O\left(N^3\right)$ to $O\left(N^2\log N\right)$.

III. Memory-Efficient MLPO Algorithm

The proposed memory-efficient implementation of the MLPO algorithm is based on the idea that the patterns of the clusters need not be stored for the entire range of θ, ϕ, or frequency values at the same time. By careful implementation, the θ, ϕ, or frequency ranges can be divided into smaller ranges so that interpolations and aggregations can be performed on those smaller ranges.

For instance, when aggregating the bottom level to the upper level, θ can be sampled in the $\left[0,\frac{\pi}{2}\right]$ range instead of $[0,\pi]$. This way, the memory required for each cluster will grow by a factor of 4 instead of 8. In the next level, the number of clusters will be reduced by a factor of 4, and the total required memory will be constant. Then, θ can be sampled in the $\left[0,\frac{\pi}{4}\right]$ range instead of $\left[0,\frac{\pi}{2}\right]$ at the next level, and this procedure can be applied till the uppermost level is reached. This scheme is illustrated in Fig. 1.

If L is the number of levels, after the aggregations are performed, $1/2^{L-1}$ portion of whole scattering pattern will be available at the Lth level. This portion can be output to a file and the remaining portion can be obtained by aggregating the second half of the pattern available at the $(L-1)$st level to the Lth level. Then, the 3rd portion of the pattern will be required. This portion will not be available at the $(L-1)$st level. Therefore, the pattern available at the $(L-2)$nd level should be aggregated to the $(L-1)$st level. Then, the 3rd portion of the whole scattering pattern at the Lth level can be obtained by aggregation from the $(L-1)$st level. In this scheme, after 2^{L-1} passes, the whole scattering pattern will be obtained. As an example, aggregations of pattern portions for a 6-level problem is depicted in Fig. 2

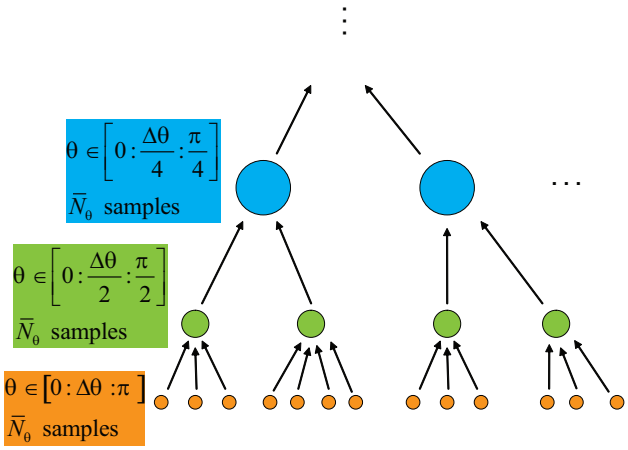

Fig. 1. Aggregation steps in the proposed memory-efficient algorithm. Note that $\theta \in [0:\Delta\theta:\pi]$ means that the pattern at that level is sampled from 0 to π with an increment of $\Delta\theta$

The following pseudo-code describes partial aggregation of the clusters at each pass:

{n is an array of size $L-1$, indicating which half of the available pattern portion should be aggregated at each pass.}
{In the first pass, first halves of the available pattern portions should be aggregated.}
$n(1:L) \leftarrow 1$
for $t=1$ to 2^{L-1} **do**
  for $l=1$ to $L-1$ **do** {all levels 1 to $L-1$}
    $m \leftarrow L-l+1$
    if mod $\left(t, 2^{m-2}\right)=1$ or $m=2$ **then**
      for all $cli \in l^{th}$ $level$ **do** {all clusters in lth level}
        {Aggregate the $n(l)$th half of the cluster.}
        $aggregate\_cluster(cli, n(l))$
      end for
      if $n(l)=1$ **then**
        $n(l) \leftarrow 2$
      else
        $n(l) \leftarrow 1$
      end if
    end if
  end for
  {Write the available portion of the whole targets scattering pattern.}
  $writeToFile$
end for

It should be noted that higher-order interpolation schemes may be desired in order to prevent interpolation error in aggregations [4]. In this case, additional sample points at the end points of the partial patterns should be included at the interpolations as illustrated in Fig. 3.

For the end points corresponding to $\theta=0$ or $\theta=\pi$, the samples corresponding to neighboring nodes on the unit sphere can be used since $(0-\alpha,\phi)=(\alpha,\phi+\pi)$ and $(\pi+\alpha,\phi)=(\pi-\alpha,\phi+\pi)$ on a unit sphere.

