Injecting Life into Toys

Songchun Fan
Duke University
Durham, USA

Hyojeong Shin
Duke University
Durham, USA

Romit Roy Choudhury
University of Illinois
Urbana-Champaign, USA

ABSTRACT

This paper envisions a future in which smartphones can be inserted into toys, such as a teddy bear, to make them interactive to children. Our idea is to leverage the smartphones' sensors to sense children's gestures, cues, and reactions, and interact back through acoustics, vibration, and when possible, the smartphone display. This paper is an attempt to explore this vision, ponder on applications, and take the first steps towards addressing some of the challenges. Our limited measurements from actual kids indicate that each child is quite unique in his/her "gesture vocabulary", motivating the need for personalized models. To learn these models, we employ signal processing-based approaches that first identify the presence of a gesture in a phone's sensor stream, and then learn its patterns for reliable classification. Our approach does not require manual supervision (i.e., the child is not asked to make any specific gesture); the phone detects and learns through observation and feedback. Our prototype, while far from a complete system, exhibits promise – we now believe that an unsupervised sensing approach can enable new kinds of child-toy interactions.

1. INTRODUCTION

We imagine a future in which toys interact and evolve with children, while remaining as inexpensive as passive toys. Our key idea is to insert a smartphone into the toy, say a teddy bear, such that the camera of the phone aligns with the eyes of the bear and the microphone and speakers are near the bear's mouth. With such a set-up, the toy has the opportunity to see, hear, and motion-sense the child's interactions, and respond back in a way that improves engagement with the toy. As simple examples, a teddy bear could say "*ouch*" when the child pulls it by an ear, could sing a *lullaby* until the child falls asleep, or could even *vibrate* when the child shakes hands with it. With connections to the cloud, toys could be brought into a network to allow for collaborative learning, ultimately leading to a new ecosystem for toy-related apps and services.

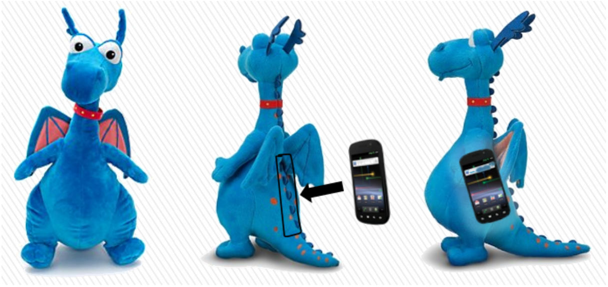

Figure 1: Smartphone inserted into toys, enabling them to sense and interact with children.

This vision is not fundamentally new, but perhaps a logical next step to the current trends in the toy industry. Today's toys have progressed from "passive" to "pre-programmed" objects, but their set of interactions are static across time and individuals. However, since each child is different in her taste and behavior which evolve with age, future toys could evolve as well. Given that mobile computing research is making rapid advances in multi-modal sensing, activity/gesture recognition, emotion analysis, etc., it seems viable that toys can be augmented with such capabilities. A smartphone inserted in the toy (Figure 1) should be able to train itself in an unsupervised manner and respond meaningfully when the child is playing with it. In longer time scales, the toy could download upgrades to itself from the cloud, and perhaps periodically re-calibrate its models to stay in sync with the child's growth.

This project, named *Buzz*, is a long-term research commitment focussed around the problem of autonomously learning a child's *gesture vocabulary*. By gesture vocabulary, we mean the set of gestures that a child would naturally perform *on the toy* while playing with it. Example gestures could be hugging the toy, shaking hands, patting it, swinging it, etc. Our broader goal is to learn the appropriate responses to a child's gestures, such that these responses from the toys can indulge the child into performing more gestures, ultimately extending the length of each interaction. Understanding this mutual relationship between gestures and reactions is challenging – during the initial bootstrap phase, neither the toy nor the child knows their counterparts' preferences and capabilities. Yet, the toy needs to gradually learn and converge on the mapping between responses and gestures. This paper does not pursue this broader challenge, but concentrates only on the first step of recognizing the child's natural gesture patterns. Recognizing this pattern library can itself be valuable to a range of applications, including educational toys, early-development monitoring, or guided entertainment.

