LightTrace: Auto-router for Designing LED Based Applications with Conductive Inkjet Printing

Tung D. Ta
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku
Tokyo, Japan 113-8656
tung@akg.t.u-tokyo.ac.jp

Fuminori Okuya
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku
Tokyo, Japan 113-8656
okuya23@akg.t.u-tokyo.ac.jp

Yoshihiro Kawahara
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku
Tokyo, Japan 113-8656
kawahara@akg.t.u-tokyo.ac.jp

Figure 1: Multiple LEDs are routed automatically to balance brightness. A group of 32 LEDs are placed to form text “SCF”. LightTrace provides balance in brightness to all the LEDs in the design, whereas for the outcome obtained using the auto-router in Eagle, some LEDs shine notably brighter than others.

ABSTRACT

The combination of graphic design and printable electronic circuits has brought a large variety of customized elements to create interactive applications. The designers aim to focus on the creative and aesthetic aspects when crafting such applications. However, current technology forces designers to consider the discouraging and complicated electrical behavior of the related circuits. Even a simple application as activating a group of LEDs by means of conductive ink presents specific challenges. For instance, given the relatively high resistance of conductive ink, it is difficult to generate a circuit pattern that evenly lights up the LEDs. Moreover, wiring a large number of LEDs is error-prone and not straightforward. Electronic-circuit designers often use built-in auto-routers from Computer Aided Design tools to route electronic circuit boards. However, these auto-routers usually neglect the resistance of the metallic pattern. This is not applicable in the case of conductive ink where its intrinsic resistance is not negligible. Nonetheless, this characteristic of the conductive ink can be used to eliminate the need for ballast resistors which are used to regulate the current through each LED when they are wired by highly conductive materials (e.g., copper in PCB or highly conductive ink). Our proposal, called LightTrace, is an LED auto-router implemented as an Adobe Illustrator extension. LightTrace computationally designs conductive patterns and adjusts their resistance to connect and evenly light up the LEDs in a single-layer sheet of paper without requiring additional resistors. We use the Traveling Salesman Problem

algorithm to achieve the shortest cross-less path through all LEDs, thus reducing the amount of ink consumed.

CCS CONCEPTS

• **Hardware** → **Wire routing**; **Software tools for EDA**; • **Human-centered computing** → *User interface toolkits*; *Systems and tools for interaction design*; *User interface design*;

KEYWORDS

Silver nano-particle ink, auto-router, LED, Traveling Salesman Problem, brightness balance

ACM Reference format:

Tung D. Ta, Fuminori Okuya, and Yoshihiro Kawahara. 2017. LightTrace: Auto-router for Designing LED Based Applications with Conductive Inkjet Printing. In *Proceedings of SCF '17, Cambridge, MA, USA, June 12-13, 2017*, 10 pages.
DOI: <http://dx.doi.org/10.1145/3083157.3083160>

1 INTRODUCTION

The introduction of sintering-free conductive ink [Kawahara et al. 2013] has attracted a lot of interest from groups of researchers, hobbyists, and designers. In addition, the fact that silver nanoparticle ink is low-cost and readily available on the market has made the fabrication of flexible electronic circuits accessible and simple. Used by hobbyists, artists, or even children, the conductive pen [AgIC 2017; BareConductive 2017; ElectronInks 2017] has become a convenient tool for freehand drawing of electronic circuits. Moreover, the conductive pen has been adopted to build electronic applications for educational purposes and paper art, or quick-fix simple electronic circuits.

Besides, the conductive ink can be used in inkjet printers to print high-resolution conductive patterns in a very short time. Conductive inkjet printing has been widely used by researchers in

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SCF '17, Cambridge, MA, USA

© 2017 Copyright held by the owner/author(s). 978-1-4503-4999-4/17/06...\$15.00
DOI: <http://dx.doi.org/10.1145/3083157.3083160>

(a) Front side showing the painting and LEDs lit up from behind. ©Jie Qi

(b) Back side having electronic circuits to power the LEDs. ©Jie Qi

Figure 2: Pu Gong Ying Tu (Dandelion Painting) [Qi 2012] ©Jie Qi

rapid prototyping of HCI-related applications such as the Extension Sticker [Kato and Miyashita 2014], the Inkantatory paper [Tsujii et al. 2014], touch interfaces [Gong et al. 2014; Karatas and Gruteser 2015] or embedded in 3D printing processes to fabricate electrically functional 3D printed objects [Swensen et al. 2015; Wasserfall et al. 2016]. In addition, conductive inkjet ink is used by designers to create interactive lighting applications with LEDs [AgIC 2015]. However, when the number of LEDs increases, wiring them becomes complex. Furthermore, the non-negligible resistance of the conductive-inkjet printed patterns causes an uneven lighting up of the LEDs (Fig. 1b).

In the work “Pu Gong Ying Tu (Dandelion Painting)” [Qi 2012] (Fig. 2), many LEDs were placed behind a painting to create interactive electronic art. This work was manually designed using LEDs and copper tape. Fig. 2b shows the complicated electronic circuit based on copper tape that is behind the picture. Instead of manually wiring, a printable digital design using conductive ink will obviously simplify the replication of this work. However, we can suppose that the author of this work would not have done it using conductive ink because it is difficult to achieve a cross-less wiring pattern. In addition, the resistance of this ink and pastes are higher than that of copper, hence it should be thoroughly considered in the design stage.

On the other hand, although the cost of printing with silver nanoparticle ink is relatively low, it is still beneficial to reduce the amount of conductive ink consumed, which helps to lower the running costs of conductive inkjet printing systems in the long run. Moreover, reducing ink consumption leads to a smaller printing area as well as shorter printing time.

