Accelerated Direct Flux Calculations Using an Adaptively Refined Icosahedron

Paul Manstetten*, Andreas Hössinger‡, Josef Weinbub*, and Siegfried Selberherr†
*Christian Doppler Laboratory for High Performance TCAD at the
†Institute for Microelectronics, TU Wien, Gußhausstraße 27-29/E360, 1040 Wien, Austria
‡Silvaco Europe Ltd., Compass Point, St Ives, Cambridge, PE27 5JL, United Kingdom
Email: manstetten@iue.tuwien.ac.at

***Abstract*—We present a method to accelerate the direct flux calculations in three-dimensional process simulations, highly relevant for plasma etching simulations. Especially for high aspect ratio structures, the direct flux calculations are the computational bottleneck when performing simulations with high spatial resolutions. Our method reduces the number of rays which have to be traced during the visibility calculation. A subdivided icosahedron is used to define the search directions on each level of refinement. The method is applicable if the apertures can be detected on the first level.**

We show that the accuracy of the integration is not reduced, when using a first level of refinement appropriate to the aspect ratio of the structure. The time for the visibility calculation is cut by 50% for meshes with more than 200k triangles when using only one level of refinement.

I. Introduction

If the model for the surface velocity in a three-dimensional process simulation depends on the flux originating from one or more remote sources above the structure, typically either a *top-down* or a *bottom-up* approach is used to obtain the flux rates for all points on the surface. The *top-down* approach casts the flux contributions from the source towards all surface points, commonly using a Monte Carlo ray tracing algorithm [1]. The *bottom-up* approach integrates the flux contribution from the source towards a surface point by (a) finding all solid angles that are not obstructed by the geometry, and (b) integrating the source contribution over these solid angles, which yields the surface velocity. Independent of the approach chosen, the accurate calculation of these surface velocities is the main computational bottleneck in three-dimensional process simulations, especially for high aspect ratios which are increasingly required as device structures move towards full three-dimensional designs, e.g., NAND flash cells [2].

We present a method that reduces the number of rays which have to be traced during the detection of the visible directions in the *bottom-up* approach. In [3], we recently presented an approach that uses an explicit representation of the surface to perform the visibility calculations using a subdivided icosahedron to define the search and integration directions. In this work, we present an adaptive algorithm that uses the hierarchy, given by the subdivisions, to refine the search directions only where necessary, thereby significantly increasing the efficiency.

The adaptive sampling scheme is illustrated for two dimensions in Fig. 1 and Fig. 2 visualizes the search directions in three-dimensions, together with the subdivided icosahedrons which are used to define the search directions.

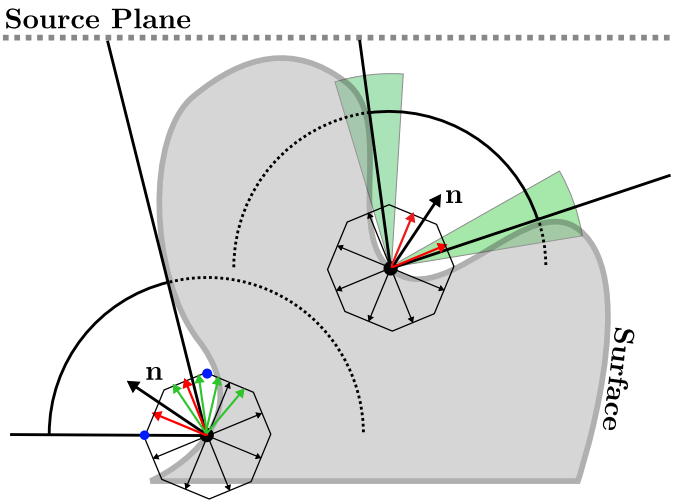


Fig. 1. Two-dimensional illustration of the *bottom-up* visibility scheme for direct flux calculation for two surface points. The search directions are colored according to the source visibility (red = visible, black = obscured). In three dimensions, the search directions are towards the centers of the triangles of the icosahedron. The green solid angles indicate the neighborhood of the aperture boundary. The blue points are identified as aperture boundary points which are used to define the regions for re-evaluation on the next level. The green arrows show the search direction on the next level for one of the boundary points.