Understanding the child's gesture consists of two sub-tasks: (1)

detecting the presence of a gesture in the sensor stream, and (2) *clustering* similar gesture signals for the purpose of classification. The problem of detecting presence is non-trivial because the child's gesture vocabulary is not known *a priori*, and hence, the toy cannot be trained on a set of pre-defined gestures. In an ideal case, the child's natural gestures need to be extracted from a continuous sensor stream, polluted by noise and other non-gestures (such as the child flicking the toy while walking past it). Figure 2 shows snippets of a real sensor stream – the deliberate gestures are difficult to tell from the inadvertent ones.

The second problem of clustering gesture signals arises from the fact that repetitions of the same gesture from a given child produce wide variance in their sensor data. This is an outcome of weak muscle memory and control in toddlers, which prevents them from faithfully reproducing an action. As a result, even if Buzz has recognized all the gesture segments present in a sensor stream, it still needs to create buckets of somewhat similar segments. That way, if the child hugged the toy at 3 different times, the segments corresponding to them will all be in the same bucket (or class). Given that these segments from the same gesture can be quite dissimilar, and even more dissimilar across children, global thresholding on similarity can lead to heavy false positives/negatives. Careful design is necessary, especially in view of the unsupervised nature of the problem.

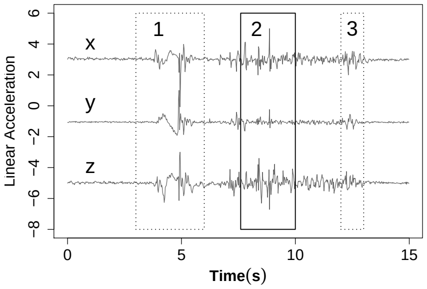

Figure 2: Difficult to identify gesture segments. (The dotted boxes are non-gestures while the solid box is an actual gesture.)

Our inspiration for solving these problems emerged fortuitously from observing children in a daycare facility. Briefly, we ran simple measurements and passively collected sensor data from toys equipped with smartphones. This data proved to be very sparse since the children played with our toys for short durations and moved to other toys. To make the toys more engaging, we repeated another round of experiments, but this time, the toy made random funny sounds whenever there was any activity recorded on the motion sensors. Surprisingly, we noticed that when a child made a deliberate gesture to the toy, the funny sound often prompted the child to repeat the gesture. However, if the motion was caused by an inadvertent action, it was not repeated. Subsequent experiments (designed to further verify this observation) consistently produced encouraging results, giving us a valuable handle on the overall problem.

With the gestures detected, Buzz employs the *dynamic time warping* algorithm (DTW) to compute (dis)similarities between all gesture pairs. DTW is particularly applicable here since it allows for expanding and contracting a signal in time. Therefore, even if a child executes the same gesture differently, DTW accommodates some of these differences, resulting in reliable characterization. Then, Buzz applies hierarchical clustering on all pair-wise dissimilarities to create the gesture classes. The use of thresholds is avoided to remain adaptive to different children and their varying gesture patterns.

This paper incorporates the above ideas into a Android based prototype (Nexus Galaxy phones inserted into 4 different soft toys) and experiments with 2 different kids between the ages of 1.5 and 3 years. As part of the ground truth collection, the child's actions are video recorded and the timing and nature of each gesture is manually noted offline. As performance metrics, we report the precision and recall across 38 gestures made in 8 experiment sessions, and observe the marked improvements in "interaction time" with and without Buzz. Results are encouraging in our opinion, with the interaction time increasing up to 3 times when running Buzz.

2. NATURAL QUESTIONS

(1) Why use smartphones? Removing the smartphone to receive phone calls or other activities can be inconvenient. True, there is no technical reason for using a smartphone – the sensing capabilities could be achieved using any specialized hardware, such as embedded sensors or integrated chips that are dedicated to toys. However, given the ubiquity of smartphones, such hardware may appear costly/unnecessary. More importantly, a smartphone can enable new functionalities (e.g., voice recognition, laughter/cry detection, social interaction) simply through software updates, requiring no hardware change. The interface of smartphones allows parents to access these features without buying new toys. In conversation with some toy makers and venture capitalists, we were also suggested that old smartphones may be better candidates since they can permanently remain inside the toy (except for recharging). While these are relevant issues for the broader success of the vision, this paper focusses on the technical aspects alone.

