Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

NONLINEAR OPTIMIZATION TO GENERATE NON-OVERLAPPING RANDOM DOT PATTERNS

Takashi Imamichi
Hidetoshi Numata
Hideyuki Mizuta
Tsuyoshi Idé

IBM Research – Tokyo
1623-14 Shimo-tsuruma, Yamato
Kanagawa 242-8502, JAPAN

ABSTRACT

We have devised a method to generate non-overlapping random dot patterns for light guides and diffuser films in liquid crystal displays (LCDs). Molecular-dynamics-based algorithms are being for this purpose and have been proven to generate high quality dot patterns. The key technical challenge is how to remove inter-dot overlap that leads to visible roughness in the luminance distribution. In this paper, we describe a new overlap removal method that penalizes the overlap of dots and minimizes the sum of the penalties by using a nonlinear optimization technique. Through computational experiments with real world data, we show that our optimization-based method runs faster than an existing simulation-based method and generates dot patterns with comparable quality.

1 INTRODUCTION

Simulation-based techniques are widely recognized as powerful tools for manufacturing. Examples include the finite element method (Kamnerdtong et al. 2008; Liu 2000), discrete event simulation (Sharda and Bury 2010, Alexander 2006), and scenario analysis in production systems (Choi et al. 2002; Pickardt et al. 2010).

This paper addresses a particular application in liquid-crystal display (LCD) manufacturing. Figure 1 shows a conventional structure for an edge-lit backlight unit for an LCD, where a number of optical components such as CCFL (cold cathode fluorescent lamp), diffuser film, and prism sheet are used to obtain uniform luminance across the display. As indicated in the figure, light scatterers are placed on the surface of the prism sheet and diffuser film. By optimizing the distribution of the scatterer pattern (AKA the dot pattern), uniform luminance can be obtained. Our work seeks to develop a faster method for generating high-quality dot patterns.

In the LCD manufacturing process, the design of light guide and diffuser film is one of the most difficult steps. In general, a dot pattern on the light guide and diffuser film must satisfy a number of requirements: The dot patterns should be (1) properly irregular and (2) sufficiently uniform. In addition, (3) the method should be capable of providing arbitrary density gradation. If the first requirement is not satisfied, the resulting dot pattern may cause moiré patterns in the luminance distribution (Figure 2). Unless the second requirement is satisfied, the resulting dot pattern may cause visible roughness (Figure 3).

For this task, a molecular-dynamics-simulation-based approach is known to produce good optical quality, and has been used in high-end LCDs. However, it is computationally expensive and has quality problems at the edges. Idé et al. (2003) proposed a two-phase algorithm to generate the dot patterns. In the first phase they generate a dot pattern by using low discrepancy sequences (LDS) (Tezuka 1993) instead of pseudo-random numbers to introduce irregularity into the dot distribution with less inter-dot overlap. In

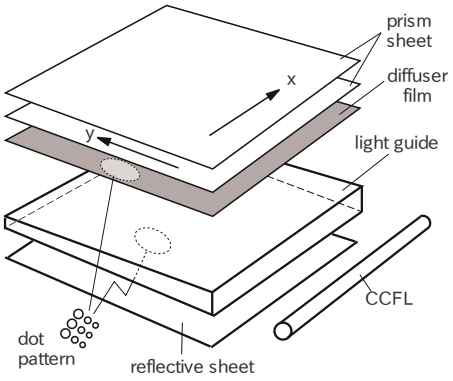

Figure 1: Conventional structure of edge-lit backlight unit. There are dot patterns on the light guide and diffuser film.

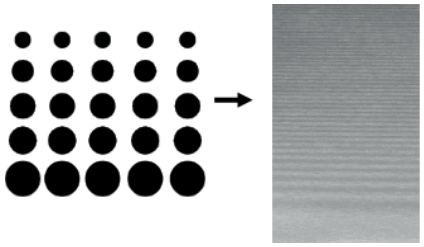

Figure 2: Regular dot distribution causes moiré patterns.

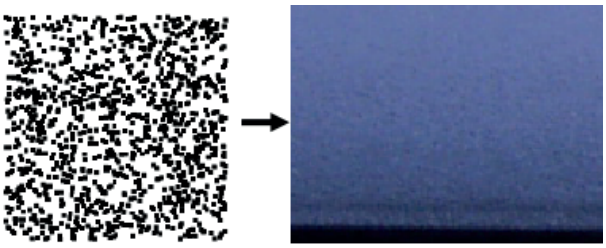

Figure 3: Insufficient uniform distribution causes visible roughness.

the second phase, the algorithm modifies the dot distribution by using a molecular dynamics redistribution method to remove inter-dot overlaps in the dot pattern. Chang and Lee (2007) and Chang, Fang, and Ju (2009) also proposed similar molecular-dynamics-simulation-based redistribution algorithms for the dots. Meanwhile, Chang and Fang (2007) proposed an approach to optimize the dot radii iteratively, where the dot positions are fixed, i.e., the dots are on predefined grid points.

