Exploring a Novel Inexpensive Tangible Interface for Non-Visual Math and Science

R. Stanton, E. Pontelli, Z. Toups, M. Manshad

Department of Computer Science, New Mexico State University
rstanton,epontell,z@cs.nmsu.edu

Abstract. Tangible interaction enables *physical* manipulation of digital data, making it ideal to support visually impaired students. Visually impaired students are frequently integrated in mainstream courses, collaborating with sighted peers and instructors. This paper describes a *tangible block localization and tracking* method, using small inexpensive sensor packages that detect color placed on an interaction surface—i.e., a standard flat-screen display. The system recursively subdivides the display surface into regions of distinct colors that the sensor package can distinguish. Once located, the sensor package can be tracked by moving the color pattern underneath to follow it, re-expanding the pattern as needed to capture the sensor package if it moves too fast. The novel tracking infrastructure supports novel approaches to teach a number of science and math concepts to visually impaired students.

Keywords: Assistive technologies, tangible interfaces

1 Introduction

Tangible interfaces enable direct manipulation of digital data [3], without necessarily relying on a visual display, a characteristic that offers exciting interaction modalities for individuals with visual disabilities. *Non*-digital tangible manipulatives (e.g., blocks, pins, bands) are commonly used in the early education stages of visually impaired children, especially in math and science. Visually impaired individuals can explore and learn without overloading the communication channels that are more commonly employed by existing assistive technologies and/or instructors (e.g., audio). Owing to their typical design, that of an open space with tangible objects, multiple users can simultaneously manipulate an information space, enabling forms of collaborative learning.

The development of *digital tangible interfaces* has been widely explored in the research literature. Nevertheless, the unique set of circumstances that arise in the *education of visually impaired students*, especially students located at remote sites or students that lack mobility pose new challenges in the development of effective digital tangible solutions, especially in the context of *tracking technologies*, which must first localize a tangible before they can track it. The practical needs and requirements of such students are at odds with prior localization technologies.

We develop a new localization algorithm for simple hardware that can be embedded in tangible blocks, enabling visually impaired and sighted users to work in the same information space, supporting learning, and accounting for the unique constraints of working with distributed students. The localization technique uses a simple and inexpensive color sensor, embedded in a tangible block, to identify its location when placed on top of a display surface (e.g., computer screen), which is normally laid flat on a table. The algorithm locates the sensor package on the screen by rapidly subdividing the screen space into easily detectable colors using a quadtree partitioning [5] pattern. Once located, the sensor's movement can be tracked by displaying a color pattern masked to be under the tangible (where the user cannot see it). By using only a screen and a computer, the system is portable; the screen also enables sighted users (e.g., instructors) to easily work with visually impaired students.

Our long-term project, *Tangible Interactive Multimodal Manipulatives (TIMMs),* is a framework aimed at providing visually impaired students with a tangible "window" into the information normally provided by visual interfaces. We expect to ultimately build a tangible interface comprised of small, block-like tokens that are used in combination with a display surface designed to facilitate interaction and collaboration. The present research aims at developing a component of the project, its localization technology, which must be amenable to the unique needs and requirements outlined below. Future work will develop tangible tracking based on the localization technology, and build educational software on top of the tangible tracking system.

2 Motivation and Needs

The present research is developed specifically for visually impaired learners, potentially situated at remote locations, tackling graphical mathematical content. This imposes a number of constraints on the design of a tangible educational platform, and renders a number of prior approaches unsuitable.

2.1 Classroom Technologies.

Classrooms are enabled by a number of traditional technologies that do not directly support visually impaired learners: pen/pencil, personal computers, projectors, etc. As an ongoing working example, we consider that, in mathematics, one of the most important topics involves graphing functions and building diagrams, something to which traditional technologies and visual interfaces are well-suited. An instructor can easily show sighted students a graph and how it transforms in response to changes in the original function. Visually impaired students receive very little to no benefit from this teaching tool, however, and there are very few alternatives that can convey the same information.