We propose LightTrace – an auto-router that is integrated into Adobe Illustrator [Adobe 2017] as an extension to help users design LED-based applications with conductive ink. We chose Adobe

Illustrator because this tool is widely used by designers to create applications with conductive ink. Our contributions are summarized as follows:

- The auto-router is based on the Traveling Salesman Problem (TSP). Thus, it finds the shortest cross-less pattern to evenly light up multiple LEDs in a single-layer sheet of paper, hence minimizing conductive ink consumption. In addition, the obtained design does not require ballast resistors.
- A fully automated tool to design LED-based applications with conductive ink. It allows the user to drag and drop LEDs and a power source and then the auto-router appropriately wires all the LEDs.

2 RELATED WORKS

2.1 Classical Electronic Circuit Routing Algorithms

Routing electronic circuits is one of the most well studied fields in Computer Aided Design (CAD) systems. In order to handle increasingly complex electronic circuit designs, auto-routers must be continuously improved. There are many efficient algorithms to route high-density circuits that can be roughly categorized into two groups as follows:

2.1.1 Grid-based. This approach assumes that all the components’ terminals are aligned to a predefined grid and routing traces are also placed on this grid. The goals are to route electronic nets and usually to minimize the total length of the routing traces. The width of the wiring pattern is usually fixed before routing. Algorithms in this category include maze routers [Hadlock 1977; Lee 1961], algorithms based on line search [Hightower 1969; Mikami and Tabuchi 1968], and algorithms based on building the Steiner tree of a weighted graph [Abboud et al. 2007].

2.1.2 Gridless. Different to grid-based approaches, gridless routing algorithms [Finch et al. 1985; Lauther 1980; Schiele et al. 1990] do not rely on the alignment of the components’ terminals on any grid. Electronic components can be freely placed on the routing floor. In addition, the routing traces are not restricted to a specific width or angle. Moreover, it is possible to dynamically change the routing width to meet routing requirements.

These types of routing algorithm are implemented in CAD or Electronic Design Automation tools (e.g., Eagle [Autodesk 2017]), which are commonly used by electronic circuit experts to design circuits and auto-route components. However, most of the auto-routers assume that the resistance of metal patterns used to wire electronic components is negligible. In fact, the dimension of metal patterns is not considered as long as the electronic scheme is routable. In contrast, dimension plays an important role when routing using conductive ink as it affects the printed pattern resistance, which is nonnegligible.

Regarding LED-based applications, routing with traditional auto-routers results in uneven lighting up of LEDs as shown in Fig. 1b and Fig. 11b

2.2 Routing Based on Tree Shape

In a sense, the problem of LEDs brightness balancing resembles that of zero-skew clock routing that is well studied in the design of high-performance VLSI systems. The zero-skew clock routing problem occurs in synchronous VLSI design, where it is necessary to minimize the difference in arrival times of the clock signal (or length) from a clock source to synchronous elements. In many approaches to achieve zero-skew clock distribution, such as [Chao et al. 1992; Ramanathan et al. 1989; Tsay 1993], routing based on clock trees [Menezes et al. 1993] is one of the most common methods. However, this cannot be the solution to balance the LEDs brightness, because the adoption of routing approaches based on a tree shape requires routing on multiple layers. Although this requirement is easy to satisfy in traditional PCB or VLSI fabrication processes, it is not easy to obtain multiple-layer circuits using conductive ink. Approaches to produce flexible double sided electronic circuits with conductive ink, as presented in [Andersson et al. 2014; Ta et al. 2015], complicate the fabrication of LED-based applications. It is preferable to both route and balance LEDs brightness in a single layer. Furthermore, using tree-based routing makes it more difficult to optimize the amount of ink used to print the wiring patterns.

2.3 Routing in Rapid Prototyping of Electronic Circuits

To support designing with flexible electronic circuits, Circuit Stickers [Hodges et al. 2014] proposed a tool to help users design with a variety of electronic components using either conductive tape or ink. Ellustrate [Lo et al. 2016] proposed another tool to guide users in the design, sketch, and fabrication of aesthetic electronics. However, these tools do not support auto-routing.

PaperPulse [Ramakers et al. 2015] is another designing tool to help creating interactive applications using paper, electronic components, and conductive inkjet ink. Similar to Midas [Savage et al. 2012], a tool to design and fabricate capacitive touch sensors, PaperPulse uses A* search algorithms, such as Lee's algorithm [Lee 1961], to route electronic nets. However, these tools ignore the resistance existing in conductive inkjet ink, which in many cases, severely affects the functionality of an electronic circuit.

From another point of view, ConductAR [Narumi et al. 2015] proposed using a computer vision technique to analyze the resistance of drawn or printed conductive ink patterns. Based on this resistance analysis, it has a function to suggest the optimized trace width when the optimal current at a specific point on the path is determined. However, ConductAR is an interactive and ad-hoc tool that is incapable of generating 2D paths to connect LEDs on demand.

3 PROBLEM FORMULATION AND SOLUTION

In order to evenly light up multiple LEDs, we need to assign resistances from the power source to each LED so that they receive the same amount of power. This resistance assignment is even more critical when the LEDs are connected in parallel. An imbalance in the resistance of each LED branch will not only cause uneven brightness but also will burn out the LEDs due to overcurrent. Existing auto-routers assume that the resistance of wiring patterns

Figure 3: A naive approach to balance the LEDs brightness

are negligible, and that to protect and balance brightness of multiple LEDs, we should use a ballast resistor for each LED. These assumptions do not hold for conductive ink, given its nonnegligible sheet resistance. In addition, when using conductive ink, we aim to avoid using ballast resistors as they increase the complexity of producing LED-based applications. In fact, the conductive printed traces should be used to act as the required ballast resistors. Consequently, by means of computational design, we can use the apparent disadvantage (i.e., high sheet resistance) of the conductive ink to our benefit by printing ballast resistors. This is extremely useful for unexperienced users as they will not need to select the ballast resistor values or connect any additional resistors for the LEDs.