We propose a new approach to the random dot pattern generation task. We first generate an initial dot pattern from a dot-density-and-simulation-based method. Then we replace each dot with a circle at an appropriate radius computed from the dot density and eliminate the overlap among the circles by applying a nonlinear optimization technique. Finally we place the dots at the positions of the circles. Using real world data, we show that our approach drastically reduces the computational costs while producing better quality scatterer patterns at the edges.

Our approach is an innovative paradigm because (1) the optimization formulation gives a macroscopic insight into the microscopic molecular dynamics simulation and (2) our approach greatly accelerates the design of the dot distributions. We demonstrate the optimization is useful to obtain a steady state in a simulation by formulating an appropriate optimization problem from the microscopic interactions among objects, where the objects correspond to the dots in this task. A typical approach in simulations is to model the global characteristics by defining microscopic interactions among the objects and our approach is similar. The difference is that we define the global objective function from the microscopic interactions among the objects and use an optimization technique to obtain a (locally) optimal solution. Simulation-based methods are suitable for tracing the changes of the system for entire periods of time, but they may take a long time to converge to a steady state. Meanwhile optimization-based methods are designed to quickly converge to a (locally) optimal solution. Note that the optimal solution in optimization corresponds to the steady state in a simulation.

Optimization-based techniques are used in manufacturing. For example, Beham et al. (2009) optimized a facility layout scenario by using metaheuristics. Villarreal et al. (2008) proposed a combination of simulation and optimization for the control of injection molding. They obtained the values of the objective function by simulation and optimize the values iteratively. To the best of our knowledge, there is no optimization-based approach for dot-position design in LCDs.

The organization of this paper is as follows. In Section 2 we describe an existing simulation-based approach. In Section 3 we introduce our new optimization-based approach for random dot pattern generation with no inter-dot overlaps. We show the experimental results in Section 4 and offer concluding remarks in Section 5.

2 SIMULATION-BASED APPROACH

In this section we briefly illustrate an existing simulation-based approach by Idé et al. (2003). Given a domain and dot density in the domain, they consider the task to generate a dot pattern whose distribution is random and uniform. Note that the domain corresponds to a light guide or a diffuser film.

Idé et al. (2003) proposed an algorithm which consists of two phases.

1. Generate an initial dot pattern from the given density. Idé et al. (2003) utilized low discrepancy sequences so as to make a random dot distribution which is relatively uniform compared with the dot distribution by using pseudo-random numbers.
2. Modify the initial dot pattern to make the dot distribution uniform. They applied the molecular dynamics simulation by defining interaction forces between dots.

They demonstrated that this method effectively improved luminance uniformity and outperformed a naive method by pseudo-random numbers.

2.1 Initial Dot Pattern Generation

We introduce a dot pattern generation by low discrepancy sequences in this subsection. We consider a situation where the domain is divided into m equal size rectangles $\{R_1, \ldots, R_m\}$, whose size is $L_x \times L_y$, and a density ρ_i is assigned to R_i $(i = 1, \ldots, m)$. Note that the density ρ_i is defined by the proportion of the total area of dots in R_i to the area of R_i. The total number of dots n is computed from the density ρ_i and the area of R_i. We introduce the probability P_i of hitting R_i in a "dart-throwing game" with dots by $P_i = \rho_i / \sum_{j=1}^{m} \rho_j$ $(i = 1, \ldots, m)$.

An initial dot pattern is generated by repeating the following tasks n times.

1. Generate a three-dimensional LDS defined within $[0, 1]^3$ and take a point (U_0, U_1, U_2).
2. Choose k from $1, \ldots, m$ which satisfies the following condition:

$$\sum_{j=1}^{k} P_j \leq U_0 < \sum_{j=1}^{k+1} P_j.$$

3. Add a dot to R_k with a position $(\eta_k + L_x U_1, \xi_k + L_y U_2)$, where (η_k, ξ_k) is the bottom left coordinate of R_k.

Idé et al. (2003) showed that the resulting dot pattern by LDS had less visible roughness than that by pseudo-random numbers. We incorporate the dot pattern generation by LDS into our approach too.