Dividing the ranges of other dimensions will reduce the required memory at each aggregation step but will not signif-

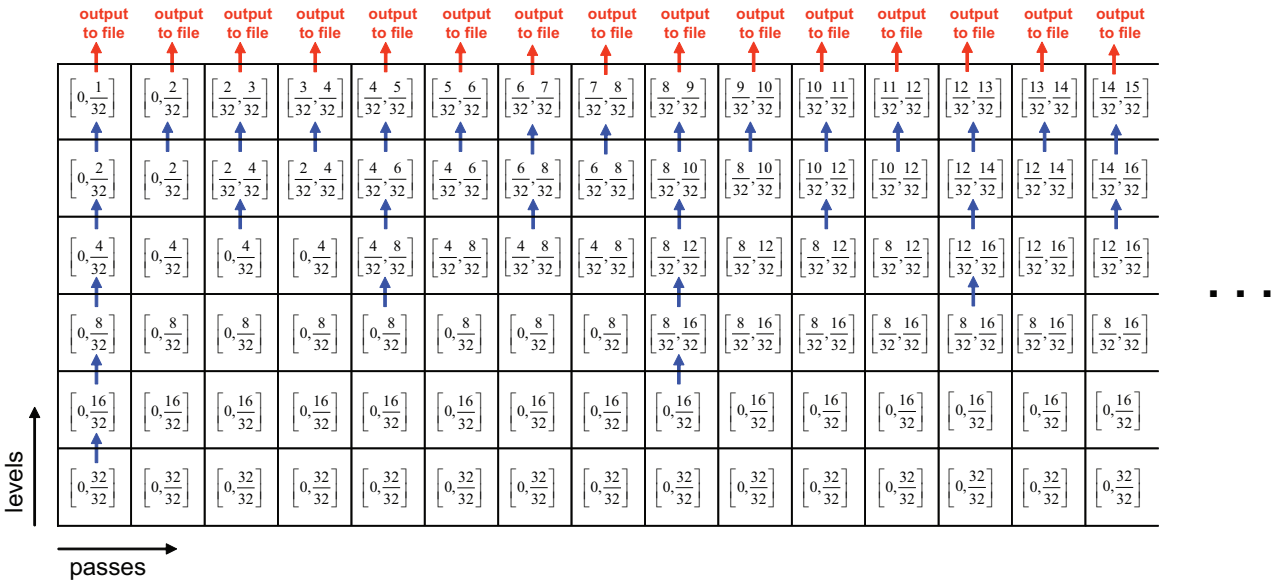

Fig. 2. Aggregations of pattern portions for a 6-level problem. Aggregations are shown as blue arrows and outputs to files are shown as red arrows.

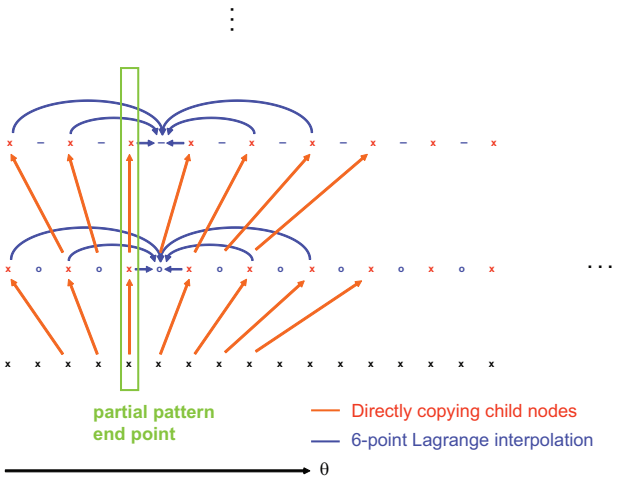

Fig. 3. Interpolating cluster patterns near the end points.

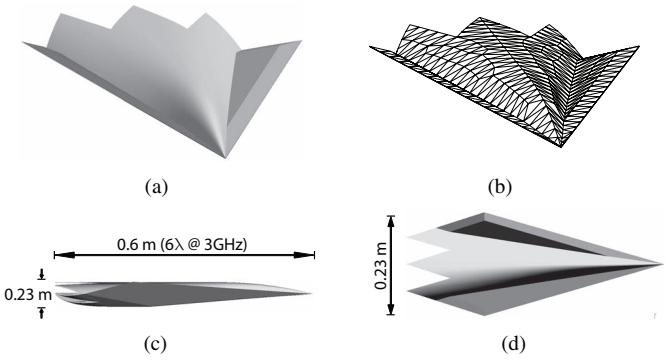

Fig. 4. Geometry of the stealth Flamme target: (a) Front view. (b) A uniform mesh example. (c) Rear view. (d) Top view.