It is possible to guarantee that all the LEDs evenly light up by connecting them in series to the power source. However, using this approach to a large number of LEDs, the required voltage for an LED-based application is enormous. For this reason, in our approach we connect all the LEDs in parallel with the power source and balance the brightness by adjusting the conductive traces resistance.

A naive approach to balance the LEDs brightness is to rectilinearly connect them in parallel to the power source and guarantee that the total resistance to each LED has the same value (as shown in Fig. 3). However, it is easy to see that this naive approach is not optimized in terms of either the ink consumption or the required routing space.

The problems our approach aims to solve are:

- *Routing*: finding a route which optimally connects all the LEDs to the power source. This route should be cross-less and, ideally, the shortest path through all the LEDs and the power source to reduce the conductive ink consumption.
- *Brightness Balancing*: guaranteeing that all the LEDs will have the same brightness by adjusting the conductive traces resistance.

We will address these problems respectively in the following two subsections. The general flow of our routing algorithm is shown in Fig. 4. For simplicity in coding, we assume that all the LEDs have the same dimension, color, forward voltage, and forward current.

3.1 Problem 1 - Routing

This problem addresses the inconvenience and difficulty of manually wiring multiple LEDs.

Problem statement: For a group of n LEDs ($L_i, i = 1, 2, \dots, n$), and the two poles of a power source (L_0, L_{n+1} respectively denoting anode and cathode), find a route that connects the anode and cathode terminals of every L_i to the power source pads L_0 and L_{n+1} , respectively, and at the same time, reduces ink consumption.

Figure 4: Routing process flow of LightTrace

Solution: To reduce ink consumption, we need to find the shortest path to connect all the LEDs to the power source. To achieve this, it is possible to recursively use some A* search algorithm. However, this algorithm can find the shortest path between two vertices, not the shortest path through all the LEDs. Theoretically, the Minimum Steiner Tree is the best solution to connect multiple vertices obtaining the minimal total length [Bern and Graham 1989]. Nevertheless, because we have to route the anode and cathode nets separately, it is unfeasible to use the Minimum Steiner Tree for both nets in a single-layer circuit. Thus, to address the routing problem above, we propose to route all the LEDs by solving the corresponding Euclidean-metric TSP, where the start and end points are fixed to L_0 (anode pad of the power source) and L_{n+1} (cathode pad of the power source), respectively (Fig. 4b). The router starts at L_0 , finds a TSP route through all LEDs L_i and ends the route at L_{n+1} . The distance between L_0 and L_{n+1} is ignored when the path length is calculated. As Euclidean-metric TSP is an NP-Complete problem [Papadimitriou 1977], a polynomial time solution is unknown. Therefore, we can only approximate the optimal path for the problem. There are several commonly used methods to solve the TSP such as the Christofides' algorithm [Christofides 1976], simulated annealing algorithm [Geng et al. 2011] and genetic algorithms (GAs) [Grefenstette et al. 1985]. For simplicity in the implementation and flexibility in evaluating the best route, we use a GA to approximate the shortest path through all the LEDs.

In the GA used to solve the TSP, an *individual* is a tour candidate that goes from L_0 , through all $L_{i=1,2,\dots,n}$, to L_{n+1} . Each individual is represented (encoded) by the array shown in Fig. 5. This representation is convenient for our case, where the start and end points are fixed. Our GA will try to find the order of the $L_{i=1,2,\dots,n}$ to achieve the shortest path. For each generation, we have a *population*

Table 1: Parameters of the GA to solve the TSP

Population Size	50
Mutation Rate	0.015
Mutation Scheme	Ordered Changing [Obitko 1998]
Crossover Scheme	Two-point Crossover [Obitko 1998]
Selection Scheme	Tournament [Goldberg and Deb 1991]
Tour Size	5
Fitness	Reciprocal of Tour Length
Number of Generations	10,000
Convergence Threshold	100

Convergence Threshold: the maximum number of consecutive generations that have the same fitness value

Figure 5: Genetic encoding

Figure 6: Equivalent circuit of multiple LEDs routed

consisting of 50 *individuals*. In order to evolve from one generation to the next, we implemented the corresponding *selection*, *mutation*, and *crossover* processes. The *fitness* of each *individual* is the reciprocal of the total length of its corresponding tour [Paul 2005]. Thus, an *individual* with higher *fitness* is better. At each generation, we run selection, mutation, and crossover, and then extract the fittest *individual* (tour) from the *population*. Our TSP solver will stop when either no better *individual* is generated during 100 consecutive generations or the total number of generations reaches 10,000. Although not implemented in this paper, these parameters might be accordingly adjusted depending on the number of LEDs. The detailed parameters setting for our GA implementation is given in Table 1. Because the GA might provide a crossing path that is by no means the optimal TSP path, we iteratively apply the 2-opt [Potvin 1996] heuristic to eliminate any crossing edges. The final routing step is to add conductive patterns along the TSP path (Fig. 4b) to route the LEDs terminals to the power pads so that they are connected, as shown in Fig. 4c.