2.2 Molecular-Dynamics-Simulation-Based Adjustment of Dot Patterns

We briefly describe a molecular-dynamics-simulation-based method to adjust dot patterns by Idé et al. (2003) in this subsection. This method is the state-of-the-art and also utilized in other approaches, e.g., (Chang and Lee 2007; Chang, Fang, and Ju 2009).

They defined an interaction force $\boldsymbol{f}_{ij}$ between dots i and j by

$$\boldsymbol{f}_{ij} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} \times \begin{cases} 1 & b_{ij} < D, \\ \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\| - b_{ij}}{L}\right) & b_{ij} \geq D, \end{cases} \tag{1}$$

where D and b_{ij} are parameters.

They computed the positions of the dots from the following equation of motion

$$m\frac{d^2\boldsymbol{x}_i}{dt^2} + c\frac{d\boldsymbol{x}_i}{dt} = \sum_{j=1}^{n} \boldsymbol{f}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

They approximated this equation as a difference equation

$$\boldsymbol{x}_i(t + \Delta t) = \boldsymbol{x}_i(t) + \frac{1}{c}\Delta t \sum_{j=1}^{n} \boldsymbol{f}_{ij}(t), \tag{2}$$

and computed $\boldsymbol{x}_i(t)$ with sufficient small Δt.

They showed that their method improved the uniformity of an initial dot pattern. However, we found two drawbacks of this approach.

First, this method runs slow if there are some regions where the density is very small. When they solve the equation (2) of motion, they calculate the sum of the interaction forces (1) between dots. If D is small, they can omit some computation because the forces are 1 for many cases. However, since the parameter D and the squared root of the density are inversely proportional, the regions with low densities increase D.

Second, there are some dots moving to regions with zero density in the resulting dot pattern. See the right figure of Figure 4.

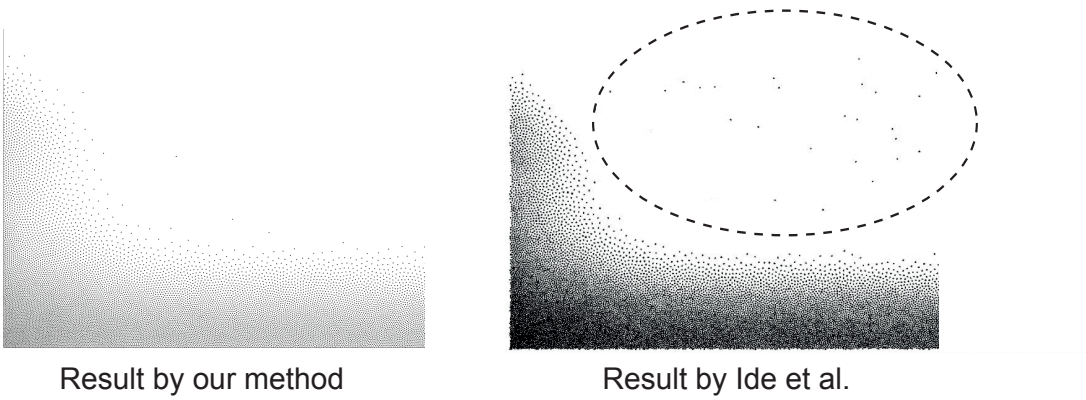

Figure 4: Comparison of the results by our method and Idé et al. (2003) with "Tsuno" instance (Some dots move to empty space depicted by a dashed ellipse where the dot density is zero).

3 OPTIMIZATION-BASED APPROACH

In this section we introduce a new optimization-based approach to the dot pattern generation. The outline of our approach is similar to the simulation-based approach by Idé et al. (2003) described in Section 2. We generate an initial dot pattern by LDS and then adjust the dot positions to remove inter-dot overlaps. See Algorithm 1 and Figure 5 for the outline of our approach. The difference between the simulation-based approach and the optimization-based approach is that we adopt an optimization technique, which is called the multi-sphere scheme by Imamichi and Nagamochi (2007), in order to eliminate inter-dot overlaps instead of the molecular dynamics simulation. The multi-sphere scheme is an approach to design efficient algorithms that compute compact layouts of given objects for the packing problem in 2D and 3D space. In the multi-sphere scheme, we first approximate each object by a set of spheres, and then search for positions of all the spheres that minimize an appropriate penalty function. For this, they formulated the problem of finding a layout of sets of spheres as an unconstrained optimization problem. This optimization problem can provide us an efficient procedure for modifying a given layout into a new layout with no overlap, where such a layout is obtained as a locally optimal solution to the optimization problem. Approximating objects by spheres makes it easy to check collisions of objects and handle rotations of objects by arbitrary angles. Note that the multi-sphere scheme is very general and can handle both rigid and deformable objects and we use the rigid case.