Manipulatives support visually impaired students' learning. During the construction of a graph, a visually impaired student typically locates two points on a corkboard, inserts a pin at each point, and then wraps a rubber band around them to construct a tactile line. These pins fall off if not placed correctly and carefully. If a pin is removed by mistake, the rubber bands can also fall off causing the loss of the representation. Because the representation is static and difficult to change, students find it difficult to explore manipulations. In typical practice, students are allowed up to *three days to construct a graph* during an exam; often students choose to skip that part of the exam.

In highly populated areas, the concentration of visually impaired students is sufficient to sustain specialized schools for visual disabilities. Instead, rural students must be accommodated in standard public schools, where visually impaired learners work with sighted learners. The small number of qualified special education teachers creates a challenge in regions where the visually impaired student population is sparse and small. Thus, there is a challenge in ensuring that each student is served by a trained special educator, forcing special educators to serve a distributed audience of students.

2.2 Needs and Requirements.

Based on these considerations, we explore the unique needs and requirements that drive the design of a localization system for tangibles for visually impaired students. First, a TANGIBLE MEDIUM is required, enabling visually impaired students to manipulate information with physical objects, rather than visual interfaces. A TANGIBLE MEDIUM would augment and extend the existing classroom technologies, building on known practice. In our ongoing example of graph construction, pairs of tangibles can serve to represent any straight line without the need for non-digital manipulatives. A TANGIBLE MEDIUM reduces reliance on the audio channel, which is frequently already overloaded by other assistive technologies (e.g., screen readers) and used by instructors. Second, there is a need for SIMULTANEOUS SUPPORT of visually impaired and sighted users working in collaboration. Due to the extant state of education and its distributed nature, visually impaired students must frequently work alongside sighted students. In addition, educators are normally sighted. Thus, the TANGIBLE MEDIUM also needs to support sighted users, optimizing their senses while enabling collaboration. Since the system will be in use by multiple users, a large display and interaction surface is ideal. Finally, any developed system needs to be PORTABLE, since many educators serve a few students at multiple distributed schools. This means that any system that relies on large pieces of equipment or extensive setup is, by default, unusable in the present context. Since there is little control over the education environment, localization must be robust for a variety of lighting conditions, which eliminates many camera-based solutions.

2.3 Related Work.

There is an extensive body of work concerning tangible interfaces, though the majority of them are not focused on the needs of individuals with visual impairments. Some of the set ups proposed in the literature are either build on expensive hardware or require continuous camera tracking, which conflicts with the specific needs of students who desire a continuous contact with the device (thus causing occlusion). The closest proposal presented is Leigh et al.'s Tangible, Handheld, and Augmented Window (THAW) [4], which uses a color pattern on a monitor that is interpreted by a smartphone camera. Developed in parallel with our own system, THAW can create interactions between a computer screen and the phone screen. Cuypers et al. [2] proposed a similar technique to determine the position and orientation of a smartphone on an interactive surface. We take the tracking methods utilized in this category as inspiration for our own work, which we attempt to simplify for use with simple sensors.

3 Localization System and Algorithm

Our tangible-locating system is composed of a set of sensor packages (i.e., tangibles) and an associated display surface. Communicating algorithms operate on the packages and surface to locate the packages. When integrated into a tangible block, the system supports the needs and requirements of the present context. The algorithm is executed manually and assumes that the sensor packages are actively awaiting messages before the program is run (future iterations will allow dynamically adding packages). When activated, the display surface algorithm subdivides the screen space using novemtree partitioning, in nine colors (Figure 2). As the sensor detects and reports a color, the algorithm further subdivides the region containing color that was identified, making it possible to clear unused portions of the screen for information display.