3.2 Problem 2 - Brightness Balancing

In order to evenly light up multiple LEDs, the currents that flow through each LED should be same. This can be achieved by adjusting the resistance from the power source to each LED. An electronic circuit routed by our TSP router has the equivalent circuit as shown in Fig. 6.

Figure 7: Plug-in integrated into Adobe Illustrator

As mentioned above, all the LEDs share the same characteristics. In this circuit, the current flowing through each LED has the same magnitude I_F , which is the nominal forward current of every LED. Corresponding to this current, the voltage drop on each LED is V_F . Based on the equivalent circuit, the brightness balancing problem can be re-stated as follows:

Problem statement: Given n LEDs that are connected to a V_0 power source as shown in Fig. 6, determine all the resistances R_i , $i = 0, 1, 2, \dots, 2n-1$ so that the current through each LED is equal to I_F .

Solution: This problem can be solved by applying the nodal equation method on the equivalent circuit. In order to balance the currents through the LEDs (i.e., $I_i = I_F$, $i = 1, 2, \dots, n$), the resistances must satisfy the following two equations:

$$\begin{cases} (n-i)R_i = iR_{n-1+i}, \\ R_0 + R_{2n-1} = \frac{V_0 - V_F - I_F \sum_{i=1}^{n-1} (n-i)R_i}{nI_F} \end{cases}$$

There are many solutions of R_i that satisfy these two equations. In our implementation, we assume that $R_i = R_{2n-1-i}$ where $i = 0, 1, 2, \dots, n-1$. With a specific source voltage V_0 chosen by the user, the values of all the R_i are determined and easily calculated.

Generate conductive pattern having the required resistance: Based on the results above, we can generate a printing pattern to satisfy these resistance constraints. The conductivity of conductive ink is characterized by its sheet resistance R_s . The resistance between the two ends of an $l \times w$ strip is calculated as $R = R_s \frac{l}{w}$. In our routing problem, because the size of an LED is small, l can be approximated as the distance between two connected LEDs, R is the resistance required between these LEDs, and R_s is a known value that depends on the conductive ink. Thus, we can easily derive the width of the conductive pattern that connects these two LEDs. Outcomes of the routing algorithm are shown in Fig. 1a, Fig. 4d, and Fig. 11a.

4 IMPLEMENTATION

The implementation of the above solutions is included as an extension for Adobe Illustrator¹.

¹Source code can be found in this repository: <https://github.com/tatung/LightTrace>

4.1 Adobe Illustrator Extension

Adobe Illustrator is commonly used by designers to create artwork. In order to avoid forcing the user to learn a completely new tool, we have integrated our auto-router into Adobe Illustrator. Both the LED shape and power source pads are available and integrated into the software toolbox. After placing the LEDs over the artboard, the user can automatically route these LEDs and the power source by running our routing script. A route layout will be generated on top of the artboard. This layout should be printed with conductive ink (Fig. 7).

4.2 LED Rotation

In the section about routing, we assume that each LED and power source pad are represented by points on a 2D plane. This simplifies the implementation of the TSP solver. However, when routing the LEDs anode and cathode to the corresponding poles of the power source, we need to consider the LEDs specific dimension. As shown in Fig. 6, we have two branches in the circuit (i.e., the anode branch composed of $R_{i=0,1,\dots,n-1}$ and the cathode branch composed of $R_{i=n,n+1,\dots,2n-1}$) that connect the LEDs to the power source. These two branches must not overlap during the routing process. We achieve this by rotating each LED around its center to an appropriate angle. Let three LEDs be connected by the TSP routing as shown in Fig. 8a. We define the axis of an LED as the vector from its anode to its cathode. Then, TSP angle θ of the LED is the angle formed by the two edges of the TSP route that connect the LED. We rotate each LED so that its axis is parallel to the bisector line of the TSP angle θ as shown in Fig. 8b. The axis direction of an LED, which defines the position of its anode and cathode, is selected so that the anode branch does not cross the cathode branch (Fig. 8c).

Figure 8: Rotation of LEDs to route the anode and cathode branches

4.3 Solving the Excessively Narrow Conductive Trace Problem

Depending on the distance and resistance between two LEDs, the width of the conductive pattern that connects these LEDs need to be adjusted. However, there are many cases where the conductive pattern is excessively narrow. In addition, depending on the type of conductive ink and printing technology used, the minimum tolerable width $W_{\text{threshold}}$ varies. Consequently, a conductive pattern having width $w < W_{\text{threshold}}$ presents a very high resistance or

(a) An excessively narrow strip has a very high resistance or is not conductive at all

(b) Equivalent meander line having the same end-point distance but that is longer and wider

Figure 9: Conversion of an excessively narrow conductive strip to its equivalent meander line. By proportionally increasing the width and length, the resistance between the end points A and B of the meander line in (b) is the same as that between the end points of the narrow strip in (a)

Figure 10: Meander lines of appropriate width are printed using conductive ink to achieve the desired resistances

is not conductive at all. In these cases, we need to appropriately increase the width and length of the conductive pattern so that its resistance does not change. LightTrace achieves this by replacing excessively narrow conductive strips with equivalent meander lines as shown in Fig. 9. A meander line is defined by the following parameters:

- l_m : distance between the two ends of the line
- w_m : line width
- N : number of pulses
- G : pulse width
- S : space between two consecutive pulses
- H : pulse height

We define the equivalent meander line of an $l \times w$ conductive strip with resistance R as a meander line that preserves distance l and resistance R between the two ends of the line. In other words,

it satisfies the following relations:

$$\begin{cases} l_m = l & \text{distance preservation} \end{cases} \quad (1)$$

$$\begin{cases} R_m = R & \text{resistance preservation} \end{cases} \quad (2)$$

$$\begin{cases} w_m = kw & k \geq \frac{W_{\text{threshold}}}{w} \end{cases} \quad (3)$$

Among the meander line parameters, space S between two consecutive pulses has a lower bound $S_{\text{threshold}}$ that depends on the conductive ink and printing technology used. Thus, setting $S < S_{\text{threshold}}$ would cause an electrical short-circuit between the edges of two consecutive pulses. This will reduce the resistance between two arbitrary points on these pulses that consequently leads to an incorrect value of the final resistance. Therefore, we need to guarantee that $S \geq S_{\text{threshold}}$.