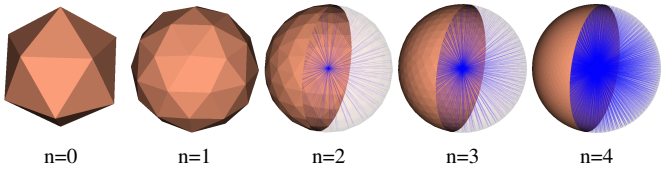


Fig. 2. Initial icosahedron (n=0) and subdivisions up to n=4. Starting from n=2, the integration directions originating at the center and pointing towards each centroid of the triangles of the spherical mesh are shown (blue rays).

II. Adaptive Visibility Scheme

For each subdivision level of the icosahedron, the search directions are defined towards the centers of the triangles of the icosahedron. The subdivision described in [3] divides each triangle into 4 triangles on the next level which cover the same area when projected onto the unit sphere (cf. Fig. 2). Instead

of evaluating all directions on the final subdivision level (*maxlevel*), our adaptive scheme starts on a lower subdivision level (*minlevel*). After evaluating all directions on *minlevel*, the boundary of the aperture is identified and all directions in the neighborhood of the boundary are re-evaluated on the next level. The aperture boundary is defined as all triangles which are connected to a vertex which shares triangles with mixed visibility information (cf. blue points in Fig. 1). This procedure is repeated until *maxlevel* is reached.

After the detection of the visible directions using the adaptive visibility scheme, the integration is performed with the resolution of *maxlevel*. We use a centroid rule [4] to approximate the flux contributed from each search direction j at surface position i.

$$F_{i_j} = \int_{\Delta_j} f_{src}(\mathbf{\Theta})(\mathbf{\Theta}\mathbf{n_i})dA = f_{src}(\mathbf{\Theta_{c_j}})(\mathbf{\Theta_{c_j}}\mathbf{n_i}) \cdot \Delta_j \quad (1)$$

$\mathbf{\Theta_{c_j}}$ is the direction towards the centroid of triangle j and Δ_j is the area of triangle j, and $\mathbf{n_i}$ is the surface normal.

Algorithm 1 describes the adaptive visibility sampling for an integration point on the surface including the integration on the maximum refinement level.

```
Input  : minlevel, maxlevel
Output: flux at surface point i
for n = minlevel to maxlevel do
    if n == minlevel then
        set all directions to be active on level n;
        set all visibility information to false level on n;
    end
    foreach direction d on level n do
        if d is active then
            trace ray into direction of d;
            if ray hit source then set visibility for d to true;
            if ray hit surface then set visibility for d to false;
        end
    end
    if n < maxlevel then
        foreach direction d on level n + 1 do
            set visibility for d according to parent on level n
        end
        foreach vertex v on level n do
            if v is boundary vertex then
                foreach triangle t connected to v do
                    mark all children of t active on next level
                end
            end
        end
    end
    if n == maxlevel then
        foreach direction d on level n do
            if visibility of direction d is true then
                integrate flux contribution (cf. (1));
                add contribution to surface point i;
            end
        end
    end
end
```

Algorithm 1: *Bottom-up* direct flux calculation at surface position i using our adaptive visibility scheme and integration on the maximum refinement level.

III. Results

We evaluated the algorithm by using a cylindrical hole with aspect ratios between 1 and 25 and resolutions between 48x48 and 80x80 on the horizontal plane (Fig. 3); this results in vertical resolutions of up to 1100, which represents computationally challenging evaluation cases. To also cover possible variations in the simulated physical processes, e.g., plasma etching, we apply variations to the wall by extending and tapering the radius at the bottom by +-25%, similar to [5]. As in [3], we used *Embree* [6] as ray tracing engine and *OpenVDB* [7] for surface advection and extraction.