(2) What defines the long-term success of the project? What is the final metric for the system? We still lack clarity on the final metrics for this project; it is quite possible that the metrics would emerge from the application of interest. In this paper, we are attempting to build general primitives for unsupervised gesture recognition, and defining the success narrowly as detection accuracy (precision and recall). However, many questions remain unexplored. For instance, what is the complete gesture vocabulary for a child? Do these toy–directed gestures reveal information about the child's behavior, growth patterns? Can some toy-responses nudge the child into behavior modifications that are otherwise difficult for parents? Is there valuable information to be gleaned from a camera capable of continuously looking at a child from a close-up? This paper is truly a preliminary step in this direction.

3. Buzz: A CORE DESIGN DECISION

When initiating the project, we decided to take up a data-driven approach. Thus, as the very first step, we developed a simple sensing module for Galaxy Nexus phones that collects the accelerometer and gyroscope readings at the highest frequency. We purchased a number of soft toys that already had back pockets in them (using velcros), and inserted the phones in the toys. We invited two children (of age 17 and 30 months) to play with the toys and video taped the sessions for obtaining ground truth[1]. The following observations – from the sensed data as well as

[1] This research is approved by Duke Institute Review Board (IRB No.B0628).

from watching the video – influenced our design decisions.

(1) Low gesture density. Upon offering the toys to the children, we immediately realized that meaningful gestures occur infrequently in time. In 8 sessions, we registered 2 gestures per minute on average, primarily because each child played with his/her toy for short durations, and came back to it several hours later. Larger experiments in a daycare facility (performed later) reinforced this observation across multiple children.

(2) Extracting gesture signals. To extract meaningful signals from a data stream, one common technique is to compute the energy of the signal on a moving time window and select the windows for which the energy is above a threshold. Unfortunately in our observation, many gestures exhibited weak energy footprints, while some non-gestures showed high energy. Figure 3 shows an example of mixed gestures and non-gestures. An accidental "Drop" presents an energy higher than intentional gestures "Grab" and "Shake".

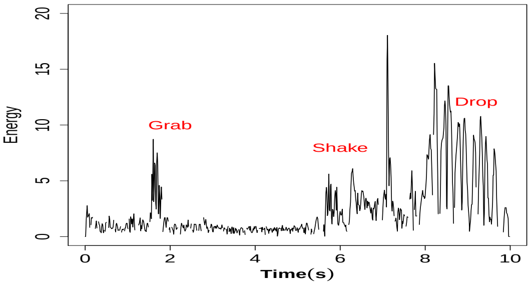

Figure 3: Non-gestures can be stronger than gestures.

(3) Start and end markers. Even if the presence of a gesture is detected, it is important to mark the start and end points of the signal segment that contains the gesture. Poor markers can cause a given gesture to match incorrectly with other gestures, ultimately affecting classification. Signal segmentation with zero background knowledge continues to be a tricky problem, and has been the point of research in the signal processing and activity recognition community [9, 2]. With noisy sensors on smartphones, the problem is more pronounced.

The above observations offered us two useful guidelines in designing Buzz. (1) Training a toy via passive sensing would be highly time consuming and perhaps impractical as a real system; data collection has to be sped up appreciably. (2) There has to be an out-of-band way – some form of supervision – to detect the presence of a deliberate gesture. An obvious approach is directly instructing the child to perform a few gestures and learning on them. However, that defeats our core purpose of learning the natural gestures of the child when she plays with the toy.