Assume that we need to convert an $l \times w$ strip pattern ($w < W_{\text{threshold}}$) to its equivalent meander line with width $w_m \geq W_{\text{threshold}}$. Because $w_m > w$, in order to make $R_m = R$, we need to increase the length between the two ends of the line. This length can be controlled by selecting appropriate values for number of pulses N , pulse width G , space between two consecutive pulses S , and pulse height H .

In our implementation, for simplicity and compactness of the equivalent meander line, we consider $w_m < H < H_{\text{threshold}}$ and $S = G$. From these two assumptions and Equations (1), (2), and (3), we have the formulas to obtain the parameters of the equivalent meander line as follows:

$$N = \left\lceil \min \left(\frac{l(w_m - w)}{2ww_m}, \frac{l - (2w_m + S_{\text{threshold}})}{2(2w_m + S_{\text{threshold}})} \right) \right\rceil,$$

$$S = G = \frac{l}{2N + 1}, \quad H = \frac{l(w_m - w)}{2wN}$$

Fig. 10 shows the printing of an equivalent meander line in the routing to obtain the desired resistances.

5 EXPERIMENTS

Settings for all the experiments in this section are listed in Table 2.

Table 2: General setting for the experiments

LED	LTST-C150CKT [LITE-ON Optoelectronics 2006]
Substrate	NB-RC-3GR120 [Mitsubishi Paper Mills Ltd. 2013a]
Conductive Ink	NBSIJ-MU01 [Mitsubishi Paper Mills Ltd. 2013b]
Inkjet Printer	Epson PX-S160T
Sheet Resistance R_s	0.13 $\Omega/\square$ (ohm-per-square)

5.1 Brightness Balance Evaluation

In order to test our Adobe Illustrator plug-in to provide even brightness for multiple LEDs, we placed 22 red LEDs to form the text “AB”, and then connected them using printed conductive inkjet ink as shown in Fig. 11. The conductive pattern in Fig. 11a was generated using our proposed LightTrace, whereas the pattern in Fig. 11b was generated using the auto-router in Eagle with trace width and clearance set to 1 mm. Both circuits are powered by a

(a) Routing by LightTrace to auto-route and adjust the resistance to LEDs so that they evenly light up

(b) Routing by the auto-router in Eagle with trace width and clearance set to 1 mm

Figure 11: Multiple LEDs are automatically routed to balance brightness. A total of 22 LEDs are placed to form the text “AB”. It is clear that using LightTrace resulted in better brightness balance for all the LEDs.

Figure 12: Voltage drop and current through each LED when routed by LightTrace

Figure 13: Voltage drop and current through each LED when routed by the auto-router in Eagle

6 V regulated DC power supply (TEXIO PA36-3B [TEXIO 2017]). The voltage drop on each LED was measured using a digital tester (Hozan DT-124 [Hozan 2017]). In order to obtain the value of the current flowing through an LED at a specific voltage drop, we measured the current through that LED for different voltage drop values in the range of 0.3 V ~ 2 V (with 1 mV steps) using a source meter (Keithley 2450 [KEITHLEY 2017]).

As shown in Fig. 12 and Fig. 13, LightTrace provides a much better balance of voltage drop on each LED. All the voltage drop values remain stable and are larger than the minimum required voltage (i.e., above the red line in Fig. 12) to light up the LEDs. On the other hand, for the pattern generated using Eagle, the voltage

drop values on the LEDs present notable fluctuations. Most of these values are not high enough to light up the LEDs (i.e., lower than the red line in Fig. 13). This results in uneven brightness of the LEDs as shown in Fig. 11b. Using LightTrace, voltage drop and current flowing through each LED, as seen in Fig. 12, are not only large enough to light up the LEDs but also have smaller standard deviation, resulting in better brightness balance of the LEDs as shown in Fig. 11a.

5.2 TSP Routing Time Performance

In order to evaluate the time performance of our routing algorithm, we ran it considering different number of LEDs and measured the

Table 3: Environment setup to evaluate time performance of the TSP routing algorithm

Operating System	Mac OS X Sierra 10.12.2
Processor	2.7 GHz Intel Core i5
Memory	8 GB 1867 MHz DDR3
Adobe Illustrator	CC 2017

Figure 14: Time performance of our TSP routing algorithm for different number of LEDs **Figure 15: Evaluation of ink consumption based on the total area of the conductive patterns considering different number of LEDs. The power source is set to 20 V and 30 V before routing. For each case, we made 11 samples with the same number of LEDs under different distributions. The area of the conductive pattern is averaged based over these samples**

time it takes to route all the LEDs. The experimental environment setup is presented in Table 3.

The experiment results in Fig. 14 shows that our TSP routing algorithm can route up to 60 LEDs in a convenient timeframe using a commodity laptop running Intel Core i5 at 2.7 GHz. However, it might be difficult to route a larger number of LEDs. The most time-consuming process is to find the TSP route through all the LEDs. This can be improved by applying optimization techniques for TSP solvers such as dynamic programming, heuristic fine-tuning, or Christofides' algorithm [Christofides 1976].