The choice for the *minlevel* used in our simulations depends on the aspect ratio of the geometry and is illustrated in Fig. 4. The *maxlevel* is set to 6 for all simulations; this corresponds to about 40k search directions per hemisphere.

All results were produced using a highly vertically focused ($n = 100$) power cosine source distribution $\Gamma(\Theta) = cos(\Theta)^n$. The properties of the power cosine distributions are visualized in Fig.5 for exponents $n = 1$ to $n = 100$. To verify that the accuracy of the integral indeed does not suffer, we compared the flux rates at various depths of the structure: the flux rates are identical, when applying the proposed algorithm.

Fig. 6a plots the variation of the calculated flux rates (amongst all points at a certain depth) for the cylindrical hole; the visible noise for high aspect ratios is not caused by the proposed algorithm but by the accuracy imposed by the integration performed on the *maxlevel*. Fig. 6b and Fig. 6c plot the variations of the tapered and extended geometry, respectively: Additional to the noise, a pattern is revealed for all aspect ratios. This pattern is produced by the perturbed normals of the explicit mesh, resulting from the extraction algorithm of *OpenVDB* [7]. Fig. 6d shows an enlarged view on one of the pattern locations: It can clearly be seen that the extracted quads (which are divided into triangles prior to the ray tracing) have perturbed normals, when the interface transits the location of a level-set grid point. This perturbation leads to a relative variation of the local flux up to 25% for the tested scenarios.

The speedups obtained when applying the algorithm for various combinations of configurations are summarized in Fig. 7: When only considering one level of refinement, a minimum speedup of 1.5 is obtained for aspect ratio 1. For surface meshes with more than 200k triangles (aspect ratios 15 and 25), the speedup is larger than 2 for all configurations.

Fig. 8 shows the cross section of a validation simulation with a linear surface reaction model $v = -F$ for the tapered structure (aspect ratio 6). In Fig. 8a, the impact of an insufficient maximum level of refinement is demonstrated: the aperture is detected but the accuracy of the integral is insufficient. In Fig. 8b, using *maxlevel* = 5, a small asymmetry

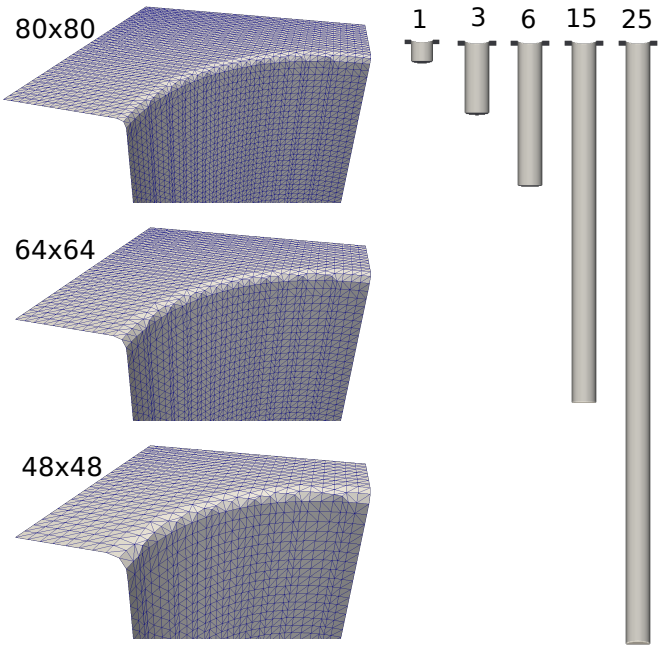


Fig. 3. Triangulated surfaces of the vertical geometry for aspect ratios between 1 and 25. The opening is shown in detail (cut view) for three different horizontal resolutions.

can still be detected at the bottom, which vanishes when using *maxlevel* = 6 (Fig. 8c).

