In Frontiers of Pen and Touch: Impact of Pen and Touch Technology on Education. Hammond, T., Adler, A., and Prasad, M. (Eds.) New York: Springer, 2017. (Published conference proceedings for CPTTE 2016.)

Chapter 3
Machine Analysis of Array Skip Counting in Elementary Math

Kimberle Koile, Andee Rubin, and Steven Chapman

Abstract The *INK–12: Teaching and Learning Using Interactive Inscriptions in K–12* project has been developing and investigating the use of pen-based technology in elementary math classes. This paper reports progress made on machine analysis of students' visual representations created using a combination of freehand drawing and a digital array tool that supports learning multiplication. The goal of the machine analysis is to provide insights into students' mathematical thinking as revealed through creation and manipulation of visual representations. For array representations, machine analysis involves interpretation of ink annotations that represent problem-solving strategies, one of which is counting by a number other than 1, aka skip counting. A subset of student work from a five-week trial in a third grade class provides a corpus for development and evaluation of the machine analysis routines. This paper describes the routines and presents findings demonstrating that the routines are able to provide accurate information about students' skip-counting strategies. It discusses the key to the accuracy—using knowledge about the structure of arrays and the nature of skip counting to bias the machine analysis routines; and presents evaluation results for two versions of routines that do not use this knowledge and that consequently suffer from high error rates. The paper also discusses current work on extending the routines to analyze the process of creating representations and future work on using the routines on thousands of pieces of student work from the five-week trial.

Kimberle Koile✉
Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts, USA
e-mail: kkoile@mit.edu

Andee Rubin
TERC, Cambridge, Massachusetts, USA, e-mail: andee_rubin@terc.edu

Steven Chapman
stevcode.com, Cambridge, MA, USA, e-mail: stev.code@gmail.com

3.1 Introduction

Visual representations play a key role in mathematics education, helping students to gain insights into mathematical concepts and demonstrate their mathematical thinking, e.g. [3, 19, 2]. The National Council of Teachers of Mathematics (NCTM) in its Principles and Standards for School Mathematics devotes a strand to the importance of representations [13], and the Common Core State Standards for Mathematics, which incorporate the NCTM mathematical process standards, emphasize representations by specifying that students learn to "model with mathematics" [14]. Representations serve several functions for students of mathematics. Visual analogs of quantities and operations provide a way for students to explore mathematical relationships and can give them insight into the structure of the number system and the properties of operations, e.g., commutativity. Representations also serve a communicative function, giving students a non-verbal language for expressing their strategies and knowledge and a means for understanding others people's approaches.

To facilitate the use of mathematical representations, the *INK–12: Teaching and Learning Using Interactive Ink Inscriptions in K-12* project (ink-12.mit.edu) has been developing and testing pen-based digital tools in elementary math, focusing on multiplication and division [7, 8, 17]. With our tablet-computer-based software, called Classroom Learning Partner (CLP), students use a combination of digital tools and freehand drawing to create and manipulate mathematical representations in an electronic *notebook*. Using the tablet pen, students choose from among the digital tools, available via a command bar along the top of their tablet computer screens, to create representations that accurately reflect mathematical quantities; they use the tablet pen to annotate via drawing, writing, and highlighting in order to explore and record connections they see among representations, symbolic expressions, problem statements, and their own verbal explanations. They then wirelessly submit their work to their teacher, who can view the work on a tablet, e.g., to identify students who might need help or to choose for class discussion examples of alternate problem-solving strategies. Viewing student work in real time in a classroom, however, can be overwhelming—even in a class of 20 students, teachers can receive over a hundred submissions of student work in a single lesson. Our aim is to help teachers by developing machine analysis routines that can provide insights into students' use of visual representations and the mathematical thinking that is revealed through creation and manipulation of those representations. Such routines could, for example, alert a teacher about errors or apparent misunderstandings reflected in students' representations.