Algorithm 1 : `OptDot(`ρ`)`

Input: Domain C, dot density ρ, dot radius r_0.
Output: Dot pattern $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_n^*\}$.

1: Generate an initial dot pattern $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ by LDS with ρ (Section 2.1).
(Note that the number n of dots are determined here. We call $\boldsymbol{x}_i$ the position of dot i.)
2: Compute the circle radius r_i for each dot i (Section 3.3).
(Note that a circle corresponds to dot i is denoted by S_i, whose center is positioned at $\boldsymbol{x}_i$ and radius is r_i.)
3: Remove overlaps of the set of circles $\{S_1, \ldots, S_n\}$ (Section 3.1).
4: Replace the circles in the resulting layout by dots.

We add two new ideas to obtain ideal dot patterns. First idea is to approximate each dot by a single circle at an appropriate radius that is determined by the density. If a dot pair is close to each other, the corresponding circles collide each other to be separated. Second idea is to assign a square region, which we call a container, to each circle that restricts the motion of the circle. We let the edge length of a container

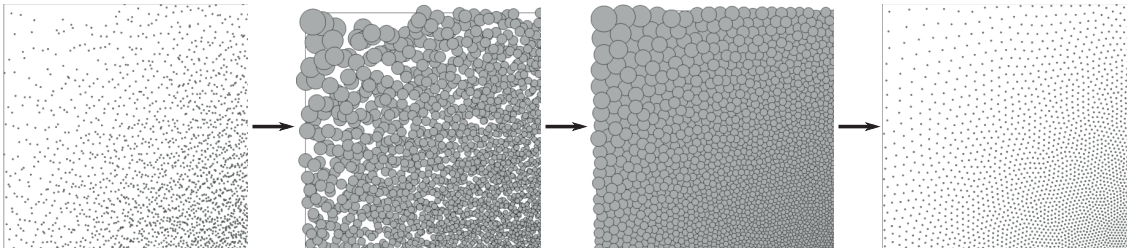

Figure 5: The outline of the generation of non-overlapping random distributed dot pattern. The figures are the magnified pictures of the results of instance LED.

four times longer than the radius of the circle (Figure 8). Because it keeps the initial dot layout to some extent, the resulting dot layout also keeps randomness which prevents moiré patterns.

3.1 Formulation

In this section we formulate the circle overlap removal problem to make an initial dot pattern uniform. We consider following three types of penalties:

1. Penalty of overlap between circles (see Figure 6).
2. Penalty of protrusion of circles from the domain C (see Figure 7).
3. Penalty of protrusion of each circle from the container assigned to the circle (see Figure 8).

The formulation has a global objective function which is the sum of the three types of penalties with no constraint. The decision variables are the positions of the circles, which correspond to the dot positions. A solution whose objective function value equals to zero gives a layout of circles where no two circles intersect and all circles do not protrude neither from the domain nor from the containers.

Let $\overline{A}$ be the complement of a set A, ∂A be the boundary of A, and $\mathrm{int}(A) = A \setminus \partial A$ be the interior of A. After translating circle S by a translation vector $\boldsymbol{x} \in \mathbb{R}^2$, the resulting layout of circle is denoted by $S \oplus \boldsymbol{x} = \{\boldsymbol{t} + \boldsymbol{x} \mid \boldsymbol{t} \in S\}$. The *penetration depth* (Agarwal et al. 2000) of two circles S and T is defined by $\delta(S, T) = \min\{\|\boldsymbol{x}\| \mid \mathrm{int}(S) \cap (T \oplus \boldsymbol{x}) = \emptyset, \boldsymbol{x} \in \mathbb{R}^2\}$, where $\|\cdot\|$ denotes the Euclidean norm. For circles S_i and S_j, the penetration depth of them is

$$\delta(S_i, S_j) = \max\{r_i + r_j - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|, 0\}.$$

For a circle S_i and a complement $\overline{R}$ of a rectangle, the penetration depth $\delta(S_i, \overline{R})$ is calculated easily too.

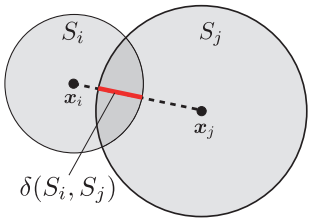

Figure 6: Penetration depth of two overlapping circles.

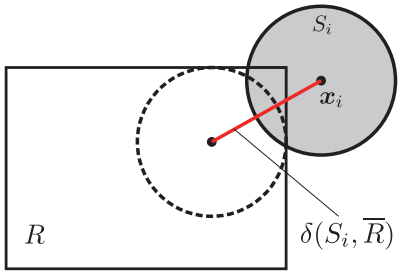

Figure 7: Penetration depth of a circle and the outside of a rectangle.

