iBrush: Writing Chinese Characters with Flashlight and Wireless Sensor Grid

Andong Zhan†, Shen Li†, Wenchang Zhou†, Tao Li†, Panlong Yang‡, Xiaobing Wu† and Guihai Chen†
†State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
‡Institute of Communication Engineering, Nanjing, China

Abstract—Light is the widely available and pervasive resource in our daily life. Existing research shows that light can be used in localization in wireless sensor networks. Meanwhile, our former work also indicates that wireless sensor networks can track light sources. In this paper, we demonstrate iBrush, a system which allows users to write Chinese characters with a flashlight and a sensor grid.


I. INTRODUCTION

By using RF signals [1][2], ultrasounds [3][4] and even UWB [5], wireless sensor networks have succeeded in many applications of localization and tracking. However, systems using RF signals suffer accuracy fluctuations and systems using ultrasound or UWB need extra hardware. In addition, the iLamp system [6] shows that light, the widely available and pervasive source, can be used in localization. In our pervious work [7], we propose the iTracking system to show how to track lights from a bulb and a flashlight. In this paper, we demonstrate iBrush, a system that allows users to write Chinese characters with flashlight. We obtain the light-intensity-distance mapping with which we can calculate the distance between the sensor and the center of the disk area illuminated by the flashlight. We also draw lines of different widths based on the moving speed of flashlight. In our demonstration, players can use an LED flashlight to shine in different positions of a 5×5 sensor grid, as shown in Fig. 1, so as to write Chinese characters which can be displayed instantly on remote PC where the sink is plugged. Our demonstration bridges modern science and technology with ancient Chinese art.

II. SYSTEM DESIGN

Our system is composed of four main parts: (1) intensity-distance mapping, (2) central location calculation, (3) width setup, and (4) real-time updating. When a user uses a flashlight facing a sensor grid, the light disk may cover several sensors. The goal of the first part is to obtain the distance from an irradiated sensor to the center of the disk. The second part is to calculate the location of the disk center by analyzing the distances from multiple anchor sensors to the center. In (3), we use the moving speed of flashlight to control the width of the displayed stroke. Finally, we need to realize the real-time updates so as to minimize latency and improve user’s experience of writing.

A. intensity-Distance Mapping

To better estimate the distance from an arbitrary point inside a light disk to the center based on its light intensity, we sample several points inside the light disk for 100 ms per sample and 20 s in all, when an LED flashlight irradiates perpendicularly above a surface at a distance of 1 m. As shown in Fig. 3, the radius of the disk is almost 30 cm and the central light intensity is above 60 LUX (measured by TelosB Motes). Additionally, we use cubic spline interpolation to guarantee monotonicity and smoothness of the intensity-distance mapping. Fig. 3 shows that the curve is almost linear when distance ranges from 5 cm to 25 cm, which means the mapping is well suited to calculate distance based on light intensity.

Fig. 1. a flashlight shine in a 5×5 sensor grid

Fig. 2. Draw a line with changing widths
in Fig. 4. Experiences also show that our intensity-distance mapping for flashlight tracking can be used in the cases of non-perpendicular irradiations.

B. Central Location Calculation

We aim to calculate the central location of a light disk with high accuracy in a simple way.

In [7], we utilize the grid cell with the largest sum of light intensities of four sensors to localize the center of the disk. It is enough to write Arabic numerals, but the location we get is not stable when the center is near the borders of neighboring cells. We need higher accuracy and stability to write Chinese characters well. Therefore, we use a novel approach, smallest enclosing disc [8], to rebuild the light disk first, and then find the central location. Experiments show that it provides higher accuracy and stability than the former method in [7].

As shown in Fig. 5, we can calculate the distance from every irradiated sensor to the center based on the intensity-distance mapping mentioned in Section II-A. Assuming that the radius of the disk is 30 cm, we can get the smallest distance from every irradiated sensor to the border of the disk. The area within this distance must be covered by the disk. So, we need to find the smallest enclosing disc to cover all these areas. To simplify the calculation, we use four nodes to approximately replace a circle, e.g., the circle with center $E$ can be replaced by a point set $\{E_0, E_1, E_2, E_3\}$. To increase the accuracy, we can use eight nodes or more to replace a circle. So we can use MINIDISC algorithm [8] to find the smallest enclosing disc to cover all these points and then get the central location.