One of the visual representations important in teaching and learning multiplication is the array, which represents multiplication in terms of rows and columns [5]. Curricula often introduce the concept of multiplication with the idea of multiple copies of a group of a particular size. Using this idea, the multiplication problem 4 x 8 can be thought of as four groups of eight and can be represented by an array with four rows and eight columns. Each row represents a group of eight, and the product is the total number of cells in the array. Students can determine the product by employing a method called skip counting, in which they count by the size of the group, e.g., 8, 16, 24, 32. Over time, as students practice with and solidify understanding of relationships between skip counting and groups, they gain a foundation of meaning for the traditional multiplication facts and, with practice, begin committing the facts to memory.

Figure 3.1 shows an example of skip counting along the edge of an array in order to answer a problem that asks how many apples are in 4 bags of apples, with 8 apples in each bag. The student creates an array by tapping on an array tool icon on CLP's command bar and filling in numbers for rows and columns when prompted—4 rows and 8 columns in this example. She then counts by 8, and at the end of each row writes the total so far. The student's use of this representation suggests that she understands the idea of equal groups and has used each row of the array to represent a group, arriving at a correct final answer of 32. When a student's answer is incorrect, knowledge of the numbers in a skip counting sequence can help a teacher distinguish between careless arithmetic mistakes and conceptual misunderstanding.

Fig. 3.1: Student's array for 4 bags of apples, 8 apples in each bag (4 x 8), with skip counting to find the product of 32

The goal of the work reported in this paper is to provide insights into students' mathematical thinking by developing machine analysis routines that will identify and interpret handwritten skip counting on arrays such as shown in Figure 3.1. The task involves two steps: identifying digital ink strokes that are used in skip counting, while rejecting other strokes that are not; then interpreting the ink strokes in order to produce a sequence of numbers. Both steps are challenging. First, to identify likely skip counting sequences, the routines must distinguish between ink that is part of a sequence and ink that is not. In the work shown in Figures 3.2a through 3.2d, for example, the ink immediately outside the array edges should be considered part of the skip-counting sequences, except for the circles around the last numbers in the sequences in Figure 3.2a. The ink inside the arrays in Figure 3.2e should be considered part of a skip-counting sequence, but the ink inside all other arrays shown in Figures 3.2a through 3.2h should not be. In Figure 3.2f, one array should be identified as having skip counting while the other should not. Finally, none of the ink in close proximity to either inside or outside edges of arrays shown in Figures 3.2g and 3.2h should be considered skip counting. Thus identifying ink that is potentially part of a skip-counting sequence is not simply a matter of grouping all ink that is outside an array edge, as might be surmised from the neatly written ink shown in Figure 3.1.

The second challenge is that of interpreting children's handwriting. Off-the-shelf handwriting recognition systems are trained on adult handwriting samples that are typically horizontal strings of text. Children's handwriting can be messy, but even when written neatly, it can differ significantly from adults' [16, 15, 6]. More importantly in our context, students write a combination of numbers, symbols, and random strokes in whatever locations they decide work best for their visual representations; the writing rarely appears in neat horizontal lines. The handwriting interpretation challenges are evident in Figures 3.2a through 3.2h.

CLP's machine analysis routines meet the challenges illustrated in Figures 3.2a through 3.2h, identifying 100% of all skip-counting sequences, with no ink sequences falsely identified as skip counting. The routines also are able to correctly interpret 94% of the numbers in skip counting sequences to the right of an array. The key to the success is to use the structure of the array and knowledge about mathematics, in this case about skip counting, to bias the analysis routines. (Interpretation of sequences along the bottom of an array is more difficult and is discussed in the Current and Future Work section.) In the following sections we describe our data set of student work, the machine analysis routines, and results of running the analysis routines on the data set. We conclude with a discussion of contributions and implications for future development.