IV. Summary and Outlook

We show that our bottom-up adaptive visibility scheme is able to sustain the accuracy of the direct flux integrals while reducing the integration time by 50% for larger meshes (>200k triangles), as long as all opening apertures are captured by the minimum level of refinement.

A possible extension of the algorithm is to also adapt the minimum refinement level depending on the search direction. Especially for high aspect ratio structures, this approach seems promising, as the source is almost completely obscured (beside nearly vertical search directions) for the majority of integrations points on the wall.

When using more than one level of refinement, the ratio between integration time and ray tracing time limits the obtainable speedup, as the integration is always performed on the maximum level of refinement. An integration scheme which adapts the resolution according to the properties of the angular source distribution (which is known beforehand) is an additional possibility to reduce run time while having limited impact on the integration accuracy.

The extraction algorithm for obtaining a mesh from the level-set representation perturbes the normals of the mesh, when the surface crosses a level-set grid point. The resulting flux variations are not inherent to our approach and could potentially be reduced by adopting or changing the extraction algorithm.

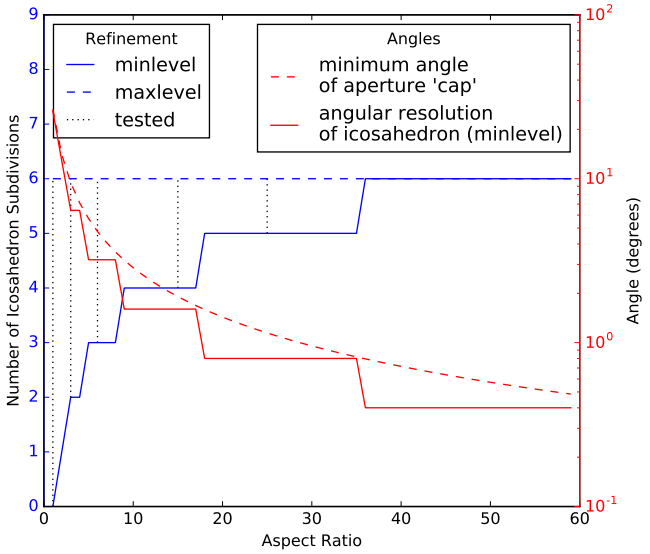


Fig. 4. Minimum angular resolution (dashed red) and corresponding subdivision level of the icosahedron (solid blue) to guarantee the detection of the aperture of the vertical geometry. The dotted vertical lines indicate the investigated aspect ratios 1, 3, 6, 15, and 25.

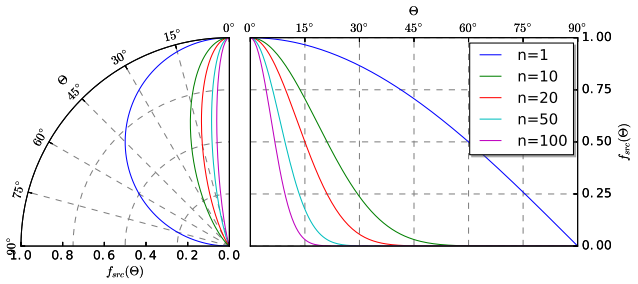


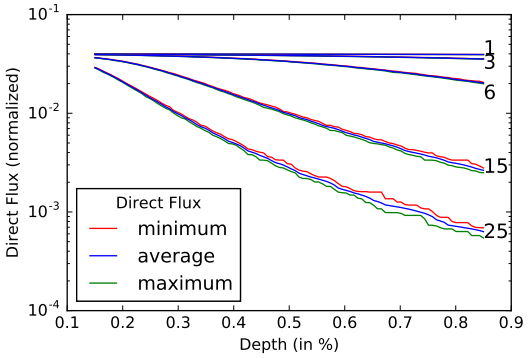
Fig. 5. Power cosine source distribution for exponents from diffuse ($n = 1$) very directed ($n = 100$) in polar (left) and cartesian (right) coordinates.