The Idea of Reactive Sensing

While we struggled to meet the two design guidelines above, we continued to closely watch the behavior in kids while they played with different types of toys. And over time we began noticing a pattern. *We observed that when a toy responds to a child's action (like a talking toy that laughs when the child presses its hand), the child tends to repeat her action (to make the toy laugh again).* This served as a valuable inspiration, and actually forms the basis of our design framework in this paper. (Later we also found scientific articles [1] that reported the same observation.) We immediately modified our smartphone to generate a random sound whenever it suspected a gesture from the child. We then conducted another round of experiments with the same two kids, and this time they often repeated their gestures intentionally, in order to hear more sounds. Observe that this single, trivial modification now met both the design guidelines. Due to repetitions of the gesture, we obtained data points in higher density, while still not forcing the child into any pre-defined gesture. Further, we exactly knew the presence of a gesture, since a non-gesture (such as moving the toy aside or stepping on the toy) would not get repeated after the sound. The following section builds around this core observation, and employs some techniques from machine learning and signal processing to suitably process the gestures.

4. DESIGN SKETCH

Figure 4 shows a functional overview of Buzz. The system is divided into two main modules – the *front end*, responsible for quickly detecting potential gesture segments and generating acoustic responses, and the *back end*, tasked with recognizing and classifying the actual gestures, and training the toy on them. In our current prototype, the classification is not real-time, meaning that we evaluate the performance of gesture detection offline. A fuller system would need to convey the recognized gesture patterns back to the front end, so the front-end can present the acoustic feedback in real-time. We leave this to future work.

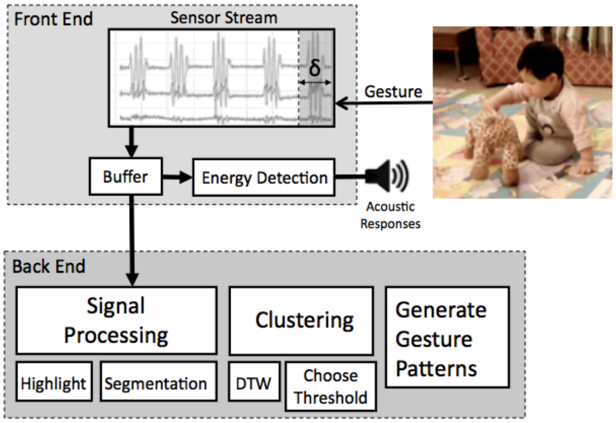

Figure 4: System sketch: The front end generates acoustic feedback based on energy detection. The backend processes the sensor data to crop out signal segments and trains on them.

4.1 Front End

To be responsive to (potential) gestures in real time, we compute the total energy of the signal within a sliding window, and if this value is above a certain threshold, an acoustic response will be played. Three parameter choices are of interest: window size, value of threshold, and the kind of acoustic responses that the toy should generate. We discuss each of them below.

Choosing an incorrect window size has important ramifications. A small window may cause the toy to react to the gesture too early, interrupting the child while she is in the middle of a gesture. Moreover, the energy in that window may not exceed the threshold, and hence, the toy may not respond at all even though its a legitimate gesture. Too long a window, on the other hand, may delay the response much after the gesture, and the child may not relate the gesture to the response at all. Given that human perceptive delay is around 50ms-100ms [8], we set the window size δ at 50ms (=5 samples at 100Hz). Also, from the gathered data, we verified that all the legitimate gestures presented substantial energy when averaged over this time window. We set the energy threshold conservatively to a low value, permitting almost all gestures and non-gestures to elicit an acoustic feedback from the toy. This threshold doesn't distinguish them, but serves as a nudge to the child to repeat her gesture, while leaving the actual classification to the *back end*. Also, frequent acoustic sounds from the toy – henceforth called a "beep" for simplicity – engages the child for longer durations.

While choosing acoustic responses, we noticed that sounds from real life confused the two children in our experiments. For example, at first we attempted to use a sound clip containing the laughter of a child – our rationale was that kids would like hearing voices of other kids. However, when they played with the toy and heard this sound, they got confused perhaps because they expected real children around them. Instead, we experimented with funny cartoon sounds that children don't hear in real life, and observed consistently better results. We are aware that this is somewhat counter-intuitive – one would assume that familiar sounds would be less confusing. Nonetheless, since these children responded better to the cartoon sounds, we decided to continue using them. In a more uncontrolled setting, the choice of feedback sounds may also need to be learned on the fly.