Additionally, according to the lower bound on the number of intersection points covered by a disk [9], there are at least $k$ sensors covered by a disk of radius $r$ centered at an arbitrary point:

$$k \geq \frac{\pi (r - \frac{l}{\sqrt{2}})^2}{l^2},$$

(1)

where $l$ is the side-length of grid cell. Therefore, $l = 20$ cm and $r = 30$ cm ensures at least 2 sensors can be covered by the disk.

C. Draw lines with different widths

To imitate writing with a Chinese writing brush, users can draw lines with different widths. In our design, when the flash light moves fast, the system displays a thin line; otherwise, it displays a thick line.

This drawing process has three steps:
1) Calculate the radius of current circle.
2) Use a trapezoid to connect previous circle with current circle.
3) Fill current circle and the trapezoid with black.

Fig. 2 shows an unfilled line. $C_0$, $C_1$, and $C_2$ are the central locations of moving light disk in three successive samples. We draw the beginning circle with a center at $C_0$ with a default radius, and the following circle with a center at $C_1$.

Fig. 6. Hash mapping from sensor ID to sequence number

Fig. 7. Sketch of a unfilled line

Fig. 3. Distance mapping for the flashlight

Fig. 4. Illuminance on unit area $dS$

Fig. 5. Location calculation
with a radius $r_1$ which increases in direct proportion to $|C_0C_1|$. Similarly, the radius $r_i$ of $C_i$ increases or decreases in direct proportion to $|C_{i-1}C_i|$. Then we use a trapezoid to connect neighboring circles. Taking the circle with a center at $C_0$ and the circle with a center at $C_1$ for example, we use the trapezoid $ABMN$ to connect them, where $C_0C_1 \perp AB$ and $C_0C_1 \perp MN$. At last, we fill all the circles and trapezoids with black color. Fig. 7 shows the sketch of a unfilled line.

D. Real-time Updating

Real-time updating is crucial to give a good user’s experience of writing. To do so, on the one hand, time synchronization between the sink and irradiated sensors is indispensable; on the other hand, frequent sampling and data transmission are inevitable. In addition, we need to handle interference among different sensors.

In [7], the sink broadcasts a “start sampling” packet at the beginning of every round for updating. On receiving the packet, sensors start to sample so as to estimate whether they are covered by the flashlight disk. We ensure that at most 9 sensors in a $3 \times 3$ grid will be irradiated and we adopt a TDMA-like method to avoid interference by using a hash function. This hash function maps every node ID in a $3 \times 3$ grid to a unique sequence number corresponding to TDMA time slot ranging from 0 to 8, as shown in Fig. 6. We use $i$ and $j$ to indicate sensor’s row number and column number respectively. According to our sensor deployment, a node $N_{ID}$ in row $i = \lfloor ID/5 \rfloor$ and column $j = ID \mod 5$. We define its sequence number as $sn = (3 \times i + j \mod 3) \mod 9$ which means a node with sequence number $sn$ will send the packet at the $sn$-th time slot. In Fig. 6, bold numbers stand for nodes’ IDs and italic numbers represent their sequence numbers. In the worst case, the flashlight disk, shown as a shadowed disk in the figure, may be detected by 9 nodes. However, by using our time slot method, interference can be avoided at the beginning.

Based on the strategies in [7], we further improve our system in three aspects:

1) In initialization phase, every sensor can get the environmental light intensity so as to send data to the sink only when sensed intensity is higher than the environmental intensity. It makes our system works in dim light as well as in complete darkness.

2) It is observed that the minimum sampling duration of TelosB Mote required for stable output is about 2 ms. We prolong the sampling duration to 5 ms for reliable sensing. For higher accuracy, we spend 10 ms to calculate the average of two samples.

3) We set one round time at the sink to 150 ms which is long enough to finish 1 broadcasting by sink to sensors (13 ms), 2 sampling by sensors (2 × 5 ms) and 9 data reporting by sensors to sink (9 × 13 ms)

III. Demonstration

We use a $5 \times 5$ sensor grid to track flashlight, which is shown in Fig. 1. We require players to use our LED flashlight in front of the vertical grid board at a distance of 1 m or so. Fig. 8 shows the flashlight trace when we write 5 Chinese characters which mean “welcome to Beijing”.

IV. Conclusion

We demonstrate how to use iBrush to write Chinese character with a flashlight and a sensor grid. In future, we plan to improve the shape at the end of a stroke so that the characters look more beautiful. Additionally, we are considering how to create colorful paintings with flashlights of different colors.

ACKNOWLEDGMENT

The work is partly supported by China NSF grants (60673154, 60721002, 60825205), Jiangsu High-Tech Research Project of China (BG2007039), and China 973 project (2006CB303004).

REFERENCES