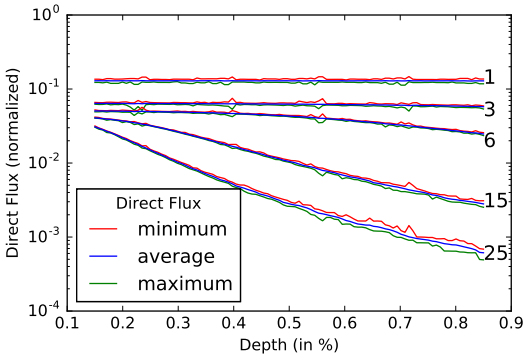
Acknowledgment

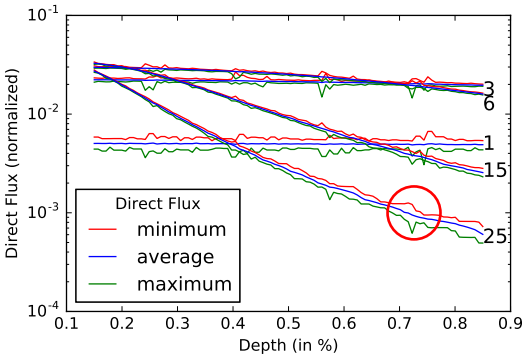
The financial support by the *Austrian Federal Ministry of Science, Research and Economy* and the *National Foundation for Research, Technology and Development* is gratefully acknowledged.

References

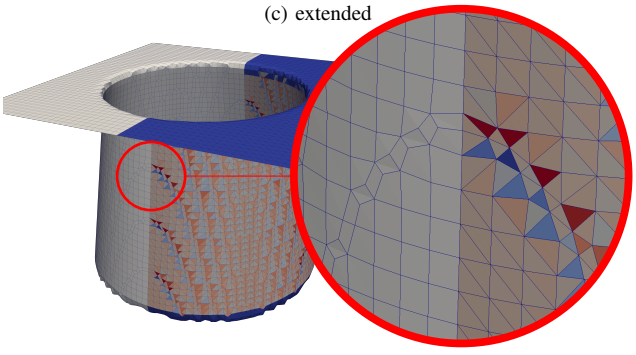
[1] O. Ertl and S. Selberherr, "Three-Dimensional Level Set Based Bosch Process Simulations Using Ray Tracing for Flux Calculation," *Microelectronic Engineering*, vol. 87, no. 1, pp. 20–29, 2010.
[2] P. Dimitrakis, *Charge-Trapping Non-Volatile Memories: Volume 1 – Basic and Advanced Devices*. Springer, 2015.
[3] P. Manstetten, J. Weinbub, A. Hössinger, and S. Selberherr, "Using Temporary Explicit Meshes for Direct Flux Calculation on Implicit Surfaces," *Procedia Computer Science*, vol. 108, pp. 245–254, 2017.
[4] K. Atkinson, "Numerical Integration on the Sphere," *The Journal of the Australian Mathematical Society. Series B. Applied Mathematics*, vol. 23, no. 03, pp. 332–347, 1982.
[5] P. Manstetten, L. Filipovic, A. Hössinger, J. Weinbub, and S. Selberherr, "Framework to Model Neutral Particle Flux in Convex High Aspect Ratio Structures Using One-Dimensional Radiosity," *Solid-State Electronics*, vol. 128, pp. 141–147, 2017.
[6] "Embree." [Online]. Available: https://embree.github.io/
[7] "OpenVDB." [Online]. Available: http://www.openvdb.org/







(c) extended



(d) explicit mesh

Fig. 6. Direct flux distributions for different wall configurations. Normalized direct flux along the wall of the vertical structure (a), tapered structure (b), and extended structure (c) originating from a power cosine source distribution ($n = 100$). While the vertical structure shows steadily increasing noise towards the bottom, the tapered and especially the extended structure reveal a pattern of increased noise (red circle). (d) shows an enlarged view on one of the pattern locations at around 75% of the total depth. The mesh is represented by quads (left, gray) and triangles (right, colored by flux).