With reactive sensing incorporated into the toys, we performed another round of experiments with the 2 children. Figures 5, 6, and 7 report on the results. Figure 5 zooms into the sensor data from one of the sessions picked randomly, illustrating the higher density of more deliberate gestures with Buzz-enabled toys. Passive sensing in contrast is sparser and interspersed with non-gestures, such as "drop", "grab", etc. While this is a single instance, Figure 6 shows aggregates across 8 different sessions – the gestures are marked manually from the videos. For reactive sensing, it shows the number of gestures that did not get repeated despite a beep (A), the number of gesture pairs repeated before and after a beep (AA), and the number of gesture pairs around a beep that were not similar (AB). Buzz recognized all the gestures in the second bar (AA), far more than passive sensing (which did not use acoustic feedback and hence could pull out only a few gestures with confidence). Finally, Figure 7 shows the total energy recorded on the sensors with passive and reactive sensing. Substantially higher energy with reactive sensing is an indicator of stronger engagement and interaction with the toy, implying higher density of gestures. This is well aligned with what we set out to achieve with reactive sensing.

4.2 Back End

Observe that the front end uses energy detection to trigger an acoustic feedback, or beeps. This implies that the timing of the gestures are likely to be around the timing of the beeps, and hence, the backend can immediately crop out the portions of the signal around each beep. However, these portions contain a lot of false positives, since many non-gestures can also exhibit a high energy footprint. The back end's task is to (1) select portions containing a valid gesture and extract the gesture segment (i.e., start and end time), (2) compute a similarity score between gestures, and finally (3) compute the number of distinct gestures. This is a difficult problem in general, but in this case we have valuable domain knowledge – that beeps prompt children to repeat their gesture.

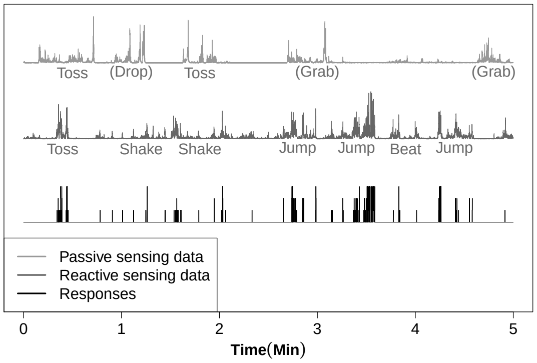

Figure 5: Zoom-in view of the sensor readings – higher gesture density with reactive sensing. Non-gestures shown in braces.

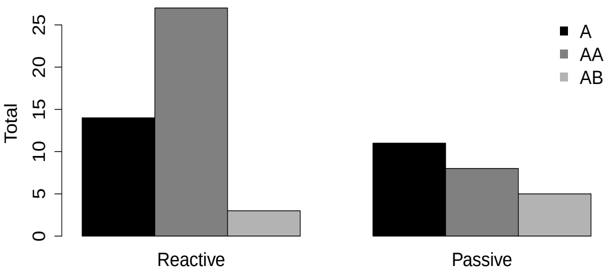

Figure 6: Acoustic feedback, or "beeps", generate pairs of repeated gestures, facilitating recognition.

Segmentation

Figure 8 shows an example signal portion, which contains some sensor signals and the timing of the beeps. Note that the beeps are active as soon as the moving window detects sufficient energy. Now, given the timing of these beeps, the question is: *are the two signal segments, preceding the two beeps, similar?* If so, then we deem them as instances of a valid gesture. To perform this similarity comparison, we first need to extract these two signal segments from the signal stream. As shown in Figure 8, Buzz estimates the duration of each signal segment as (δ + *the duration of the beep*), where δ is the length of the sliding window discussed earlier. The segment starts δ time before the start of the beep and ends at the end of the beep. Since the beep timings are precisely known, Buzz extracts out the signal segments and advances to the next step of computing their similarity.