5.3 Ink Consumption

In this section, we evaluate the amount of ink needed to route multiple LEDs based on the area of the conductive pattern. To the best of our knowledge, we could not find any auto-router that is capable of routing and balancing brightness of multiple LEDs. The experiment on this section was conducted using our TSP router

Figure 16: Routing a large number of LEDs to create interactive lighting artworks. By controlling brightness of separated groups of LEDs, we can create a depth perspective in the pictures

only, for different number of LEDs. Multiple LEDs are randomly scattered using a uniform distribution on an A4-sized (210 mm × 297 mm) artboard. The power source is set to 20 V and 30 V before routing. Fig. 15 shows the ink consumption of the conductive patterns generated using LightTrace. As expected, routing more LEDs demands more ink. However, increasing the power source voltage helps to reduce the amount of ink needed. Reducing the pattern area implies higher resistance. In order to maintain the power constant, higher voltage should be provided to compensate the higher resistance effect. In future versions of LightTrace, the automatic determination of V_0 will be available for the benefit of users.

6 APPLICATIONS

6.1 Electronic Artwork with Multiple LEDs

Using a large number of LEDs can help to create impressive lighting artwork. In many artworks, hundreds of LEDs might be required [Qi 2012], and the routing for these LEDs is troublesome and error-prone. With LightTrace, user can rapidly route a lot of LEDs and then print the conductive pattern using a conductive ink inkjet printer (Fig. 16). We can also use transparent PET films to route different types of LEDs on different layers, hence providing the ability to mix multiple types of LEDs in a single artwork.

6.2 Characters and Emojis with LEDs and Conductive Ink

Characters and emojis are commonly used when designing interactive applications using LEDs. However, routing characters and

Figure 17: Alphabet characters and emoji using LEDs and conductive ink

Figure 18: Lighting greeting card generator using a smartphone application

sophisticated emojis is complicated and cumbersome even to an expert in electronic circuits. Using LightTrace, we are able to quickly route all characters in the alphabet table and some emojis such as smiley or frowning faces. Printed and powered characters and emoji examples can be seen in Fig. 1 and Fig. 17.

6.3 Quickly Made Greeting Card

Initially, our algorithm is implemented and integrated into Adobe Illustrator so that designers feel familiar with the working environment. However, our algorithm can be also implemented on different environments that are friendlier to novice users. Thus, we propose a greeting card designing smartphone application (Fig. 18) in which users can take a photo of the scene that they want to be the cover of the greeting card. Then, as another feature of our application, the user can drag and drop multiple LEDs on top of the photo. Finally, our application automatically routes and generates the designing files that the user can directly send to the printer (color and conductive ink inkjet printer are required). After printing, everything is ready to be assembled (real LEDs attachment, back and cover glue, as seen in Fig. 18).

7 DISCUSSION AND FUTURE WORK

Our LightTrace plug-in can automatically generate a ready-to-print conductive pattern, which after being printed using conductive inkjet ink, guarantees that all LEDs are evenly lit up. This not only reduces the effort of working with conductive inkjet ink but also helps users to focus more on the creative design. Not only LED-based applications can benefit from LightTrace but other 2-terminal

components with specified power consumption such as sensors, actuators and speakers can also be autorouted by our algorithm. Beside implementation in Adobe Illustrator, LightTrace can also be easily ported to other popular CAD systems such as Fritzing and KiCAD. However, LightTrace still faces some issues that need to be improved, including:

- *Routing time:* With a large number of LEDs (more than 100 LEDs), our auto-router consumes a lot of time. At this moment, this number is enough for most cases. However, in the case that a user would like to light up 1,000 LEDs or more to fill a whole wall or ceiling, a faster algorithm to solve the TSP is necessary. One solution is to apply better heuristics or dynamic programming to the TSP solver.
- *Scalability Evaluation:* Scalability of our auto-router must consider several parameters including the number of LEDs, voltage of the power source, sheet resistance of the conductive ink, total space that all LEDs occupy, and LEDs distribution.
- *Multicolor LEDs:* In this work, we focused on routing only one type of LED, hence considered the same dimension, color, and forward voltage and current. However, our auto-router can be extended to use multiple-color LEDs with different forward voltage and current. This can be achieved by solving the general nodal equations system for the circuit in Fig. 6.
- *Power consumption optimization:* As mention in the ink consumption section, the conductive pattern area, resistances, and power source voltage have a tight relationship. Based on this relationship, we can find the adequate tradeoff between resistance values and power source voltage required to drive multiple LEDs.

8 CONCLUSION

We proposed LightTrace as a designing tool for working with LEDs and conductive inkjet ink. This tool, implemented as an Adobe Illustrator extension, helps in reducing the effort of the routing process and guarantees that a large number of LEDs will evenly light up. Besides the deep integration into Adobe Illustrator, our extension provides:

- A ready-to-go tool for arranging multiple LEDs
- An auto-router using the TSP algorithm to find the shortest cross-less path to connect multiple LEDs on a single-layer sheet of paper
- A mechanism to adjust the width of the conductive patterns for controlling the resistance to each LED, hence allowing to control the brightness of each LED without requiring additional resistors.

In this research, we aim to help designers who have limited experience working with electronic circuits, to easily make interactive applications with LEDs and conductive inkjet ink. LightTrace allows designers to avoid working on complicated electronic circuits and lets them focus on creative designing.

ACKNOWLEDGMENTS

This work was supported by JST ERATO Grant Number JPM-JER1501, Japan.