3.2 Methodology

Student work in the final assessment for a five-week trial in a third grade class of 22 students is at the center of our current machine analysis efforts. The purpose of the trial was to investigate how pen-based technology that combines drawing with digital tool use can support elementary students in learning the concepts of multiplication and division. The assessment consisted of 12 multiplication and division problems—six word and six non-word—from Singapore Math's *Math In Focus* curriculum, which was being used in the school district.

Our data for developing and testing the machine analysis routines consists of the CLP electronic notebooks that contain the students' work on the assessment problems. Each student notebook contains final representations for each notebook page, along with re-playable interaction histories that capture the process of creating and interacting with the representations. This paper reports on machine analysis routines that operate on the final representations.

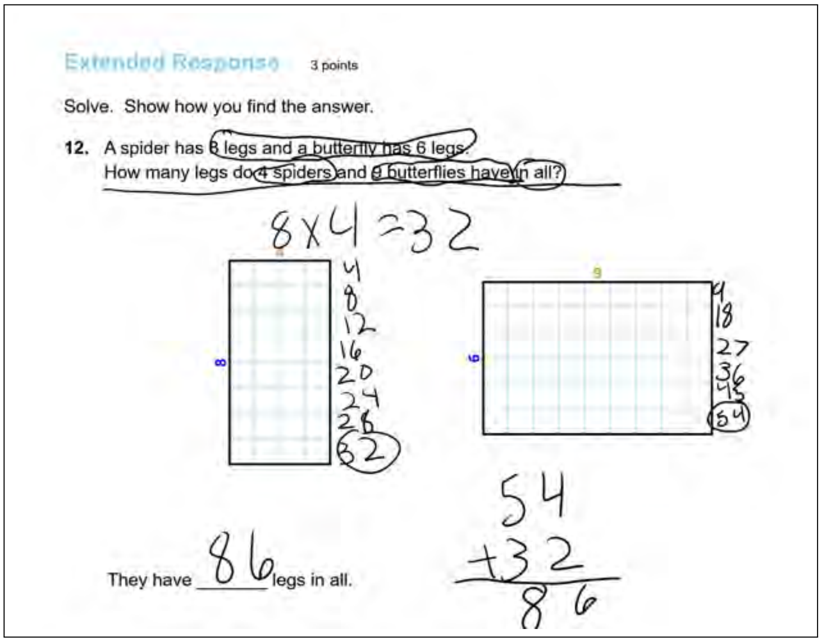