Gesture Similarity

Computing the similarity between two signal segments, g_i and g_j, can be performed using various techniques from literature. In this context, however, we have prior knowledge that kids may execute the same gesture in slightly different durations, due to immature muscle memory [5]. Since the *dynamic time warping* algorithm (DTW) [3] can compare expanded and contracted versions of signal segments, it fits well for this application. The algorithm searches the best alignments between two time series vectors by trying to minimize the sum of absolute difference between each pair. This method has founded frequent applications in speech recognition, where similar words can be spoken at different speeds.

In our case, the vectors are multivariate, because we have acceleration and rotation, each with 3 dimensions. To include all the information, we combine 6 dimensions of a signal segment into

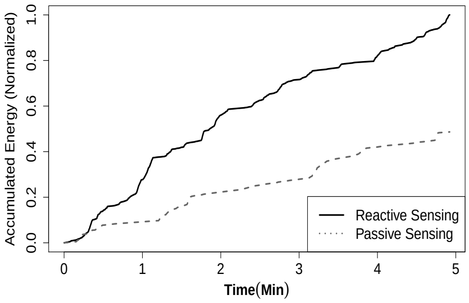

Figure 7: Higher total energy recorded by sensors suggests longer engagement with the toy with reactive sensing.

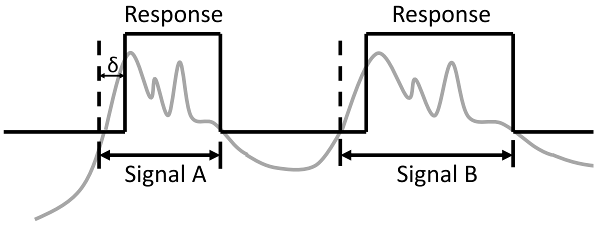

Figure 8: The timeline of sensor fluctuations and the beeps.

one matrix. Each row corresponds to a sample reading, and each column is a sensing dimension. Euclidean distance is used as a measure of (dis)similarity between two elements. Under this set up, DTW calculates the distances between every two matrices after aligning them – this distance is the "dissimilarity" score between two gestures. The output is a dissimilarity matrix where element d_{ij} denotes the distance (or dissimilarity) between signal segment i and j.

Referring back to Figure 8, we now have the dissimilarity score of signal segments A and B. However, we still need to resolve if this dissimilarity is adequately small to deem them as the same gesture. One possibility is to use a threshold – if their dissimilarity is greater than the threshold, then they are not. However, choosing a global threshold may be risky. Instead, we adopt an unsupervised technique – hierarchical clustering – to understand how all the gestures are scattered in the dissimilarity space, and then extract the valid gestures from them. As a result, we obtain an estimate of the number of distinct gestures exhibited by the child.

Distinct Gestures via Hierarchical Clustering

A hierarchical cluster groups its elements based on the distances between them, and presents in a hierarchical tree structure as shown in Figure 9. In this figure, for example, segments 1 and 2 are very similar, with a distance of around 20 on the Y axis. Furthermore, they are more similar to 11 than to 4, and so on. To be able to extract the distinct gestures, we need to cut the tree at some value of dissimilarity – each disconnected sub-tree below that value will be a valid gesture, and all the segments in that sub-tree would be deemed as instances of the same gesture. For instance, if we cut the tree at the height of 45, then 10 and 12 would correspond to the same gesture, and 1, 2, 4, and 11 would be segments of another gesture. However, we need to cut the tree without choosing a global threshold. For this, we design the following heuristic.

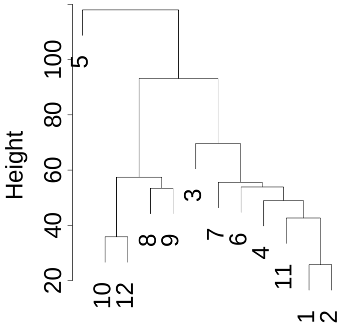

Figure 9: Hierarchical cluster tree with 12 segments.

The lack of ground truth on the actual number of gestures leaves us no choice but to try cutting the tree to every possible number of classes. If we cut out only one big class, we cannot even separate signals from noises. On the other hand if we cut the tree into small classes, the worst case is that every class contains only one signal. Such classes cannot be regarded as gestures, because a gesture should be repeated for at least once. In other words, a gesture class should contain consecutive signals that are similar. We define *signal pair*, which is the combination of two consecutive signal segments. A class is regarded as a *gesture*, if it contains at least one *signal pair*. The key intuition of our heuristic is that, the best cutting should give the largest number of *gestures*, while maximizing the number of *signal pairs*.