REFERENCES

- Nadine Abboud, Martin Grottschel, and Thorsten Koch. 2007. Mathematical methods for physical layout of printed circuit boards: an overview. *OR Spectrum* 30, 3 (May 2007), 453–468. DOI: <https://doi.org/10.1007/s00291-007-0080-9>
- Adobe. 2017. Vector Design Tool - Adobe Illustrator. (2017). <http://www.adobe.com/products/illustrator.html>.
- AgIC. 2015. Slice of Light. (2015). <https://agic.cc/projects/1?locale=en>.
- AgIC. 2017. AgIC Pen. (2017). <https://shop.agic.cc/collections/products/products/circuit-marker>.
- Henrik A Andersson, Anatoliy Manuilskiy, Stefan Haller, Magnus Hummelgård, Johan Sidén, Christine Hummelgård, Håkan Olin, and Hans-Erik Nilsson. 2014. Assembling surface mounted components on ink-jet printed double sided paper circuit board. *Nanotechnology* 25, 9 (Feb. 2014), 094002. DOI: <https://doi.org/10.1088/0957-4484/25/9/094002>
- Autodesk. 2017. EAGLE PCB design software. (2017). <http://www.autodesk.com/products/eagle/overview>.
- BareConductive. 2017. BarePaint. (2017). www.bareconductive.com.
- Marshall W Bern and Ronald L Graham. 1989. The Shortest-Path Problem. *Scientific American* 260, 1 (Jan. 1989), 84–89. DOI: <https://doi.org/10.1038/scientificamerican0189-84>
- Ting-Hai Chao, Ting Hai Chao, Yu Chin Hsu, Yu-Chin Hsu, Jan-Ming Ho, Jan Ming Ho, A B Kahng, Kenneth D Boese, and Andrew B Kahng. 1992. Zero skew clock routing with minimum wirelength. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 39, 11 (1992), 799–814. DOI: <https://doi.org/10.1109/82.204128>
- Nicos Christofides. 1976. *Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*. Technical Report AD-A025 602. Carnegie-Mellon University.
- ElectronInks. 2017. CircuitScribe. (2017). <http://www.electroninks.com>.
- A C Finch, K J Mackenzie, G J Baldson, and G Symonds. 1985. A method for gridless routing of printed circuit boards. In *the 22nd ACM/IEEE Conference on Design Automation*. ACM Press, New York, New York, USA, 509–515. DOI: <https://doi.org/10.1145/800139.804593>
- Xiutang Geng, Zhihua Chen, Wei Yang, Deqian Shi, and Kai Zhao. 2011. Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing Journal* 11, 4 (2011), 3680–3689. DOI: <https://doi.org/10.1016/j.asoc.2011.01.039>
- David E Goldberg and Kalyanmoy Deb. 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms*. Morgan Kaufmann, 69–93. DOI: <https://doi.org/10.1016/B978-0-08-050684-5.50008-2>
- Nan-Wei Gong, Jürgen Steimle, Simon Olberding, Steve Hodges, Nicholas Edward Gillian, Yoshihiro Kawahara, and Joseph A. Paradiso. 2014. PrintSense: a versatile sensing technique to support multimodal flexible surface interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, Toronto, Canada, 1407–1410. DOI: <https://doi.org/10.1145/2556288.2557173>
- John Grefenstette, Rajeev Gopal, Brian Rosmaita, and Dirk Van Gucht. 1985. Genetic algorithms for the traveling salesman problem. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications*. 160–168.
- F O Hadlock. 1977. A shortest path algorithm for grid graphs. *Networks* 7, 4 (1977), 323–334. DOI: <https://doi.org/10.1002/net.3230070404>
- David W Hightower. 1969. A solution to line-routing problems on the continuous plane. In *DAC*. ACM Press, New York, New York, USA, 1–24. DOI: <https://doi.org/10.1145/800260.809014>
- Steve Hodges, Nicolas Villar, Nicholas Chen, Tushar Chugh, Jie Qi, Diana Nowacka, and Yoshihiro Kawahara. 2014. Circuit Stickers: Peel-and-stick Construction of Interactive Electronic Prototypes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, Toronto, Canada, 1743–1746. DOI: <https://doi.org/10.1145/2556288.2557150>
- Hozan. 2017. Hozan DT-124. (2017). <http://www.hozan.co.jp/E/catalog/Measuring/DT124.html>.
- Cagdas Karatas and Marco Gruteser. 2015. Printing multi-key touch interfaces. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, Osaka, Japan, 169–179. DOI: <https://doi.org/10.1145/2750858.2804285>
- Kunihiro Kato and Homei Miyashita. 2014. Extension sticker: A Method for Transferring External Touch Input Using a Striped Pattern Sticker. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, Honolulu, Hawaii, USA, 59–60. DOI: <https://doi.org/10.1145/2658779.2668032>
- Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. 2013. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, Zurich, Switzerland, 363–372. <http://dx.doi.org/10.1145/2493432.2493486>
- KEITHLEY. 2017. 2450 SourceMeter[®] SMU Instrument. KEITHLEY. http://www.tek.com/sites/tek.com/files/media/media/resources/2450-Datasheet_1KW-60904-0.pdf.
- Ulrich Lauther. 1980. A data structure for gridless routing. In *Proceedings of the 17th Design Automation Conference (DAC '80)*. ACM Press, New York, New York, USA, 603–609. DOI: <https://doi.org/10.1145/800139.804593>
- C Y Lee. 1961. An Algorithm for Path Connections and Its Applications. *IEEE Transactions on Electronic Computers* EC-10, 3 (1961), 346–365. DOI: <https://doi.org/10.1109/TEC.1961.5219222>