3.1 Hardware Description.

The present tangible-locating system is based on a simple sensor that reads the color of light combined with a flat monitor for display. A sensor package consists of an Arduino Fio with an RGB Sensor [1], XBee radio, and LiPo battery. The sensor packages are embedded in 3D-printed tangible blocks with the sensor facing down(Figure 1). As a display surface, we used a 40-inch Samsung television at 1920×1080 resolution ($\sim$55 DPI). A computer executes the display surface algorithm.

Fig. 1. A tangible with sensor package embedded in the 3D-printed enclosure

The sensor package runs a simple program waiting for color data requests. On request, it will take RGB readings and send them back. A calibration phase (described in the next section) is necessary if the display surface or its brightness is changed.

3.2 Display Surface Algorithm.

The display surface algorithm, running on the connected computer, uses readings from the sensor packages to direct partitioning of the display and localize the sensor, using the following phases: (1) Run calibration and record baseline RGB values for each color; (2) Divide the display into 3×3 grid and fill with the distinct colors; (3) When triggered, repeat until the specified number of iterations (n, described below) have been

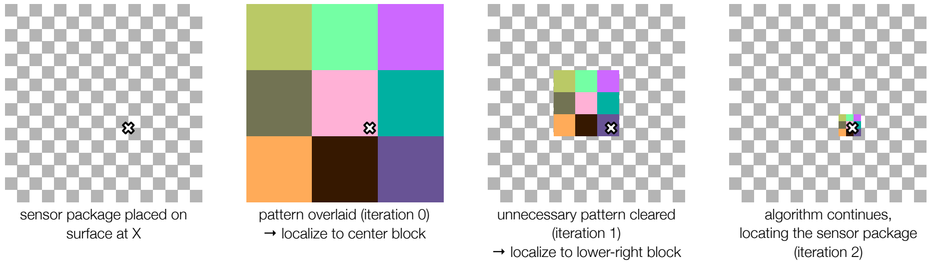

Fig. 2. The sensor package is located on the surface at the black-and-white × . In each iteration, 8/9 of the screen is made available for displaying information

completed: **(a)** Get RGB values from all sensors; **(b)** Identify the colors under each; **(c)** Re-partition the identified rectangles at 1/9th the previous scale; (4) Return upper-left x, y pixel values of identified rectangles.

The algorithm requires a fixed number of iterations (n) be specified in advance. A higher number of iterations (n) will ideally locate the sensor within a smaller area: $screen\_width/3^n \times screen\_height/3^n$. For resolutions up to 2560×1440 on a 40-inch display (DPI = 73.43), the algorithm can reduce the search space to less than a 1 cm × 1 cm square in 4 iterations. However performing more iterations increases the likelihood of a partition edge or intersection ending up beneath the sensor, which could be mis-identified. Because the sensor package's reporting is dependent on characteristics of the display surface's color gamut and its brightness, it is necessary to calibrate the system when these change. During calibration, the user places one or more sensor packages on the display. The display then flashes each of the nine colors in sequence, and the value of each RGB channel is then stored. The color from future readings is identified by finding the most similar of the calibration readings.

4 Tracking Algorithm

Once localized, a sensor package may be actively tracked above the display surface. This tracking method utilizes the same sensor package configurations described previously. The display surface utilizes a square image of fixed size divided into a 3×3 grid of different, distinct colors. After localization, the tracking pattern is rendered underneath the tangible. From this point, the program continually requests updates from the color sensor. By matching each reading to the nearest of the collected calibration readings, it is possible to infer whether the sensor has been moved and in which direction.

A concern related to the frequency of messages is the choice of colors used in the 3×3 grid. For obvious reasons, simple color-matching schemes used with similar colors will often produce erroneous tracking results, potentially causing the tracking pattern to move away from the actual tangible. For our prototype we used a set of colors generated by the SciencesPo mèdialab tool, i want hue [6], using the full range of colors. This set was chosen from 8 potential sets of nine colors, each generated by i want hue. Then, an empty square of the same dimensions as the tracking pattern was rendered on the screen. The sensor was placed somewhere in this square, after which we had the system flash the pattern for 100 milliseconds and collected a reading from the sensor to determine if the sensor had identified the correct color. This was repeated approximately 200 times for each color set. The set we ultimately chose identified the correct color 89.86% of the time.

