

A Parallel Random Walk Solver for the Capacitance Calculation Problem in Touchscreen Design

Zhezha Xu¹, Wenjian Yu^{1,2}, Chao Zhang¹, Bolong Zhang^{1,3}, Meijuan Lu¹, Michael Mascagni³

¹Department of Computer Science and Technology, ²Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China. ³Departments of Computer Science, Mathematics, and Scientific Computing, Florida State University, Tallahassee, FL 32306, USA.

Email: zhezhaoxu@gmail.com, yu-wj@tsinghua.edu.cn, eric.3zc@gmail.com, blzhang.m@gmail.com, 857815138@qq.com, mascagni@fsu.edu.

ABSTRACT

In this paper, a random walk based solver is presented which calculates capacitances for verifying a touchscreen design. To suit the complicated conductor geometries in touchscreen structures, we extend the floating random walk (FRW) method for handling non-Manhattan conductors. A unified dielectric pre-characterization scheme is proposed to suit arbitrary dielectric profiles while keeping high accuracy. The algorithm is finally implemented on a computer cluster, which enables massively parallel computing. Numerical experiments validate the accuracy of the proposed techniques and the up to 67X parallel speedup. Compared with other schemes, the unified dielectric pre-characterization scheme exhibits the highest accuracy while costing the least in terms of memory usage.

Keywords

Capacitance calculation; Floating random walk; Massively parallel computing; Multi-dielectric pre-characterization; Non-Manhattan geometry; Touchscreen.

1. INTRODUCTION

The flat panel display (FPD) has been a widespread and important human-computer interaction device in our daily life. In recent years, touch panel technology has been combined with FPD to largely enhance the interactivity and user experience of various customer electronics. This kind of touchscreen device includes both the display components, like those based on the thin-film transistor (TFT) active matrix [1, 2], and touch sensor components. This makes the internal structure of the touchscreen even more complicated. Most touchscreens utilize the capacitive touch sensor (see Fig. 1), because of its advantages in durability, reliability and capability [5]. To validate the functionality (like Multi-Touch, Force-Touch) and sensitivity of the touchscreen, calculating the relevant capacitances becomes an important and frequent task during the design of high-quality touchscreens.

The capacitance calculation problem in touchscreen design involves simulating the electrostatic field within the whole structure including touch sensor, surrounding FPD wires, and even the finger stylus. It calls for an accurate and efficient field-solver based solution. This problem is similar to the capacitance extraction problem in the design of vary large scale integrated

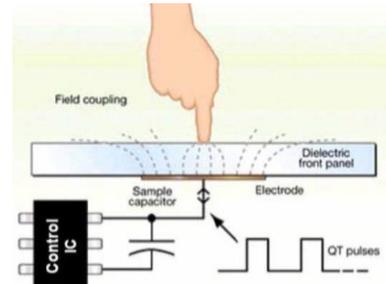


Fig. 1. The illustration of capacitive touch panel.

(VLSI) circuits. However, there are distinct differences between them, as listed in Table 1.

Table 1. The differences between capacitance extraction for VLSI circuit and capacitance calculation for touchscreen design.

	VLSI circuit capacitance extraction	Touchscreen capacitance calculation
Conductor geometry	Mostly Manhattan shape, with moderate aspect ratio	Generally non-Manhattan shape, with very large aspect ratio
Dielectric environment	On-chip dielectric insulators; relatively fixed dielectric profile	In-device dielectrics and out-device air; arbitrary dielectric configuration
Accuracy demand	Mainly on self-capacitance for accurate delay calculation	Need accurate coupling capacitances

A lot of field-solver techniques have been proposed for accurate capacitance extraction for VLSI design. They include the domain discretization method (finite difference method [6] and finite element method), the boundary element method (BEM) [7, 8], and the floating random walk (FRW) method [9–12]. The first two classes of methods involve volume or surface discretization and result in a system of linear equations. On the contrary, the FRW method is based on the Monte Carlo method, and has the advantages of more scalability for very large structures, tunable accuracy, better parallelism, and much smaller memory usage. Recent work on structures with large cylindrical through-silicon vias [13] also revealed that, the FRW method is more reliable on accuracy than the BEM capacitance solvers. However, the efficiency of FRW based techniques mainly depends on the assumption that the considered geometries are all of Manhattan shape, which is only true for VLSI circuits.

As for the capacitance calculation problem in the touchscreen design, the assumption of Manhattan shapes is not hold (see Table 1). Furthermore, the aspect ratio (lateral dimension over thickness) of metal in touchscreen structures can be larger than 1000, which causes difficulty for the discretization based methods like BEM. Secondly, the manufacturers of touchscreens are very diverse, which means a good capacitance solver for touchscreen verification should suit various configurations of dielectric

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

GLSVLSI '16, May 18 - 20, 2016, Boston, MA, USA

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-4274-2/16/05...\$15.00

DOI: <http://dx.doi.org/10.1145/2902961.2903011>

material. Therefore, the strategies that pre-characterize the FRW transition probabilities for certain multi-dielectric profiles (process technology) in [10, 14] are unfavorable. And, a larger range of dielectric permittivity should be considered due to the inclusion of air. Lastly, since the touch sensor acquires the touch location by detecting the difference of coupling capacitances, more accuracy in capacitance calculation is needed for simulating the touchscreen structures. This makes the FRW method with more reliable accuracy the best choice. The remaining problem is how to reduce its runtime while pursuing high accuracy.