- LITE-ON Optoelectronics. 2006. *SMD LED Product Datasheet LTST-C150CKT*. LITE-ON Optoelectronics. http://optoelectronics.liteon.com/upload/download/DS-22-98-0002/S_110_LTST-C150CKT.pdf.
- Joanne Lo, César Torres, Isabel Yang, Jasper O'Leary, Danny Kaufman, Wilnot Li, Mira Dontcheva, and Eric Paulos. 2016. Aesthetic Electronics - Designing, Sketching, and Fabricating Circuits through Digital Exploration. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '16)*. ACM Press, New York, New York, USA, 665–676.
- Noel Menezes, Ashok Balivada, Satyamurthy Pullala, and Lawrence T Pillage. 1993. Skew Reduction in Clock Trees Using Wire Width Optimization. In *IEEE 1993 Custom Integrated Circuits Conference*. 961–964. DOI: <https://doi.org/10.1109/CICC.1993.590684>
- Koichi Mikami and Kinya Tabuchi. 1968. A computer program for optimal routing of printed circuit conductors. In *IFIP Congress* (2). 1475–1478.
- Mitsubishi Paper Mills Ltd. 2013a. Resin Coated Paper NBSIJ-MU01. (2013). http://www.k-mpm.com/agnanoen/agnano_media.html.
- Mitsubishi Paper Mills Ltd. 2013b. Silver Nano-particle Ink NBSIJ-MU01. (2013). http://www.k-mpm.com/agnanoen/agnano_ink.html.
- Koya Narumi, Steve Hodges, and Yoshihiro Kawahara. 2015. ConductAR: an augmented reality based tool for iterative design of conductive ink circuits. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, Osaka, Japan, 791–800. DOI: <https://doi.org/10.1145/2750858.2804267>
- Marek Obitko. 1998. Introduction to Genetic Algorithms. (1998). <http://www.obitko.com/tutorials/genetic-algorithms/crossover-mutation.php>.
- Christos H Papadimitriou. 1977. The Euclidean travelling salesman problem is NP-complete. *Theoretical Computer Science* 4, 3 (June 1977), 237–244. DOI: [https://doi.org/10.1016/0304-3975\(77\)90012-3](https://doi.org/10.1016/0304-3975(77)90012-3)
- Topon Kumar Paul. 2005. Traveling Salesman Problem by Genetic Algorithm. (2005). <http://www.iba.t.u-tokyo.ac.jp/english/Online/TSP/Method.html>.
- Jean-Yves Potvin. 1996. Genetic algorithms for the traveling salesman problem. *Annals of Operations Research* 63, 3 (1996), 337–370. DOI: <https://doi.org/10.1007/BF02125403>
- Jie Qi. 2012. Pu Gong Ying Tu (Dandelion Painting). (2012). <http://technolgie.com/pu-gong-ying-tu-dandelion-painting/>.
- Raf Ramakers, Kashyap Todi, and Kris Luyten. 2015. PaperPulse: An Integrated Approach for Embedding Electronics in Paper Designs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, Seoul, Korea, 2457–2466. DOI: <https://doi.org/10.1145/2702123.2702487>
- P Ramanathan, K G Shin, and Kang G Shin. 1989. A clock distribution scheme for nonsymmetric VLSI circuits. In *1989 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*. IEEE Comput. Soc. Press, 398–401. DOI: <https://doi.org/10.1109/ICCAD.1989.76978>
- Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: Fabricating Custom Capacitive Touch Sensors to Prototype Interactive Objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. ACM, Massachusetts, USA, 579–587. DOI: <https://doi.org/10.1145/2380116.2380189>
- W. L. Schiele, T. Kruger, K. M. Just, and F. H. Kirsch. 1990. A gridless router for industrial design rules. In *27th ACM/IEEE Design Automation Conference*. ACM Press, 626–631. DOI: <https://doi.org/10.1109/DAC.1990.114929>
- John P Swensen, Lael U Odhner, Brandon Araki, and Aaron M Dollar. 2015. Injected 3D electrical traces in additive manufactured parts with low melting temperature metals. In *2015 IEEE International Conference on Robotics and Automation*. IEEE, Seattle, Washington, USA, 988–995. DOI: <https://doi.org/10.1109/ICRA.2015.7139297>
- Tung Ta, Masaaki Fukumoto, Koya Narumi, Shigeki Shino, Yoshihiro Kawahara, and Tohru Asami. 2015. Interconnection and double layer for flexible electronic circuit with instant inkjet circuits. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM Press, New York, New York, USA, 181–190. DOI: <https://doi.org/10.1145/2750858.2804276>
- TEXIO. 2017. TEXIO Regulated DC Power Supply PA36-3B. (2017). <http://www.texio.co.jp/en/product/detail/5>.
- Ren-Song Tsay. 1993. An Exact Zero-Skew Clock Routing Algorithm. *Computer-Aided Design of Integrated Circuits and Systems* 12, 2 (1993), 242–249. DOI: <https://doi.org/10.1109/43.205004>
- Takahiro Tsujii, Naoya Koizumi, and Takeshi Naemura. 2014. Inkantatory paper: dynamically color-changing prints with multiple functional inks. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, Honolulu, Hawaii, USA, 39–40. DOI: <https://doi.org/10.1145/2658779.2659103>
- Florens Wasserfall, Daniel Ahlers, Norman Hendrich, and Jianwei Zhang. 2016. 3D-Printable Electronics - Integration of SMD Placement and Wiring into the Slicing Process for FDM Fabrication. In *Proceedings of the 27th Annual International Solid Freeform Fabrication*. 1826–1837.