5 Performance

We empirically determined the accuracy and time-to-fix for our localization system, as well as evaluated its tracking speed. For both sets of tests, we used a 40-inch LED television running as second monitor at 1920×1080 (∼55 PPI) resolution. For the localization accuracy tests, we reduced the testing area to 1917×999 to avoid interference from the operating system UI or other interface widgets; the tracking speed tests were run over the full area of the screen. With 3 iterations of the localization algorithm, the display is subdivided into a 27×27 grid, this resulted in the sensor being located to a 71×37 pixel rectangle (32mm $\times$ 17mm). When we discuss Manhattan distance errors, it is measured in these rectangles.

We calculate the time to fix the position based on the timings determined in developing our code and considering the expectations by a human user. The total time to update the localization display, poll all sensors for readings, and identify the corresponding color/direction is ∼200 ms. Since we perform$n = 3$ readings, and the first reading does not require a pause, each fix cycle required approximately 600 ms.

The present localization system clamps all results to the measured rectangle. To assess accuracy of the localization algorithm, we determine two accuracy values to report: **(1)** the percentage of results in the correct rectangle and **(2)** the distance of the measured points to the ground truth in terms of the number of rectangles away the result is. As seen in Figure 3, most localization tests were correct or only off by one. To collect data points, the software was configured to locate tangibles in response to a mouse click. For the accuracy tests, we performed the following repeatedly: **(1)**At the time of the first mouse click, the x,y coordinates of the mouse cursor are recorded; **(2)**then, the sensor package is centered over the same location as the cursor, and the cursor is moved away; **(3)**on the next mouse click event, the localization system is run and the position of the identified section of screen area is recorded. One drawback of this testing methodology is the introduction of random human error. We expect this to introduce minor random error into the resulting data: a Manhattan distance of 1–2.

For the analysis, 150 readings were taken. These were analyzed by Manhattan distance between the identified rectangle and the correct one; it was noted that for Manhattan distances 4 and above (10/150 readings), the number of readings dropped to 1–2, suggesting they were outliers. This set of outliers were removed, producing 137 usable readings. Overall, the results are positive, 72 of the 137 readings were correct (52.6%), 32 (23.4%) were one rectangle off (Manhattan distance of 1), and 8 were off by two (5.8%); in combination, these account for 112 of 137 readings (81.8%) (Figure 3). Given the size of the rectangles and the size of the tangible, these readings are all potentially valid, that is, the tangible would be located over the rectangle recognized.

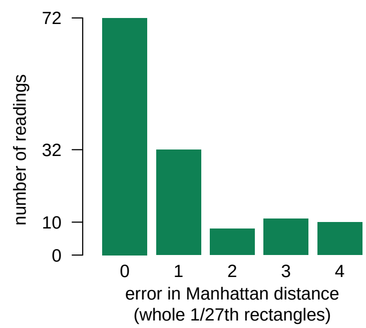

Fig. 3. Histogram of number of readings versus error

We analyze the tracking method in terms of the speeds at which it can track sensor packages above our display, and in terms of the system reliability. An initial keypress prompts the system to record the starting location. Immediately following this we move the tangible in a straight line. After covering some distance over the surface of the screen, we trigger the end of the recording via another keypress and the speed is calculated. If the system is unable to keep the tracking pattern beneath the sensor, we flag this by pressing another key and immediately ending the recording. The number of times such tracking failures occur is recorded.