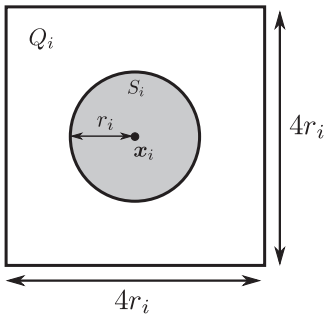

Figure 8: A container of a circle.

We then formally define the circle overlap removal problem by

$$\begin{aligned} \text{minimize} \quad & \sum_{1\le i<j\le n} \delta(S_i \oplus \boldsymbol{x}_i, S_j \oplus \boldsymbol{x}_j)^2 + \sum_{1\le i\le n} \delta(S_i \oplus \boldsymbol{x}_i, \overline{R})^2 + \sum_{1\le i\le n} \delta(S_i \oplus \boldsymbol{x}_i, \overline{Q_i})^2, \qquad (3) \\ \text{subject to} \quad & \boldsymbol{x}_i \in \mathbb{R}^2,\ 1 \le i \le n \end{aligned}$$

as an unconstrained nonlinear optimization problem. Note that all penalties contained in the objective function are differentiable.

It is required to enumerate all colliding pairs of circles to compute the objective function (3) and its gradient. We apply the fast algorithm to detect all colliding pairs by Imamichi and Nagamochi (2008). This algorithm splits the domain into thin slabs and applies a plane sweep method to each slab. It requires $O(n \log n + K)$ time, where K is the number of colliding pairs of circles. This time complexity is proven to be optimal by Imamichi and Nagamochi (2008). In Figure 9 we split the domain into three slabs. The circles checked by the plane sweep method applied to the middle slab are depicted in gray color.

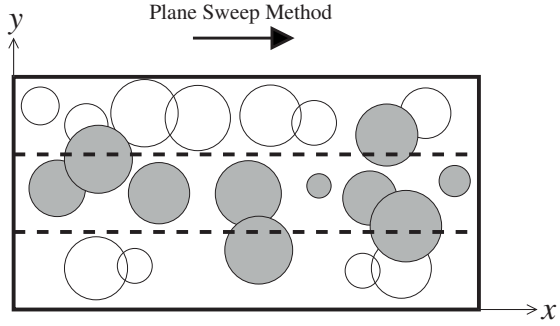

Figure 9: Enumerate all colliding pairs of circles by applying the plane sweep method to each slab.

3.2 Limited Memory BFGS Method

To solve the circle overlap removal problem (3) we apply the *Limited Memory BFGS (L-BFGS) method* by Liu and Nocedal (1989). L-BFGS method is a variant of the quasi-Newton method for large scale unconstrained nonlinear optimization problems, which is known to converge quickly to a locally optimal solution in practice.

L-BFGS method optimizes a b-dimensional minimization problem

$$\text{minimize } f(\boldsymbol{z}) \quad \text{subject to } \boldsymbol{z} \in \mathbb{R}^b$$

by applying a line search iteratively into the direction $\boldsymbol{d}_k = -H_k \boldsymbol{g}_k$, where H_k is an approximate inverse Hessian and $\boldsymbol{g}_k$ is the gradient in the kth iteration. Let $\boldsymbol{s}_k = \boldsymbol{z}_{k+1} - \boldsymbol{z}_k$ and $\boldsymbol{y}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$. L-BFGS method computes H_k from m pairs $(\boldsymbol{s}_i, \boldsymbol{y}_i)$ $(i = k-m, \ldots, k-1)$ by

$$\begin{aligned} H_k = {} & (V_{k-1}^{\mathsf{T}} \ldots V_{k-m}^{\mathsf{T}}) \frac{\boldsymbol{s}_k{}^{\mathsf{T}} \boldsymbol{y}_k}{\|\boldsymbol{y}_k\|^2} I (V_{k-m} \ldots V_{k-1}) \\ & + \rho_{k-m} (V_{k-1}^{\mathsf{T}} \ldots V_{k-m+1}^{\mathsf{T}}) \boldsymbol{s}_{k-m} \boldsymbol{s}_{k-m}{}^{\mathsf{T}} (V_{k-m+1} \ldots V_{k-1}) \\ & + \rho_{k-m+1} (V_{k-1}^{\mathsf{T}} \ldots V_{k-m+2}^{\mathsf{T}}) \boldsymbol{s}_{k-m+1} \boldsymbol{s}_{k-m+1}{}^{\mathsf{T}} (V_{k-m+2} \ldots V_{k-1}) \\ & + \ldots \\ & + \rho_{k-1} \boldsymbol{s}_{k-1} \boldsymbol{s}_{k-1}{}^{\mathsf{T}}, \end{aligned}$$

where

$$\rho_k = \frac{1}{\boldsymbol{s}_k \boldsymbol{y}_k{}^{\mathsf{T}}}, \quad V_k = I - \rho \boldsymbol{y}_k \boldsymbol{s}_k{}^{\mathsf{T}}.$$