Our heuristic traverses the number of classes and picks the one that maximizes both the number of *gestures* and the number of *signal pairs*. Table 1 shows an example. Each row considers increasing number of classes, and the bold numbers denote which *signal pairs* are present in the classes. Observe that for a few classes, one class may contain many signals – so the number of *signal pairs* would be high but the number of *gestures* is low. For many classes, several classes may have only one signal segment each – so now, both numbers of *gestures* and *signal pairs* would be low. Buzz picks the highlighted row, and announces 2 valid *gestures* as a final result – segments 8 and 9 are instances of the same gesture, while segments 1, 2, 3, 4, 6, and 7 are instances of another gesture. Of course, Buzz does not know what these gestures semantically mean (whether its a hug, or a dance, or a hand-shake).

Clus.	Content of Clusters	Gestures	Pairs
1	**everything**	1	6
2	<5> <**everything else**>	1	5
3	<5> <10 12 **8 9**> <**3 7 6 4** 11 **1 2**>	2	4
4	<5> <10 12 **8 9**> <3> <**7 6** 4 11 **1 2**>	2	3
5	<5> <10 12> <**8 9**> <3> <**7 6** 4 11 **1 2**>	2	3
6	<5> <10 12> <**8 9**> <3> <7> <6 4 11 **1 2**>	2	2
7	<5> <10 12> <**8 9**> <3> <7> <6> <4 11 **1 2**>	2	2
8	<5> <10 12> <8> <9> <3> <7> <6> <4 11 **1 2**>	1	1
9	<5> <10 12> <8> <9> <3> <7> <6> <4> <11 **1 2**>	1	1
10	<5> <10 12> <8> <9> <3> <7> <6> <4> <11> <**1 2**>	1	1
11	<5> <10> <12> <8> <9> <3> <7> <6> <4> <11> <**1 2**>	1	1
12	<5> <10> <12> <8> <9> <3> <7> <6> <4> <11> <1> <2>	0	0

Table 1: Classifying distinct gestures without global threshold.

5. PERFORMANCE

We performed the above gesture recognition procedure on the small data set we gathered from two children[2], over a span of two weeks. From the recorded videos of the children, we noted 38 valid gestures (of 6 types), and many noisy non-gestures. We compare the timings of the detected gestures ($G1$, $G2$, $G3$, $G4$) with the true timings of the actual gestures (labeled as jump,

[2] Daycare facilities were not comfortable with video recording children precluding us from larger scale experiments.

shake, beat, etc.). Figure 10 illustrates the overall performance. Gesture G2 classifies "beat" and "toss" as the same gesture and recognizes both of them. Gesture G3 recognizes almost all instances of "hop". Gesture G4 recognizes one of the instances of lift and misses out on the other. However, gesture G1 performs worse classifying "shake" and "jump" to be the same gesture, and incurring a number of false positives. The overall precision and recall from these experiments were 85.1% and 81.6%, respectively. We also observed 2.9x improvement in average interaction time, i.e., the time for which the child played with a toy in a given session. We believe these results are encouraging, although not yet conclusive given the small size of our data set.

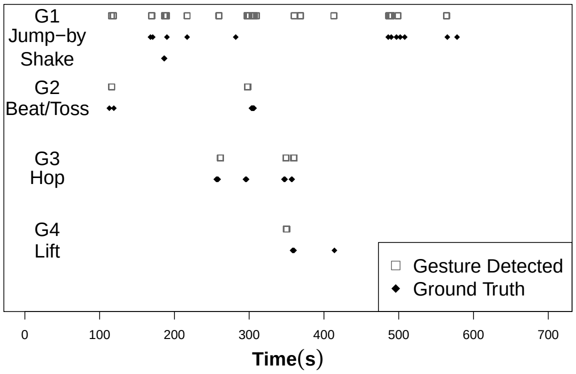

Figure 10: Gesture recognition performance with Buzz

6. MANY LIMITATIONS

Needless to say, this paper is a small step towards the broader vision, and substantial work remains, as discussed here.