There is some work on capacitance extraction for the liquid crystal display (LCD) based FPD design [3, 4]. However, they did not consider the touchscreen structure and related problems. The methods were based on pattern matching or an unknown field solver technique, and do not have sufficient accuracy or runtime efficiency for our purpose.

In this work, we aim to extend and apply the state-of-the-art FRW capacitance extraction techniques to the problem of touchscreen design. By proposing a technique to handle arbitrary conductor shape and a unified dielectric pre-characterization scheme, we are able to perform FRW based simulation for the touchscreen structures. The experiments on several touchscreen structures validate the accuracy and efficiency of the proposed techniques. To further reduce the runtime for accurate simulation, we implemented the algorithm on a large-scale computer cluster. The results of massively parallel computing reveals good scalability of the FRW based capacitance solver.

2. BACKGROUND

2.1 Floating Random Walk Algorithm for Capacitance Extraction in VLSI Design

The FRW method for calculating electrostatic capacitance originated from expressing the electric potential of a point \mathbf{r} as an integral of the potential on surface S enclosing \mathbf{r} [9], [10]:

$$\phi(\mathbf{r}) = \oint_S P(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)}, \quad (1)$$

where $P(\mathbf{r}, \mathbf{r}^{(1)})$ is called surface Green's function and can be regarded as a probability density function with non-negative values. Therefore, $\phi(\mathbf{r})$ is the statistical mean of $\phi(\mathbf{r}^{(1)})$, and can be calculated with a Monte Carlo (MC) procedure sampling S . The domain enclosed by S is called the transition domain, and usually \mathbf{r} is the center of the transition domain.

The problem of capacitance extraction is to calculate the capacitances related to a specified conductor (called *the master conductor*). For master conductor i , a Gaussian surface G_i is constructed to enclose it (see Fig. 2). According to Gauss's theorem, the charge of conductor i is given by

$$Q_i = \oint_{G_i} F(\mathbf{r}) g \int_{S^{(1)}} \omega(\mathbf{r}, \mathbf{r}^{(1)}) q(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r}, \quad (2)$$

where $F(\mathbf{r})$ is the dielectric permittivity at point \mathbf{r} , $q(\mathbf{r}, \mathbf{r}^{(1)})$ is the probability density function for sampling on $S^{(1)}$, the surface of a transition domain. g is a constant, which satisfies $\oint_{G_i} F(\mathbf{r}) g d\mathbf{r} = 1$. $q(\mathbf{r}, \mathbf{r}^{(1)})$ may be different from $P(\mathbf{r}, \mathbf{r}^{(1)})$, and

$\omega(\mathbf{r}, \mathbf{r}^{(1)})$ is the weight value [10]. Thus, Q_i can be estimated as the statistical mean of sampled values on G_i , which is also the mean of sampled potentials on $S^{(1)}$ multiplying the weight value. If the sampled potential is unknown, the construction of the transition domain and the spatial sampling procedure will repeat until a point with known potential is obtained (e.g. on a conductor surface). This forms a floating random walk (FRW) including a sequence of hops. Each hop is from the center of a transition domain to its boundary. With a number of such walks, the

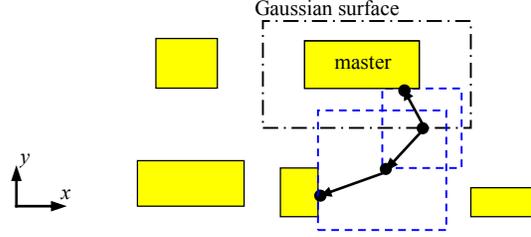


Fig. 2. Two examples of random walk in the FRW method for capacitance extraction (a 2-D top view).

statistical mean of the weight values for the walks terminating at conductor j approximates the capacitance C_{ij} between conductors i and j (if $j \neq i$), or the self-capacitance C_{ii} of master conductor i .

Although the surface Green's function for a spherical transition domain has simple analytical expression, the cubic transition domain is widely adopted because it fits well with the VLSI layout including mostly Manhattan shapes [9-12]. This yields larger probability for terminating a walk quickly. The sampling probability and weight value for a cubic domain can be pre-calculated and tabulated, so as to accelerate the sampling operation.

The runtime of the FRW method is proportional to the number of random walks. Several variance reduction techniques has been proposed to reduce the number of walks [10], [13], i.e. accelerate the convergence of MC procedure. A walk consists of a couple of hops. For a structure including many conductors, employing an efficient space management technique [12] is crucial for reducing the time for performing a hop.

