1E-03

Traveling Salesman Problem Based Conductive Inkjet Printed Pattern Generation for Brightness Balancing of Multiple LEDs

Tung Ta D.† Fuminori Okuya† Yoshihiro Kawahara† Tohru Asami†
† Graduate School of Information Science and Technology, The University of Tokyo

1. Introduction

Conductive inkjet printing [1] has reduced a lot of laboring in designing interactive applications with electronic circuits. One of the most common applications of conductive inkjet printing is to light up a bunch of LEDs. This seems simple and easy enough for novice users. However, with multiple LEDs, routing is really troublesome and tedious, even seems impossible for most users. Furthermore, due to non-negligible resistance existed in conductive inkjet ink, some LEDs will be very bright while others will be dark (Figure 1b). An auto-router which considers these two problems will extremely benefit non-expert users. PaperPulse [2] or EDA tools (e.g. Eagle) are usually used to reduce laboring in electronic parts placement and routing. However, these tools assume that the resistance of wiring patterns is trivial. None of them solve the problem of LEDs brightness balancing. In this paper, we propose an auto-router as an Adobe Illustrator extension which computationally designs conductive patterns, adjusts its resistance to connect LEDs so that all of them will light up uniformly. Our contributions consist of:

- A Traveling Salesman Problem based auto-router which finds a near optimal pattern to make it possible to light up multiple LEDs uniformly with less ink usage.
- An integrated tool for designing LEDs based applications with conductive ink.

2. Problem Formulation & Solution

In order to light up multiple LEDs uniformly, we need to assign carefully resistances from the power source to each LED. Existing auto-routers assume that they can ignore the resistances of wiring patterns, and we can put ballast resistors for each LED. These assumptions are not true for conductive ink, given its non-negligible sheet resistance. Besides, using conductive ink, we do not want to put ballast resistors as it increases the complexity of making LEDs based applications. Conductive printed traces should act as ballast resistors. Our goals are 1) *Routing*, finding a route which optimally connect all LEDs to the power source and 2) *Brightness balancing*, assuring that all LEDs will have the same brightness by adjusting the resistances of conductive traces.

2.1 Problem 1 – Routing: Traveling Salesman

<u>Problem</u>: For a group of n LEDs ($L_i, i = 1, 2, \ldots, n$), and two

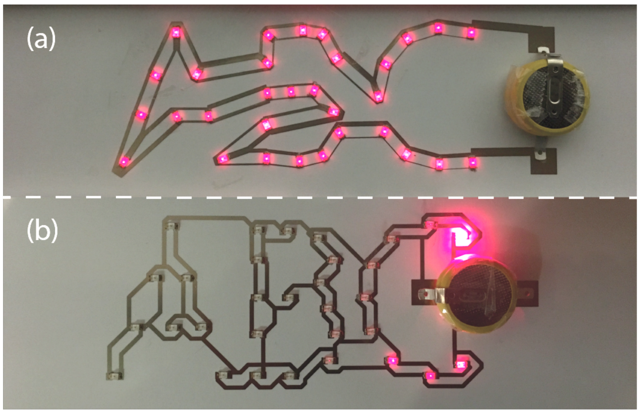

Figure 1. (a) "ABC" text routed by our algorithm. (b) "ABC" text routed by the auto-router in Eagle

poles of a power source (L_0, L_{n+1} respectively as anode, cathode), find a route that connects anodes and cathodes of L_i to L_0 and L_{n+1} respectively, and at the same time, reduces ink usage.

<u>*Solution*</u>: It is possible to recursively use A* search algorithms to connect all LEDs to the power source. However, this method does not optimize the ink usage. To reduce ink usage, we need to find the shortest path to connect all LEDs and the power source. We achieved this by solving Travelling Salesman Problem (TSP) for a graph of LEDs and the power source. A general solution for TSP is unknown. In our implementation, we use genetic algorithm to approximate the shortest path through all LEDs.

In genetic algorithm for solving TSP, an *Individual* is a tour candidate which goes from L_0, through all $L_{i=1,2,\ldots,n}$, to L_{n+1}. For each generation, we have a *Population* which consists of 50 Individuals. In order to evolve from a generation to the next generation, we implemented *selection*, *mutation* and *crossover*. The fitness of each individual is the reciprocal of the total length of the tour. At each generation, we run selection, mutation and crossover, then get the fittest individual (tour) of the population. Our TSP solver will stop when there is no better individual for 100 consecutive generations. Next step is to route LEDs terminals to power pads along the TSP route so that they are put in a parallel scheme (Figure 2)

2.2 Problem 2 – Brightness Balancing

In order to make multiple LEDs light up uniformly, the currents which flow through each LED should be same. This can be done by adjusting the resistances from the power source to each LED.