Small Data Set. The results from this paper is not meant to be conclusive, rather an indication of viability. We are making an earnest effort to reach to larger bodies of children, even visit each of their homes, and run the experiments under parental supervision. Of course, we are facing IRB and several other logistical hurdles.

Better Algorithms. This paper uses simple techniques for gesture recognition, partly in view of running the algorithms on the phone. It is entirely possible that more sophisticated algorithms (such as Hidden Markov Models) can be executed on the phone itself, or if necessary, on the cloud. The search for the optimal techniques is left to future work.

Energy Consumption. This paper sidesteps the question of energy consumption, however, opportunities exist. Upon detecting that a child is not playing with a toy, almost all sensors can be turned off, except perhaps the least energy-hungry sensor – compass. Further, emerging chips (Qualcomm Snapdragon) are offering continuous sensing capabilities lasting for a week. We believe energy can be addressed in the context of this short/bursty usage pattern.

Useful Non-gestures. We defined non-gestures as actions that were not directed to the toy – examples are, stepping over the toy, moving it away, pulling it in a box full of toys, etc. On second thoughts, perhaps non-gestures could also be used for acoustic responses from the toy. The toy could scream "that hurts" when the child steps on it. While this makes gesture detection perhaps easier, the space of responses now grows larger, making the gesture-response mapping harder.

7. RELATED WORK

The idea of using smartphones to enhance traditional toys is not entirely new. Ubooly is a recently launched toy that allows users to insert an iphone into a small bear, such that the phone screen remains exposed. The screen displays the face of the bear and speaks or produces facial expressions when the child touches the screen. Laugh & Learn Apptivity Case is another new casing for phones, allowing parents to protect their phones while kids can play with them. Notori [4] is a play kit that combines mobile apps with traditional wooden toys. While these toys are beginning to exploit smartphone capabilities, to the best of our knowledge, none attempts to recognize gestures in an unsupervised manner. Adult–facing devices, such as Kinects, Wii's, Nike Fuelband, smart–watches [6] have, on the other hand, concentrated on mature activity recognition. However, these too are built on supervised platforms. In the academic community, recent research has investigated various problems in unsupervised gesture recognition [7]. We adopt these techniques, and customize them to the space of toy-children interactions.

8. CONCLUSION

We explore the possibility of bringing smartphone–based gesture recognition to children's toys. We believe that a new ecosystem could emerge, with new kinds of toys, apps, and even internet of toys. A community of toys and children could emerge, even in real time via the cloud, enabling new kinds of social interactions. While this paper takes only a small step in pursuit of this vision, we hope it conveys the rich prospects underlying the fusion of toys with mobile computing.

9. REFERENCES

[1] Discipline kids with positive and negative consequences. http://discipline.about.com/od/disciplinebasics/a/Discipline-Kids-With-Positive-And-Negative-Consequences.htm.

[2] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.

[3] T. Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software*, 2009.

[4] Y. Katsumoto and M. Inakage. Notori: Reviving a worn-out smartphone by combining traditional wooden toys with mobile apps. In *SIGGRAPH Asia 2013 Emerging Technologies*, SA '13, pages 13:1–13:2, New York, NY, USA, 2013. ACM.

[5] R. Lindert. birth to 2: muscle memory, 03 2012. Copyright Scholastic Inc.

[6] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *BSN '06*, pages 113–116, Washington, DC, USA, 2006. IEEE Computer Society.

[7] D. Patterson, D. Fox, H. Kautz, and M. Philipose. Fine-grained activity recognition by aggregating abstract object usage. In *Wearable Computers, 2005.*, pages 44–51, 2005.

[8] B. Shneiderman. *Designing the user interface: strategies for effective human-computer interaction.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986.

[9] D. Wyatt, M. Philipose, and T. Choudhury. Unsupervised activity recognition using automatically mined common sense. In *AAAI'05*, pages 21–27. AAAI Press, 2005.