2.2 Characteristics of the Touchscreen Structure and Its Capacitance Calculation

With Table 1, we have already summarize the differences between the capacitance calculation problems in VLSI design and touchscreen design. In Fig. 3, we show a typical dielectric profile and example conductor layouts (in top view) of the touchscreen structure. In Fig. 3(a) we see that the top dielectric layer is air, i.e. the relative permittivity $\epsilon_5=1$. And, the lateral dimension of a metal is usually much larger than its thickness. Fig. 3(b)-(c) show arbitrary-angle polygons and conductor with slits (holes). They include the geometries of a touch sensor and the wiring structures around it. It is obvious that the FRW method should be extended to handle these general non-Manhattan geometries.

The non-Manhattan conductor considered in this work can be regarded as a straight prism with an arbitrary polygon as the bottom. It has top and bottom faces parallel to the xoy axis plane.

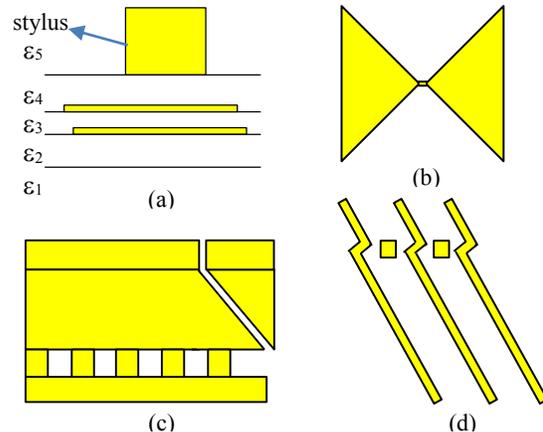


Fig. 3. (a): the cross-section view of a touchscreen structure. (b)-(d): some examples of top-view layout of the structure.

However, its projection on the xoy plane (i.e. the top view) is an arbitrary 2-D polygon, instead of an axis-aligned rectangle.

Now, the problem includes a number of 3-D conductor blocks. Each block is either a Manhattan cuboid or a convex straight prism with side faces perpendicular to the xoy plane. Note that a concave polygon can be easily decomposed into several convex ones. The master conductor may include a couple of connected conductor blocks. While running the FRW algorithm for a capacitance calculation, the cubic transition domain is only considered, since it nicely touches the surface of the conductor and leads to a faster termination of the random walk.

In the FRW method, the distance calculation between a point and a block or between two blocks is required. Because of the non-Manhattan conductors, the calculation becomes complicated, creating difficulty for the following parts of FRW method:

- 1) The generation of the Gaussian surface, which must enclose the master conductor and not intersect any conductor.
- 2) The construction of the transition cube for each hop, which requires finding the nearest conductor for a point.

Finally, because we are facing divergent process technology recipes for the touchscreen, it is desirable to have a unified dielectric pre-characterization scheme instead of pre-calculating the FRW transition tables for each process technology [10, 14]. Also note that the inclusion of air leads to a large range of dielectric permittivities. This prevents us from building a unified set of pre-characterization tables for VLSI capacitance extraction [11] and applying them to the touchscreen structure.

3. TECHNIQUES FOR CALCULATING THE TOUCHSCREEN CAPACITANCES

In this section, we first extend the FRW method for handling non-Manhattan conductor geometry. Then, a technique for building and using unified dielectric pre-characterization tables is proposed. Lastly, the FRW method is implemented on a large computer cluster to achieve considerable computational speedup.

3.1 Handling Arbitrary Conductor Geometry

To tackle the two difficulties caused by the non-Manhattan conductors, we first extend the aligned-box distance for Manhattan geometry to the non-Manhattan situation.

Definition 1: The 2-D aligned-box distance between a 2-D point P and a convex polygon A : $dist_a(P, A)$, is the half size of the axis-aligned square which is centered at P and touches A .

In Fig. 4(a), we show some typical positions of points around polygon, A . The Manhattan transition squares centered at the point and the corresponding aligned-box distance are illustrated. In Fig. 4(b), we show the basic idea for calculating $dist_a(P, A)$. We first find the visible edges of A in relation to point P . If the Manhattan square centered at P touches A 's edge, the edge must be a visible edge. For each edge A_iA_{i+1} , we calculate the cross product of $\overrightarrow{A_iP}$ and $\overrightarrow{A_iA_{i+1}}$. If the result is a positive value, the edge A_iA_{i+1} is visible, and we get the area of triangle PA_iA_{i+1} . As shown in Fig. 4(b), the area is useful for calculating the size of the

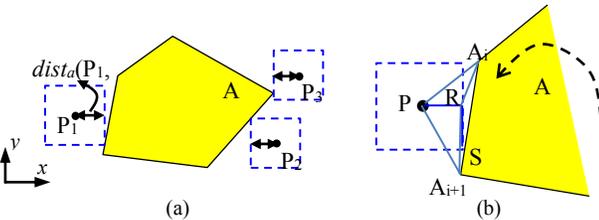


Fig. 4. (a) A convex polygon A and the aligned-box distances between it and nearby points. (b) The illustration for calculating the aligned-box distance.