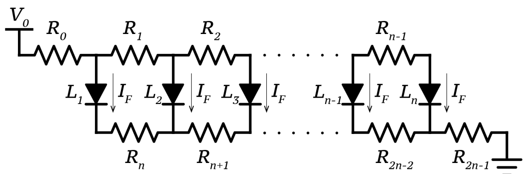

Figure 2. Equivalent circuit of multiple LEDs routed

An electronic circuit routed by our TSP router has an equivalent circuit as Figure 2.

In this circuit, the currents flow through all LEDs have the same magnitude I_F, which is the forward current written in the datasheet of the 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: Given *n* LEDs which are put in parallel to a V_0 power source as Figure 2, find all the resistances $R_{i=0,1,2,\ldots,2n-1}$ so that the current through each LED is equal to I_F.

Solution: This problem can be solved by applying nodal equation method on the equivalent circuit. In order to balance the currents through LEDs ($I_i = I_F$), the resistances need to satisfy:

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

There are many solutions of R_i which satisfy these two equations. In our implementation, we assume that $R_i = R_{2n-1-i}$ where $i = 0, 1, 2, \ldots, n-1$. With a specific value of V_0 chosen by user, these R_i are determined.

Generate conductive pattern from required resistance

Based on the results above, we can generate a printing pattern to satisfy these resistance constrains. Conductivity of a conductive ink is characterized by its sheet resistance R_s. The resistance between two ends of a $l \times w$ strip is calculated as $R = R_s \frac{l}{w}$. In our routing problem, *l* is the distance between two connected LEDs, R is the resistance required between these two connected LEDs, R_s is a known value which depends on the conductive ink. Thus, we can easily derive the width of the conductive pattern which connects these two LEDs. An output of the routing algorithm might look like Figure 1a.

3. Experiment

In order to test our plug-in in controlling brightness of multiple LEDs, we have put 31 red LEDs to form a text “ABC”, and then connected them by printed conductive inkjet ink (sheet resistance $R_s = 0.2\ \Omega/\square$) as in Figure 1.

In Figure 1a, 31 LEDs are connected by a pattern which was generated by our algorithm. In Figure 1b, 31 LEDs are connected by a pattern which was generated by the auto-router in Eagle with trace width and clearance are set to 1 mm. Both of these two samples are powered by a 6 V power source (two serial 3 V cell batteries).

Table 1. Mean and standard deviation of voltage drops and currents through all 31 LEDs.

	Our Algorithm (Figure 1a)	Existing Auto-router (Figure 1b)
Voltage Mean (V)	1.675 (σ = 0.056)	0.744 (σ = 0.665)
Current Mean (mA)	3.115 (σ = 2.439)	1.069 (σ = 4.679)

It is clear that by using our algorithm, voltage drops on each of the LEDs and the currents which flow through these LEDs have smaller standard deviation (σ), which indicates better balance of LEDs’ brightness. On the other hand, in Figure 1b, when using existing auto-router, many LEDs have low voltage drop which result in non-bright LEDs.

4. Future Work

Our proposal reduces laboring of manually routing multiple LEDs and allows user to focus more on creative design. Performance of TSP routing implementation can be improved by introducing dynamic programming into searching for the optimal path. Besides, in the case of densely distributed LEDs, when adjusting resistance, the widths of the resulting conductive patterns might be too big that it overlaps other LEDs. Better heuristic and LEDs clustering might help to solve this problem.

5. Conclusion

We proposed an algorithm to auto-route and balance the brightness of multiple LEDs based on TSP and resistance adjustment of conductive printed pattern. In this research, we aimed at helping designers, who have limited experience working with electronic circuits, to easily make interactive applications with LEDs and conductive inkjet ink.

Acknowledgement

This project is supported by JST Erato Kawahara Universal Information Network project.

Reference

[1] Y. Kawahara, S. Hodges, B. S. Cook, C. Zhang and G. D. Abowd, "Instant inkjet circuits: Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices," in *UbiComp '13*, Zurich, 2013.

[2] R. Ramakers, K. Todi and K. Luyten, "PaperPulse: An Integrated Approach for Embedding Electronics in Paper Designs," in *CHI 2015*, Seoul, 2015.