Since L-BFGS method requires $O(n)$ space under the assumption that m is constant, it is suitable for large scale problems. We adopt L-BFGS method because we deal with the cases with more than 100,000 dots. Note that the parameter m is recommended to be in the range $3 \leq m \leq 7$ by Liu and Nocedal (1989), where we let $m = 6$ in our experiments.

3.3 Circle Radius

We explain how to determine radii of circles that replace the dots. By giving appropriate radii to the circles, we realize the uniformity of the dot patterns.

Let r_0 be the dot radius and ρ be a dot density by area in a specific region X_i. Note that ρ is equal to the proportion of total area of dots to the area of whole space. Let r be the radius of a circle that replace a dot in X_i.

We design r from the density ρ in the densest packing of circles in the plane, i.e., the hexagonal lattice. The radius r' of the circles placed in the hexagonal lattice is equal to

$$r' = r_0 \sqrt{\frac{\pi}{2\sqrt{3}\rho}}. \tag{4}$$

Through preliminary experiments, we found that resulting dot patterns computed using the radius r' had visible roughness to some extent. This is because the radius computed from the layout of circles where the circles touch each other. Hence, we adopt a slightly bigger radius r instead of r' defined by

$$r = r_0 \sqrt{\frac{\pi}{\sqrt{3}\rho}}.$$

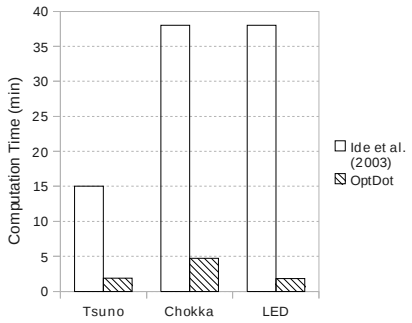

Figure 10: Computation time (min).

4 EXPERIMENTS

We conducted experiments using real world data and compare the results with the results by Idé et al. (2003). We implement our algorithm by C++. We set the maximum number of iterations to 200 for both algorithms. We run both algorithms on the same PC with Core 2 Duo T9300 CPU (2.5GHz).

We use three instances of real world data, which is called Tsuno, Chokka and LED. Table 1 shows the number of dots of the instances. The dot radius r_0 is equal to 46μm for all instances.

Table 1: Number of dots in the initial dot patterns.

Instance	Number of dots	Number of dot pairs
Tsuno	111211	6183887655
Chokka	287080	41207319660
LED	117088	6854741328

Figure 10 and Table 2 show the computation time for each instance. Our algorithm runs much faster for all instances.

Table 2: Computation time (min).

Instance	Idé et al. (2003)	`OptDot`
Tsuno	15	1.9
Chokka	38	4.7
LED	38	1.8

We compare the quality of the resulting dot distribution by the number of overlapping or very close dot pairs because such dot pairs cause visible roughness. Table 3 and Table 4 show the number of dot pairs which collide each other (the distance between dot centers is less than 46μm) and the number of dot pairs which do not intersect each other but are very close (the distance between dot centers is less than 60μm), respectively. For all instances, the results by our algorithm contains less dot pairs which are overlapping or very close.

We show the resulting dot patterns by our algorithm for all instances. Figure 5, Figure 11, Figure 12 is the results of LED, Chokka, Tsuno instances, respectively. We can see that our algorithm succeeded to generate uniform dot distributions. Also we show a comparison at an edge in Figure 4. We observed that the result by ours algorithm has fewer number of dots moving to the empty space, thanks to the containers assigned to the circles.

5 CONCLUSION

We have proposed a new optimization-based approach to random dot pattern generation. In our method, we first generate an initial random dot pattern using a randomized LDS, and then remove unwanted inter-dot

Table 3: Number of dot pairs that collide each other (the distance between them is less than 46μm).

Instance	Initial layout	Idé et al. (2003)	`OptDot`
Tsuno	24339	1	0
Chokka	36827	0	0
LED	57949	1315	0

Table 4: Number of dot pairs that are too close to each other (the distance between them is less than 60μm).