transition square. Triangle PA_iA_{i+1} can be regarded as the combination of four triangles: PA_iR , PRA_{i+1} , RA_iS , and RSA_{i+1} , where S is the contact point and R is the midpoint of a transition square's edge. The four triangles all have half the size of the transition square as a bottom edge, while the corresponding heights form the x -distance and y -distance between points A_i and A_{i+1} . So, the cross product of $\overrightarrow{A_iP}$ and $\overrightarrow{A_iA_{i+1}}$ over the sum of the x -distance and y -distance equals the half edge length. $dist_a(P, A)$ is the maximum of such half edge length got from all visible edges, or corresponds to the situation where the Manhattan square touches A 's vertex [see P_3 's square in Fig. 4(a)]. The latter can be obtained with the Manhattan bounding box of A , using an existing technique. This analysis gives as Theorem 1.

Theorem 1: Suppose polygon A has vertices A_1, A_2, \dots, A_n , in the anti-clockwise order. A_i has coordinates (x_i, y_i) , $i=1, 2, \dots, n$. Suppose the point P has coordinates (x, y) . Then,

$$dist_a(P, A) = \max \left\{ \max_{1 \leq i \leq n} \frac{(x-x_i)(y_{i+1}-y_i) - (y-y_i)(x_{i+1}-x_i)}{|x_{i+1}-x_i| + |y_{i+1}-y_i|}, dist_a(P, B_A) \right\}, \quad (3)$$

where B_A is the Manhattan bounding box of polygon A . ■

The vertical distance between a point and a conductor block is defined as follows.

Definition 2: The vertical distance between a point $P(x, y, z)$ and a non-Manhattan conductor block A is:

$$dist_v(P, A) = \max \{ z - z_{\max}(A), z_{\min}(A) - z \}, \quad (4)$$

where $z_{\min}(A)$ and $z_{\max}(A)$ are the minimum and maximum z coordinates of A , respectively. So, the 3-D aligned-box distance is:

$$dist_a(P, A) = \max \{ dist_v(P, A), dist_a(P, P_A) \}, \quad (5)$$

where P_A stands for the xoy projection of A . The minimum distance between current position of random walk and its surrounding conductors is used to construct the Manhattan transition cube.

Based on the above definitions, we can further define the aligned-box distance between two convex polygons, and the 3-D distance between two non-Manhattan conductor blocks. They can be used to generate a valid Gaussian surface. We first calculate the minimum distance d_{\min} between a master conductor A and its neighboring conductors. Then, we place the Gaussian surface G_A about $d_{\min}/2$ distance away from A . It is guaranteed that such a Gaussian surface will not intersect any conductor. Based on the considered conductor geometry, the Gaussian surface surrounding the master conductor also forms a convex straight prism. In Fig. 5, the xoy projection of the Gaussian surface is shown. Each edge of A 's projection is inflated outward to obtain an edge where every point's aligned-box distance to A is $d_{\min}/2$. Then, the edges obtained by inflation are connected by adding edges, resulting in the xoy projection of G_A . If A 's projection has n edges, the number of edges of G_A 's projection is between n and $2n$. The whole Gaussian surface is finally obtained by raising the xoy projection along the z axis.

In our approach, we still use Manhattan (axis-aligned) transition cubes in the FRW algorithm. During the random walk procedure, the 2-D aligned-box distance can be used to determine

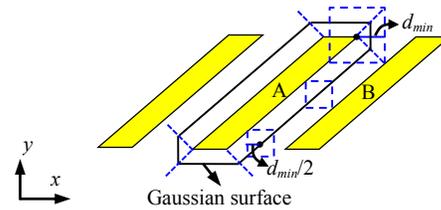


Fig. 5. The top view of a non-Manhattan conductor structure, and a valid generation of the Gaussian surface.

the Manhattan transition cube touching the side face of a non-Manhattan conductor. An idea is to allow the transition cube to rotate so as to help the transition cube touch the conductor surface better [13]. However, for touchscreen structures this is unnecessary because the walker seldom appears near the side walls of those thin-slice-alike conductors.

3.2 A Unified and Accurate Dielectric Pre-Characterization Method

In order to make the FRW method using transition cubes workable for multi-dielectric structures, the sampling probabilities and weight values for transition cubes containing multiple dielectrics must be pre-calculated. We call this the procedure of dielectric pre-characterization. In this subsection, we first outline the dielectric pre-characterization approaches used for extracting VLSI structures [11, 14]. We explain why they are not suitable for touchscreen structures. Then, an idea of building a unified set of dielectric pre-characterization tables is proposed. We also discuss how to balance the memory cost and the runtime benefit.

3.2.1 Two existing approaches

Based on a technique numerically characterizing the surface Green's function for two-dielectric-layer transition cubes, an approach was proposed in [10] for handling structures with multiple dielectric layers. For a given dielectric profile, the sampling probabilities and weight values for various two-dielectric-layer cubes are calculated and tabulated offline, and then used during the random walks. It is less efficient for actual VLSI process technology with 10 or more dielectric layers, because each FRW hop crosses one dielectric interface at most. An improvement was recently presented in [14], which pre-characterizes cubic transition domains with three or four dielectric layers so as to reduce the runtime of FRW. However, this greatly increased the number of dielectric configurations of the transition cube, and therefore the memory cost. A distinct drawback of this approach [10, 14] is that we must re-calculate the dielectric pre-characterization if the process technology changes.