By collecting the information above, we are able to approximate the speed at which the sensor packages can be reliably tracked moving in a single direction and see how reliable the tracking method is. For each recording, the speed at which sensor packages were moved was adjusted based on observing whether or not the sensor was lost in the prior reading; if the sensor was not lost, we move it slightly faster during the next recording. Sensor packages were moved in different directions for each recording, but without any change in direction for that recording's duration.

We collect two sets of results as described above. The first set consists of 204 recordings using a single sensor package above the display surface. The second set is recorded based on two sensors moved above the display surface concurrently, with 228 recordings for each sensor package.

Of all the recordings collected with a single sensor, 176 were not flagged as unreliable, or 86%. The reliable sensor readings were moved an average of 18.17 cm across the display surface. From these, the highest speed was 5.63 cm/second and the average was 3.02 cm/second. From the readings flagged as unreliable, they were moved an average of 4.24 cm before the tracking pattern was lost.

The proportion of reliable readings to unreliable for the concurrent sensors was significantly different from that of the single sensor recordings—approximately 66.8%. When taking these reliable recordings, the concurrent sensors were moved an average of 12.83 cm. The highest speed of reliable recordings was 7.07 cm/second. The average speeds for the sensors was 2.81 cm/s.

From these results we can see that, in the current state of the tracking method, achieving similar reliability with different numbers of sensors requires moving the sensors at a slower speed due to the greater potential for mis-identifying colors or otherwise losing the location of one of the sensor packages.

6 Discussion and Conclusions

We argue for the suitability of our system within the context of visually impaired student education based on the needs and requirements derived from observations on the state of education for such students. The combination of a collection of sensor packages with any display screen that could be laid flat comprises our tangible medium. Because such a display can be of any size and multiple tangibles can be localized above it, we likewise establish a collaborative interaction space. The system uses the display screen, briefly, during localization. The system is portable, making it ideal for itinerant educators.

The most obvious point of comparison for our novemtree partitioning method of localization is Leigh et al.'s THAW method [4]. Using the camera of an Apple iPhone, Leigh et al. were able to locate the phone's position above the screen to within an average of 6.33 mm. Using a cheaper hardware setup on a larger screen, we were able to locate a sensor package to a 33 mm $\times$ 17 mm rectangle using 3 readings per point.

We have described a tangible-tracking technology designed to support visually impaired students collaborating with sighted instructors and peers. The validated algorithm we describe will be of use to others with similar needs. The present locating technology is sufficient to accomplish math teaching for visually impaired students. Near future work develops applications to teach linear equations, as in our earlier example.

Using the novemtree algorithm and hardware within 3D-printed tangible blocks, we hope to provide a less cumbersome, more accessible input modality for visually impaired learners. The present research is one step in a larger research agenda to support visually impaired student learning. Future versions of the system may further support students by offering tangibles that can move themselves on the surface or extendable cables that can be stretched between tangible blocks.

References

1. Avago Technologies. Apds-9960 digital proximity, ambient light, rgb and gesture sensor. Data Sheet AV02-4191EN, 2013.
2. Tom Cuypers, Yannick Francken, Cedric Vanaken, Frank Van Reeth, and Philippe Bekaert. Smartphone localization on interactive surfaces using the built-in camera. In *Proc. Procams*, volume 2, pages 3–4. Citeseer, 2009.
3. Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 234–241, New York, NY, USA, 1997. ACM.
4. Sang-won Leigh, Philipp Schoessler, Felix Heibeck, Pattie Maes, and Hiroshi Ishii. Thaw: Tangible interaction with see-through augmentation for smartphones on computer screens. In *Proc. of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '15, pages 89–96, New York, NY, USA, 2015. ACM.
5. Hanan Samet. The quadtree and related hierarchical data structures. *ACM Comput. Surv.*, 16(2):187–260, June 1984.
6. SciencesPo MediaLab. i want hue: Colors for data scientists. `http://tools.medialab.sciences-po.fr/iwanthue`, 2017. Accessed: 2017-01-20.