Instance	Initial layout	Idé et al. (2003)	`OptDot`
Tsuno	45523	66	16
Chokka	73055	2	0
LED	104530	37844	32644

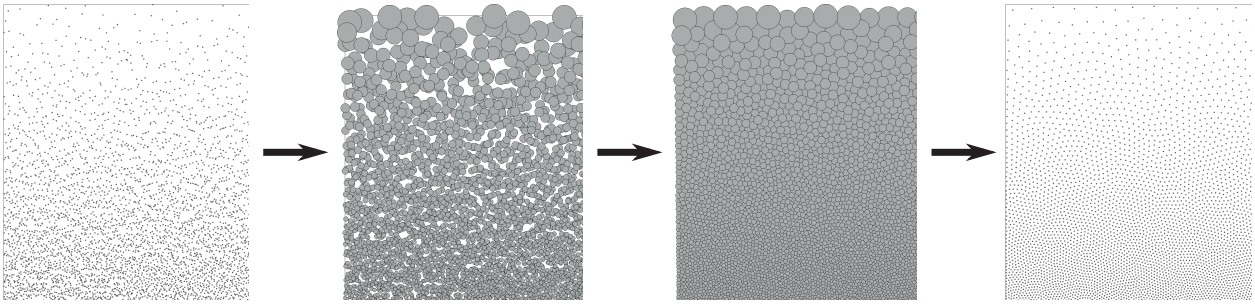

Figure 11: Magnified pictures of the results of Chokka instance.

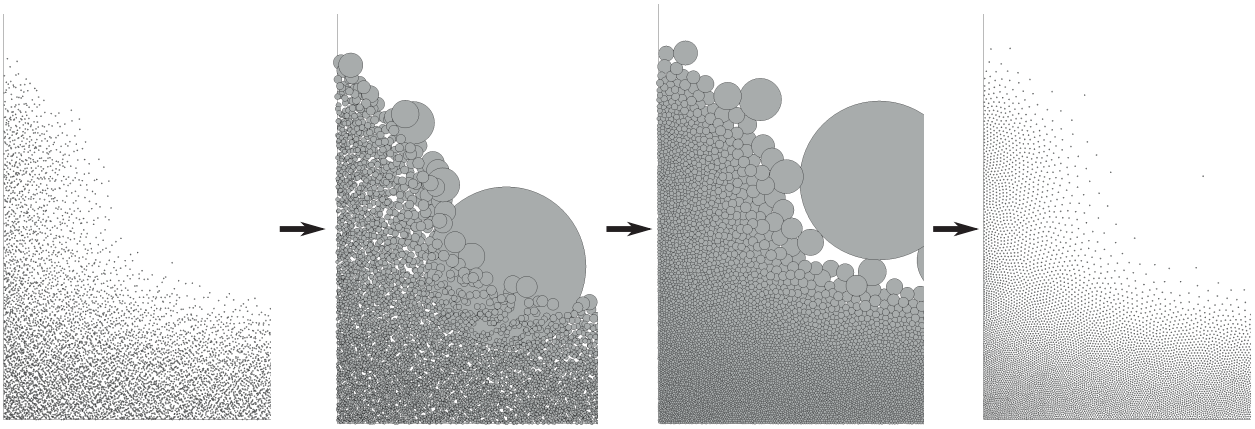

Figure 12: Magnified pictures of the results of Tsuno instance.

overlaps in the initial pattern. The key feature of our method is the use of the multi-sphere scheme for the previously proposed simulation-based approach. By applying the L-BFGS method, we showed that the proposed method is orders of magnitude faster than the state-of-the-art method, and improves the quality of the dot patterns especially near the edges.

For future work, it would be interesting to study the applicability of our approach to non-circular scatterers such as rectangles. One possible approach would be to approximate each scatterer (or dot) by a set of circles instead of a single circle, as studied by Imamichi and Nagamochi (2007). Also, from a practical perspective, it would be interesting to evaluate the physical characteristics of the optical components with the dot patterns generated by our approach.

REFERENCES

Agarwal, P. K., L. J. Guibas, S. Har-Peled, A. Rabinovitch, and M. Sharir. 2000. "Penetration depth of two convex polytopes in 3D". *Nordic Journal of Computing* 7 (3): 227–240.

Alexander, C. W. 2006, December. "Discrete event simulation for batch processing". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1929–1934. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Beham, A., M. Kofler, S. Wagner, and M. Affenzeller. 2009, December. "Coupling Simulation with Heuristiclab to Solve Facility Layout Problems". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 2205–2217. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chang, J.-G., and Y.-B. Fang. 2007. "Dot-pattern design of a light guide in an edge-lit backlight using a regional partition approach". *Optical Engineering* 46 (4): 043002–1–043002–9.