Another pre-characterization approach is called the dielectric homogenization method [11]. Its main idea is assuming that any cubic transition domain with multiple dielectric layers can be approximated by a cube with four equal-thickness dielectric layers, no matter how many dielectric layers it actually contains. In Fig. 6, we show a structure with five dielectric layers to illustrate different transition cubes and pre-characterization strategies. When employing the dielectric homogenization method, one can use the blue transition cube, whereas one has to choose the red one if the approach of [10] is used. So, the dielectric homogenization method brings better runtime efficiency to FRW.

Now, we explain how to pre-characterize the cube with four equal-thickness dielectric layers. Suppose the four dielectrics have relative permittivities: $\bar{\epsilon}_1, \bar{\epsilon}_2, \bar{\epsilon}_3, \bar{\epsilon}_4$, their ratios determine the sampling probabilities and weight values for the cube. This means the cube with permittivities $(\bar{\epsilon}_1, \bar{\epsilon}_2, \bar{\epsilon}_3, \bar{\epsilon}_4)$ is equivalent to that with $(\bar{\epsilon}_1/\bar{\epsilon}_{\max}, \bar{\epsilon}_2/\bar{\epsilon}_{\max}, \bar{\epsilon}_3/\bar{\epsilon}_{\max}, \bar{\epsilon}_4/\bar{\epsilon}_{\max})$, where $\bar{\epsilon}_{\max} = \max\{\bar{\epsilon}_1, \bar{\epsilon}_2, \bar{\epsilon}_3, \bar{\epsilon}_4\}$. So, we need only consider situations where one permittivity is 1, and the other three have values in the interval (0, 1], for the pre-

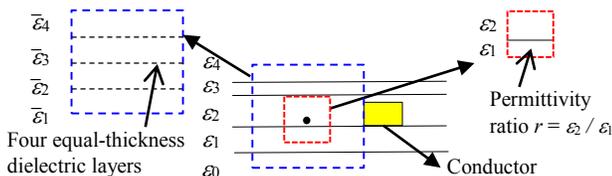


Fig. 6. An example for illustrating the transition cubes in the dielectric homogenization method and the proposed method.

characterization. Suppose the value of three permittivities is sampled with step size t , and the value range is $[s, 0.1]$ instead of $(0, 1]$. For the VLSI capacitance extraction, s equals 0.5 is a reasonable setting, which means the adjacent dielectrics always have permittivities with ratio no more than 2. So, the number of sampling dielectric configurations is about $4 \times [(1-s)/t+1]^3$, the constant 4 means that any one of $(\bar{\epsilon}_1, \bar{\epsilon}_2, \bar{\epsilon}_3, \bar{\epsilon}_4)$ can be 1. By symmetry, which means dielectric configuration $(1, a, b, c)$ is equivalent to $(c, b, a, 1)$, the number of sampling dielectric configurations can be reduced to $2 \times [(1-s)/t+1]^3 - [(1-s)/t+1]^2$. The reason for subtracting $[(1-s)/t+1]^2$ is that the dielectric configurations for the form $(1, 1, b, c)$ are counted twice.

We further count the data size for the dielectric homogenization method. According to [10], $4 \times 6N^2$ real numbers are needed to store the sampling probabilities and weight values (i.e. GFT and WVT tables) for a single configuration, where N is the segment number along transition cube edge. So, the dielectric homogenization method pre-calculates:

$$\text{Size}_{\text{DHM}} = 48N^2[(1-s)/t+1]^3 - 24N^2[(1-s)/t+1]^2 \quad (6)$$

real numbers. If $N = 32$, $s = 0.5$, $t = 0.05$, and single-precision numbers are used, the size of the pre-calculated data is about 238 MB. This is also the memory cost for running the FRW procedure.

Besides the runtime efficiency, another advantage of this approach is that the pre-characterization does not depend on the process technology. It is unified for arbitrary dielectric profiles. However, it has two drawbacks. The first one is on the error related to this four equal-thickness dielectric approximation. As pointed out in [14], this may induce significant error while handling the structures with more dielectric layers. The second one is the large memory cost for handling touchscreen structures. Since air, with $\epsilon=1.0$, is usually involved, the adjacent dielectrics may have permittivity ratio larger than 2. This means we set a smaller value for s , for example $s = 0.1$. With (6), we can calculate that the total memory cost for the pre-characterization increases to about 1.22 GB. This is a large number, and will limit the usage of the capacitance calculation.