TABLE I

GROWTH OF THE NUMBER OF TRIANGLES, θ SAMPLES, ϕ SAMPLES, AND FREQUENCY SAMPLES AS N INCREASES

$N/2\pi$ (Target Size/λ in Frequency Range)	Number of Triangles	N_θ	N_ϕ	N_f
$[0, 1.5]$	628	49	101	17
$[0, 3]$	1604	97	201	33
$[0, 6]$	5200	193	401	65
$[0, 12]$	19288	385	801	129
$[0, 24]$	75634	769	1601	257
$[0, 48]$	300020	1537	3201	513

icantly improve the memory efficiency. This is because the patterns of the bottom-level clusters dominate the memory requirement. On the other hand, aggregating the bottom-level clusters directly to the upper level without storing their patterns will reduce the required memory.

IV. NUMERICAL RESULTS

To demonstrate the accuracy and efficiency of the MLPO algorithm, backscattering RCS pattern of the scaled Flamme geometry [5] shown in Fig. 4 is computed for all directions on the unit sphere.

RCS values are evaluated for the frequency ranges given in Table I, and CPU times are compared in Figure 7. For the sake of simplicity, first column of Table I is given as $N/2\pi$, which is the electrical size of the target in λ.

From the backscattering RCS results shown in Fig. 5, it is observed that, on the x-y, x-z, and y-z planes and in the 0–3 GHz frequency range, the MLPO algorithm and direct PO evaluation results are in excellent agreement. Fig. 6 presents the RCS error, $10 \log |RCS_{MLPO} - RCS_{PO}|$, which is below 1%.

From Table I, it is observed that, as N increases, the number of triangles grows with $O\left(N^2\right)$ for large N. Numbers of θ,

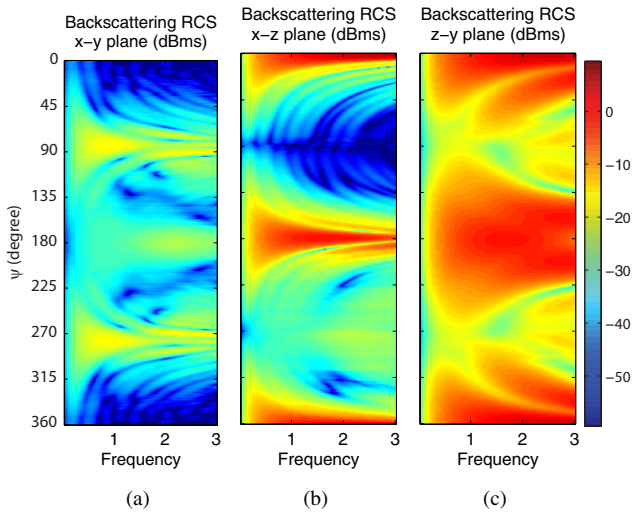

Fig. 5. Backscattering RCS pattern of the Flamme geometry: (a) x-y, (b) x-z, and (c) z-y planes.

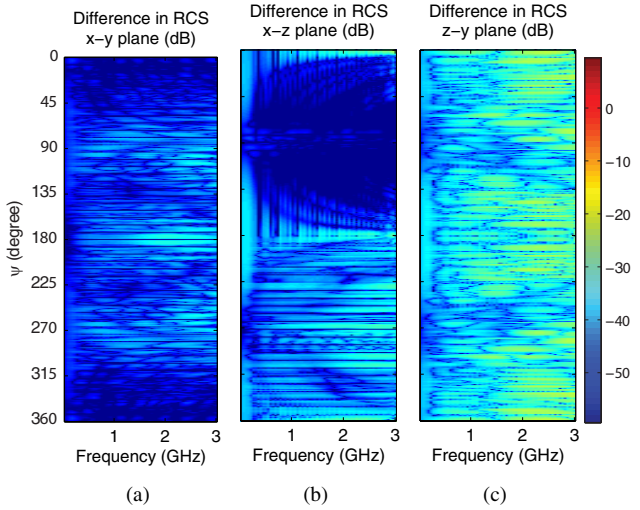

Fig. 6. Absolute error of the MLPO algorithm: (a) x-y, (b) x-z, and (c) z-y planes.