Chang, J.-G., Y.-B. Fang, and S.-P. Ju. 2009. "Using a generalized molecular dynamics force model for random microstructure generation of different aspect-ratios and orientation for use in the optical design of an LED edge-lit backlight". *Computer Physics Communications* 180 (8): 1259–1270.

Chang, J.-G., and C.-T. Lee. 2007. "Random-dot pattern design of a light guide in an edge-lit backlight: integration of optical design and dot generation scheme by the molecular-dynamics method". *Journal of the Optical Society of America A* 24 (3): 839–849.

Choi, S. D., A. R. Kumar, and A. Houshyar. 2002, December. "A simulation study of an automotive foundry plant manufacturing engine blocks". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 1035–1040. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Idé, T., H. Mizuta, H. Numata, Y. Taira, M. Suzuki, M. Noguchi, and Y. Katsu. 2003. "Dot pattern generation technique using molecular dynamics". *Journal of the Optical Society of America A* 20 (2): 248–255.

Imamichi, T., and H. Nagamochi. 2007. "A Multi-sphere Scheme for 2D and 3D Packing Problems". In *SLS 2007: Proceedings of Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, edited by T. Stützle, M. Birattari, and H. H. Hoos, Volume 4638 of *Lecture Notes in Computer Science*, 207–211. Heidelberg: Springer.

Imamichi, T., and H. Nagamochi. 2008. "Performance Analysis of a Collision Detection Algorithm of Spheres Based on Slab Partitioning". *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E91-A (9): 2308–2313.

Kamnerdtong, T., S. Chutima, and J. Parirukvijit. 2008, December. "The improvement of deformations and charateristics of HGA during clamping using finite element analysis". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 1335–1343. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Liu, C. 2000. "Finite element analysis of the effect of sequential cuts and toolchip friction on residual stresses in a machined layer". *International Journal of Mechanical Sciences* 42 (6): 1069–1086.

Liu, D. C., and J. Nocedal. 1989. "On the limited memory BFGS method for large scale optimization". *Mathematical Programming* 45 (1): 503–528.

Pickardt, C., J. Branke, T. Hildebrandt, J. Heger, and B. Scholz-Reiter. 2010, December. "Generating Dispatching Rules for Semiconductor Manufacturing to Minimize Weighted Tardiness". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 2504–2515. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Sharda, B., and S. J. Bury. 2010, December. "Bottleneck analysis of a chemical plant using discrete event simulation". In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 1547–1555. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tezuka, S. 1993. "Polynomial arithmetic analogue of Halton sequences". *ACM Transactions on Modeling and Computer Simulation* 3:99–107.

Villarreal, M. G., R. Mulyana, J. M. Castro, and M. Cabrera-Ríos. 2008, December. "Simulation Optimization Applied to Injection Molding". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 1995–2003. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

TAKASHI IMAMICHI is a researcher in IBM Research - Tokyo. He belongs to the Analytics and Optimization group. He received the Bachelor of Engineering, Master of Informatics, and Doctor of Informatics degrees from Kyoto University in 2004, 2006, and 2009, respectively. His research interests include cutting and packing problems and metaheuristics. Dr. Imamichi is a member of the Operations Research Society of Japan and the Scheduling Society of Japan. His email address is imamichi@jp.ibm.com.

HIDETOSHI NUMATA is a researcher in IBM Research - Tokyo. He received his B.S., M.S., and Ph.D. degrees in basic science from the University of Tokyo in 1992, 1994, and 1998, respectively. In 1998, he joined the display technology group in IBM Tokyo Research Laboratory. He is currently a member of the optical interconnect technology group in IBM Research - Tokyo. He is a member of OSA, the Japan Society of Applied Physics, and the Institute of Electronics, Information and Communication Engineers. His email address is hnumata@jp.ibm.com.

HIDEYUKI MIZUTA is a staff researcher of IBM Research Tokyo. He received B.S., M.S., and Ph.D. degrees in Physics from the University of Tokyo. He is a member of IPSJ and ACM SIGSIM. His research interests include dynamic economic-social systems with heterogeneous agents, SSME (Services Science, Management and Engineering) and smarter cities for sustainable society. His email address is e28193@jp.ibm.com.

TSUYOSHI IDÉ is a Senior Researcher and the manager of the Analytics & Optimization group in IBM Research - Tokyo. He received M.Sc. and Ph.D. degrees in theoretical physics from the University of Tokyo, in 1997 and 2000, respectively. In 2000, he joined IBM Research as a researcher in display technology. Since 2003, he has been working on data mining research. His current research interest includes knowledge discovery techniques from time-series data. His email address is goodidea@jp.ibm.com.