3.2.2 The proposed method

To meet the requirements for simulating the touchscreen structures, we wish to combine the high accuracy and low memory cost of the approach in [10] and the unified aspect of the dielectric homogenization method. The idea is to compute pre-characterization for arbitrary two-dielectric cube configurations and then employ two-dielectric transition cubes during random walks. As shown in Fig. 6, we need to consider different choices of permittivities (ϵ_1, ϵ_2) and the position of dielectric interface. Due to the equivalence of different in-cube dielectric configurations and symmetry, we need only consider the situation where the two permittivities are $(1, r)$, $0 < r \leq 1$. We may use r , denoting the two-dielectric configuration $(1, r)$ when there is no ambiguity. Let s denote the smallest value of r , and N_{TDC} denote the number of sampling two-dielectric configurations. If we use equal-sized sampling of the r value, the number of samples is $n = (1-s)/t+1$, where t is the step size. Because there are $N-1$ positions for dielectric interface, we finally get:

$$N_{\text{TDC}} = n(N-1) = (N-1)[(1-s)/t+1]. \quad (7)$$

This approach pre-calculates:

$$\text{Size}_{\text{OUR}} = 24N^2 \cdot N_{\text{TDC}} = 24N^2(N-1)[(1-s)/t+1]. \quad (8)$$

real numbers. It corresponds to about 177 MB, if $N = 31$, $s = 0.1$, $t = 0.015$. Note that we have considered that the ratio of dielectric permittivity can increase to 10 in touchscreen structures. And, t is set to a smaller value to achieve higher accuracy.

This is a unified dielectric pre-characterization method, which means that the generated sampling probabilities and weight values

suit any dielectric configuration. If the actual permittivity ratio r of two adjacent dielectrics is between two sampled values: r_i and r_{i+1} , linear interpolation is employed. For example, let V_r denote the sampling probabilities of a two-dielectric configuration r . Then,

$$V_r = V_{r_i} \frac{r - r_i}{r_{i+1} - r_i} + V_{r_{i+1}} \frac{r_{i+1} - r}{r_{i+1} - r_i} . \quad (9)$$

This approach overcomes the shortage of the approaches in [10, 14], and avoids the large error caused by dielectric homogenization [11]. Although there are over 100 MB of pre-characterized data, not all of this is loaded to the memory while calculating a given structure. For example, if the structure includes dielectrics with permittivities (4, 3.2, 4, 1), corresponding to the permittivity ratios 0.8 and 0.25, we only need load data corresponding to dielectric configurations (1.0, 0.790), (1.0, 0.805), (1.0, 0.235) and (1.0, 0.250), which is only about 11 MB. This is an advantage over the dielectric homogenization method.

The only drawback of the proposed method is the computational speed, which is essentially the same as the FRW method in [10]. To reduce the runtime, an idea is to combine the dielectric homogenization method (with $s=0.5$) and our proposed idea to trade off running speed, memory and accuracy. We generate the data for both pre-characterization methods. While performing FRW, we use the homogenization method to allow making the large hop as long as possible. Otherwise, there is a permittivity ratio exceeding 2 within the four equal-thickness dielectrics, and we use the two-dielectric transition cube pre-characterized by the proposed method. This mixed approach improves memory/runtime tradeoff. However, it may still induce significant error because dielectric homogenization is employed. We will show this in the section on numerical experiments.

3.3 Massively Parallel Simulation on a Computer Cluster

The computational time of the FRW algorithm is inversely propositional to the square root of number of walks. This means its runtime increases substantially with greater accuracy. For calculating capacitances during the touchscreen design, highly accurate coupling capacitances are required. But, there is not an efficient way to accelerate the calculation of coupling capacitances [13]. So, a feasible way may be leveraging its potential for straightforward parallelization. To this aim, we implement the parallel FRW algorithm on a Cluster Environment with MPI. Fig. 7 shows the flowchart of this algorithm. Each process executes the random walk procedure independently and

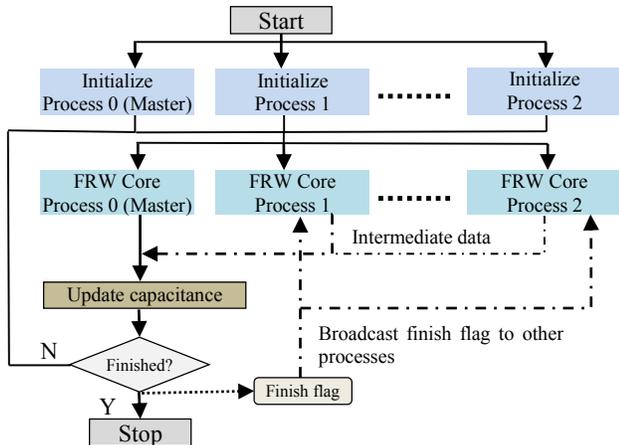


Fig. 7. Flowchart of the parallel FRW on a computer cluster.

all processes except the master process will send intermediate data to the master every m walks ($m=1000$). Then, the master process updates the capacitance value, we check the program termination criteria via total number of walks or estimated error. If it is satisfied, the master process broadcasts the finish flag to all other processes to terminate the computation. This is the classic master/worker parallel paradigm.