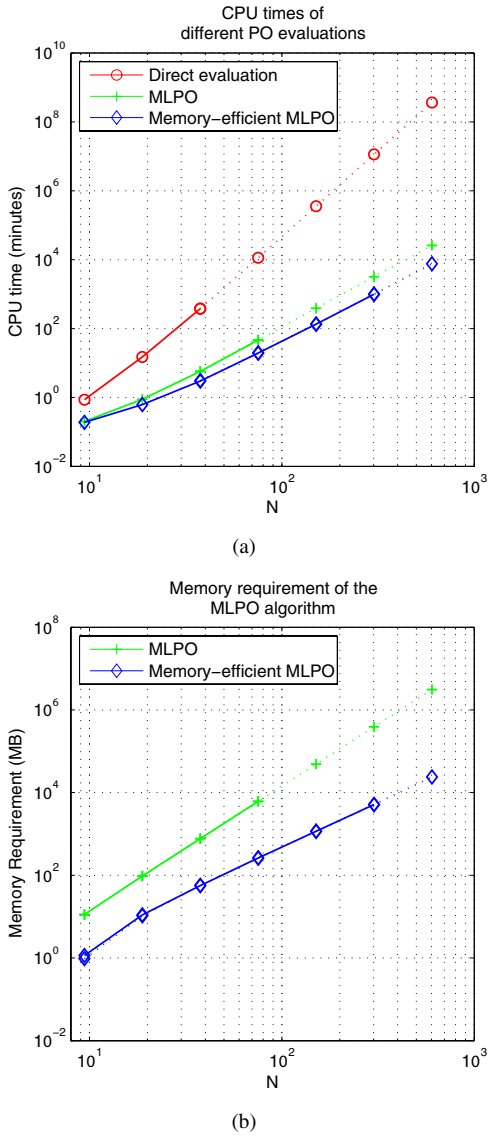

Fig. 7. Efficiency of the MLPO algorithm: (a) CPU time. (b) Memory requirement. Dashed curves represent estimated values.

ϕ, and frequency samples (N_θ, N_ϕ, and N_f, respectively) increase with $O(N)$. Therefore, the total complexity of the direct PO evaluation turns out to be $O(N^5)$. This can be verified from the computation times presented in Fig. 7. Since both axes are in log scale, slopes of the curves indicate the complexity. For instance, $\log(N^5) = 5\log(N)$ and when plotted versus $\log(N)$, the curve is a straight line with slope 5. Similarly, $\log(N^3\log(N)) = 3\log(N) + \log(\log N) \approx 3\log(N)$ and when plotted versus $\log(N)$, the curve is approximately a straight line with slope 3.

For the 48λ problem presented in Table I and Fig. 7, computing the scattering pattern with the memory-efficient MLPO algorithm takes approximately 53 hours. Computation time of the same problem with the direct PO integration is estimated to be 180,000 hours. Thus, a speed-up of nearly 3000 can be achieved with the MLPO algorithm. On the other hand, conventional MLPO algorithm would require 388 GB of memory, whereas the memory-efficient MLPO algorithm requires only 5 GB of memory, which is 77 times more efficient.

V. Conclusions

For 3-D problems, a memory-efficient MLPO algorithm is proposed to evaluate the PO integral, so that the memory complexity of $O(N^3)$ is reduced to $O(N^2\log N)$. The backscattering RCS pattern of a stealth geometry is calculated via the proposed algorithm and the results are compared with the direct PO evaluation in order to verify the accuracy and efficiency of the algorithm. For a 48λ problem, it is shown that the MLPO algorithm can achieve a speed-up of more than 3000 at the cost of requiring 388 GB of memory. In addition to achieving a similar speed-up, the memory-efficient MLPO

algorithm can further reduce the memory requirement by a factor of 77, thus enabling the solution of the same problem with 5 GB of memory.

ACKNOWLEDGMENT

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Research Grant 105E172, by the Turkish Academy of Sciences in the framework of the Young Scientist Award Program (LG/TUBA-GEBIP/2002-1-12), and by contracts from ASELSAN and SSM.

REFERENCES

[1] A. Boag, "A fast physical optics (FPO) algorithm for high frequency scattering," *IEEE Trans. Antennas Propagat.*, vol. 52, pp. 197–204, Jan. 2004.

[2] A. Boag and E. Michielssen, "Fast physical optics (FPO) algorithm for double-bounce scattering," *IEEE Trans. Antennas Propagat.*, vol. 52, pp. 205–212, Jan. 2004.

[3] W. B. Gordon, "Far-field approximations to the Kirchhoff-Helmholtz representation of scattered fields," *IEEE Trans. Antennas Propagat.*, vol. 23, pp. 590–592, July 1975.

[4] A. Manyas and L. Gürel, "Multilevel PO algorithm for non-uniform triangulations," *European Conference on Antennas and Propagation (EuCAP 2006)*, Nice, France, Nov. 2006.

[5] L. Gürel, H. Bağcı, J. C. Castelli, A. Cheraly, and F. Tardivel, "Validation through comparison: measurement and calculation of the bistatic radar cross section (BRCS) of a stealth target," *Radio Science*, vol. 38, no. 3, pp. 12-1–12-10, June 2003.