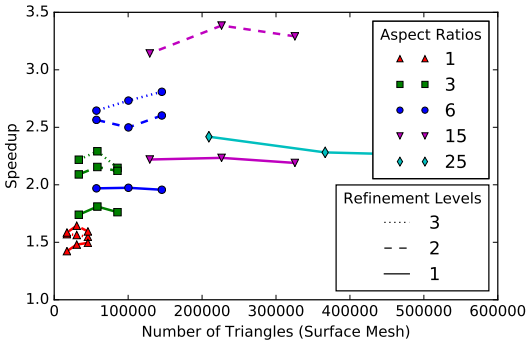
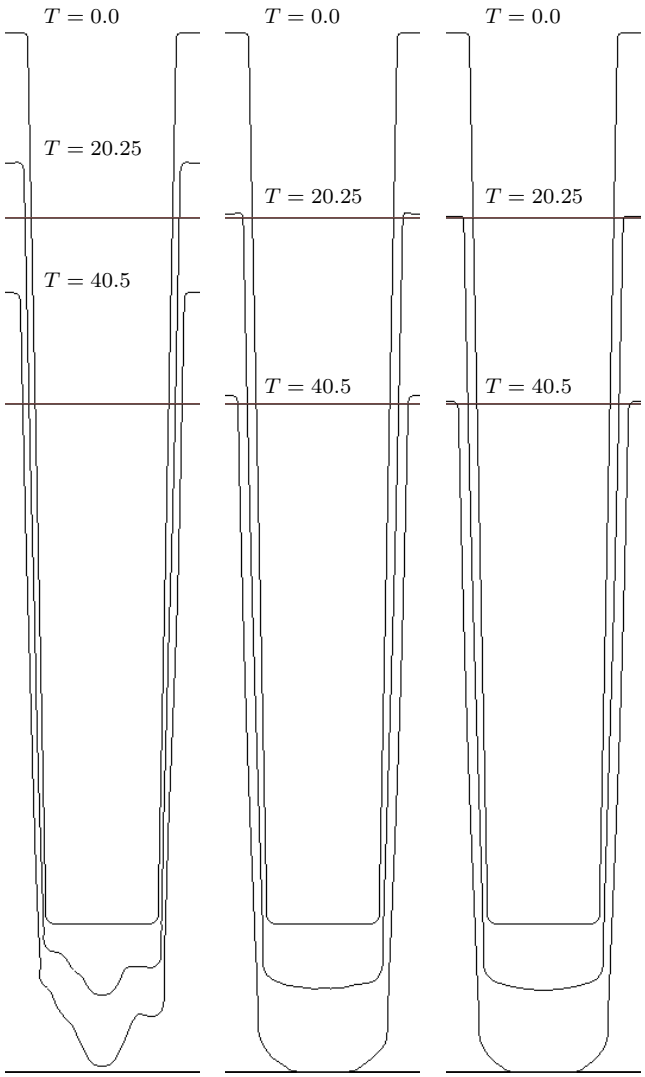


Fig. 7. Speedup obtained by applying the adaptive refinement for different surface mesh sizes and aspect ratios. The line style indicates the number of refinements; *maxlevel* is 6 for all results.



(a) *minlevel*=3, *maxlevel*=3

(b) *minlevel*=4, *maxlevel*=5

(c) *minlevel*=5, *maxlevel*=6

Fig. 8. Three-dimensional simulation results for a linear surface model $v = -F$. Cross section through the tapered structure of aspect ratio 6 from $T = 0$ to $T = 40.5$ for different refinement settings (a,b,c). The straight lines indicate the analytic solution for a horizontal surface.