4. EXPERIMENTAL RESULTS

We have implemented the FRW method and the proposed techniques in C++. With the TechGFT program in [10], we have pre-calculated the sampling probabilities and weight values (i.e. GFTs and WVTs) for handling multi-dielectric structures. Three multi-dielectric touchscreen structures are tested. They include non-Manhattan conductor blocks. Some details are as follows.

Case 1: This case contains 1423 conductor blocks in two metal layers. The dielectric layers have relative permittivity of 4.0, 3.2, 4.0, 1.0. The metal heights in the two layers are 70nm and 220nm.

Case 2: This case is a small structure with 11 conductor blocks in two metal layers. The dielectric layers have relative permittivity of 4.0, 3.5, 7.0. The heights of the two metal layer are both 100nm. The top-view of one layer is similar to that shown in Fig. 3(b).

Case 3: This case contains 808 conductor blocks in four metal layers. The dielectric layers have relative permittivity of 3.9, 6.5, 3.5, 6.5, 4.2, 3.2, 4.0, 1.0. The heights of the four metal layers are 340nm, 220nm, 400nm and 70nm, respectively. The metal layout include the geometry patterns shown in Fig. 3(c)(d).

We first validate the accuracy of the proposed techniques with Raphael [6], which employs FDM with dense discretization. Then, we compare different approaches of dielectric pre-characterization. The experiments in Section 4.1 are carried out on a Linux server with Intel Xeon E5-2650 2.0GHz CPU with the termination criterion set to 0.5% $1-\sigma$ error on the self-capacitance. In Section 4.2, we carry out the parallel-computing experiment on a High-Performance Computing Cluster which consists 740 nodes with Intel Xeon X5670 2.93GHz CPU and InfiniBand QDR network.

4.1 Accuracy Validation and Comparisons

Because RWCap in [10] is not able to handle non-Manhattan shapes, we cannot compare our algorithms with it. Instead, following the strategy treating multiple dielectrics in [10], we obtain an algorithm called **FRW-2** for our non-Manhattan touchscreen structures. The FRW algorithms including the proposed unified dielectric pre-characterization method in Section 3.2 are denoted by **FRW-2unify** and **FRW-mixed**, corresponding to the strategy only using two-dielectric transition cubes and the strategy combining it and the dielectric homogenization approach, respectively. The results of FRW-2 with Raphael are listed in Table 2. Note that the results for Case 3 are not listed, because Raphael runs out of memory for it. We see that the results of FRW-2 are well correlated with those of Raphael even though they employ different methods and boundary assumptions. The results of coupling capacitances are also compared, which show similar correlation. This validate the accuracy of the techniques proposed in Section 3.1.

Table 2. The computational results of FRW-2 and Raphael (Capacitance in unit of $10^{-12}F$, Memory in unit of MB)

Case	C_{self} (Raphael)	C_{self} (FRW-2)	Mem. (FRW-2)	Error (%)
1	621.0	600.1	96.0	+3.4
2	78.4	78.7	56.9	+0.4

Now, regarding FRW-2 as the standard, we evaluate the approaches in FRW-2unify and FRW-mixed for handling multiple dielectrics. We run the both for 3000 times for each case and use the mean value as the capacitance value extracted. For Case 1 and

Case 2, the both methods produce negligible error, i.e. 0.03%. For Case 3, we plot the distributions of C_{tot} in Fig. 8. From it, we see that they both approximate the normal distribution, with Std about 0.5% of the means value as prescribed. The relative errors of FRW-2unify and FRW-mixed are 0.01% and -13.13%, respectively. This reveals a significant error caused by the mixed method, which includes the dielectric homogenization procedure.

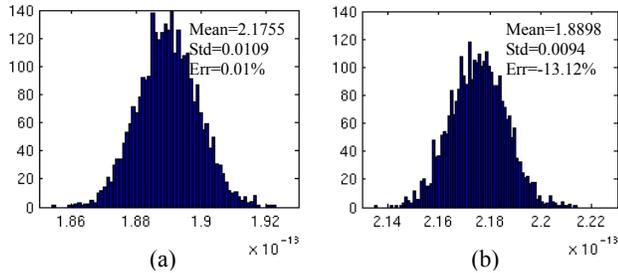


Fig. 8. Distribution of C_{self} calculated with (a) FRW-2unify and (b) FRW-mixed running 3000 times, respectively.

The runtime (for single run) and memory usage of the three FRW programs are listed in Table 3. From it, we see that FRW-2unify has almost the same runtime as FRW-2, which is shorter than FRW-mixed for Case 1 and 2. For Case 3, FRW-mixed is the fastest, because the dielectric homogenization takes effect for the case with more dielectric layers. However, FRW-mixed consumes over 10X more memory, and has significant error as shown in Fig. 8. The pre-calculated data sizes of FRW-2unify and FRW-mixed are about 177MB and 415MB, respectively. Obviously, FRW-2unify is superior to FRW-mixed, and more adaptable to the technology change in touchscreen design problems than FRW-2.

Table 3. The CPU time and memory usage of FRW-2, FRW-2unify and FRW-mixed (Memory in unit of MB).

Case	FRW-2		FRW-2unify		FRW-mixed	
	Time (s)	Mem.	Time (s)	Mem.	Time (s)	Mem.
1	2.3	9.6	2.4	12.4	2.8	250.9
2	538.9	5.7	530.2	11.2	629.0	249.6
3	221.5	21.0	227.1	34.7	40.3	273.2

4.2 Performance of Parallel Computing

To obtain more accurate coupling capacitances, the FRW-2unify is run with 0.1% $1-\sigma$ error criterion on the self-capacitance. The parallel speedup for the three cases is shown in Fig. 9, with different number of processes. Because of the divergence among random walks, and the communication and synchronization costs among different computing nodes, the speedup cannot reach the ideal situation. With 120 processes, the runtime for simulating Case 3 is reduced from 7452 seconds in serial computing to 111 seconds. This means a $67\times$ speedup.

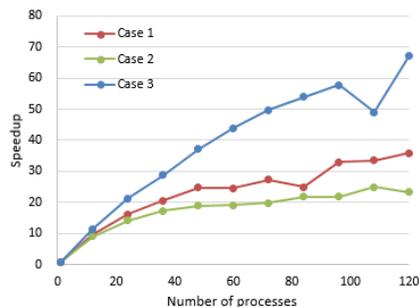


Fig. 9. The parallel speedup vs. the number of processes.

5. CONCLUSIONS

A floating random walk based solver (i.e. FRW-2unify) is presented for the capacitance-calculating problem in touchscreen design. It includes the technique handling non-Manhattan geometries and a unified and accurate dielectric pre-characterization scheme. Experiments validate their effectiveness and advantages, and show the speedup brought by parallel computing on a large computer cluster as well.

6. ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China under Grant No. 61422402.

7. REFERENCES

- [1] Park H., Kim S., Kim S., Jo Y., Kim S. and McCartney R. I. 2008. Electrical models of TFT-LCD panels for circuit simulations. *Journal of the Society for Information Display*, 16 (2008), 509-515.
- [2] Son Y. S. and Cho G. H. 2008. Design considerations of channel buffer amplifiers for low-power area-efficient column drivers in active-matrix LCDs. *IEEE Trans. Consumer Electronics*, 54 (Feb. 2008), 648-656.
- [3] Takagi M., Yamaguchi K., Chida H. et. al. 2012. Layout and reticle verification for FPD. In *Proc. SPIE 84410M* (2012), 1-4.
- [4] Uchida Y., Tani S., Hashimoto M., et. al. 2005. Interconnect capacitance extraction for system LCD circuits. In *Proc. the 15th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, (2005). 160-164.
- [5] Egan J. 2014. Resistive vs. Capacitive Touchscreens. Available: <http://blog.junipersys.com/resistive-vs-capacitive-touchscreens/>
- [6] Synopsys Inc.. Raphael: 2D, 3D resistance, capacitance and inductance extraction tool. Available: <http://www.synopsys.com/Tools/TCAD/InterconnectSimulation/Pages/Raphael.aspx>
- [7] Nabors K. and White J. 1991. FastCap: A multipole accelerated 3-D capacitance extraction program. *IEEE Trans. Computer-Aided Design*, 10 (Nov. 1991), 1447-1459.
- [8] Yu W. and Wang Z. 2004. Enhanced QMM-BEM solver for three-dimensional multiple-dielectric capacitance extraction within the finite domain. *IEEE Trans. Microwave Theory Tech.*, 52 (Feb. 2004), 560-566.
- [9] Le Coz Y. and Iverson R. B. 1992. A stochastic algorithm for high speed capacitance extraction in integrated circuits. *Solid-State Electronics*, 35 (Jul. 1992), 1005-1012.
- [10] Yu W., Zhuang H., Zhang C., Hu G. and Liu Z. 2013. RWCAP: A floating random walk solver for 3-D capacitance extraction of very-large-scale integration interconnects. *IEEE Trans. Computer-Aided Design*, 32 (Mar. 2013), 353-366. Available: <http://learn.tsinghua.edu.cn:8080/2003990088/rwcap.htm>
- [11] Rollins G. 2010. Rapid3D 20X performance improvement. Available: <http://www.synopsys.com/Community/UniversityProgram/Pages/Presentations.aspx>
- [12] Zhang C. and Yu W. 2013. Efficient space management techniques for largescale interconnect capacitance extraction with floating random walks. *IEEE Trans. Computer-Aided Design*, 32 (Oct. 2013), 1633-1637.
- [13] Zhang C., Yu W., Wang Q. and Shi Y. 2015. Fast random walk based capacitance extraction for the 3-D IC structures with cylindrical inter-tier-vias. *IEEE Trans. Computer-Aided Design*, 34 (Dec. 2015), 1977-1990.
- [14] Zhang B., Yu W. and Zhang C. 2015. Improved pre-characterization method for the random walk based capacitance extraction of multi-dielectric VLSI interconnects. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, DOI: 10.1002/jnm.20