Fig. 3.2a: Extra ink encirling skip count numbers

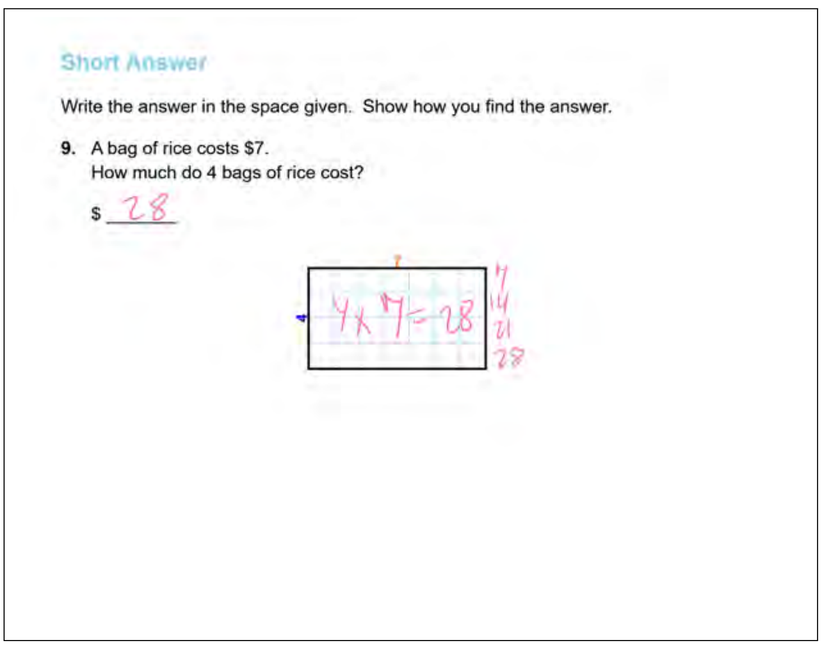


Fig. 3.2b: Extra ink inside array

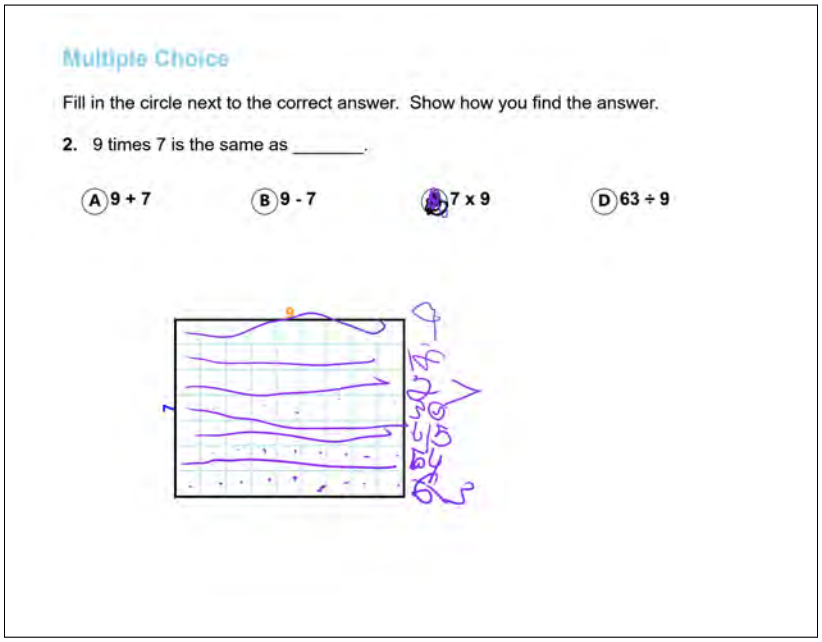


Fig. 3.2c: Messy handwriting, extra ink inside

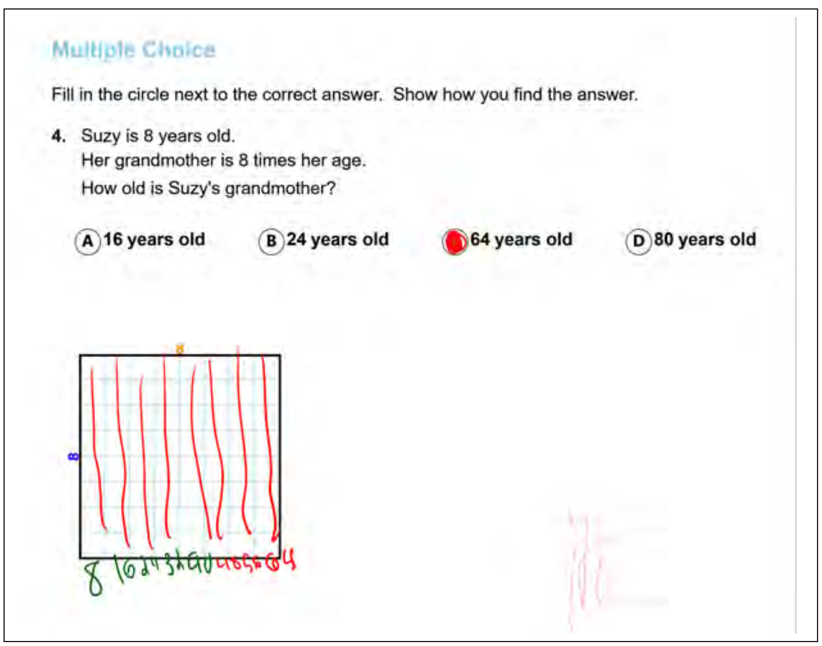


Fig. 3.2d: Extra ink inside, crowded ink along edge

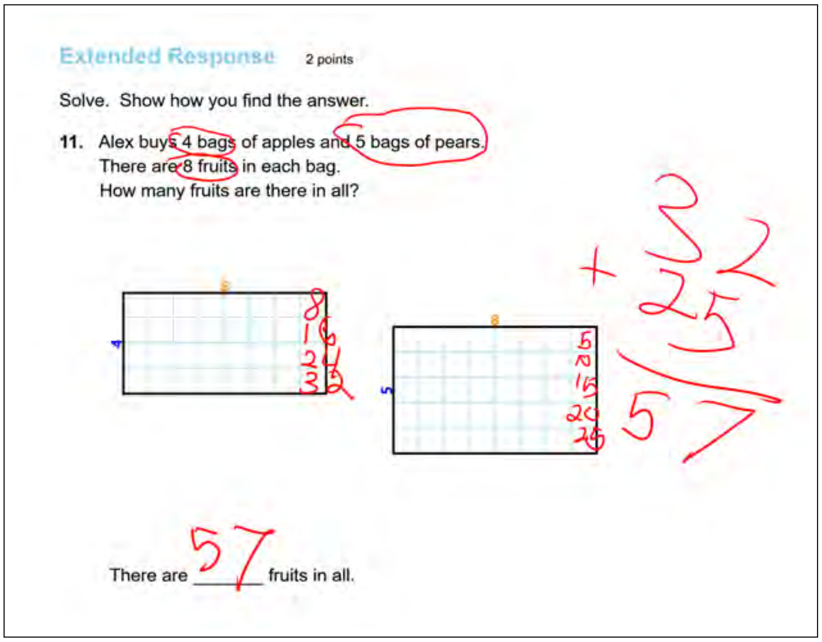


Fig. 3.2e: Skip counting inside and outside arrays

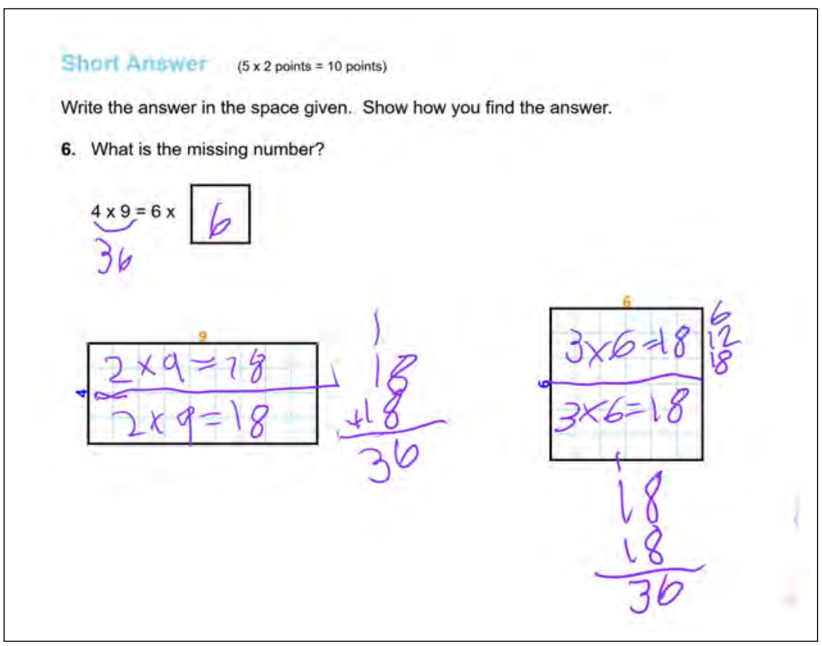


Fig. 3.2f: Extra ink inside and outside arrays

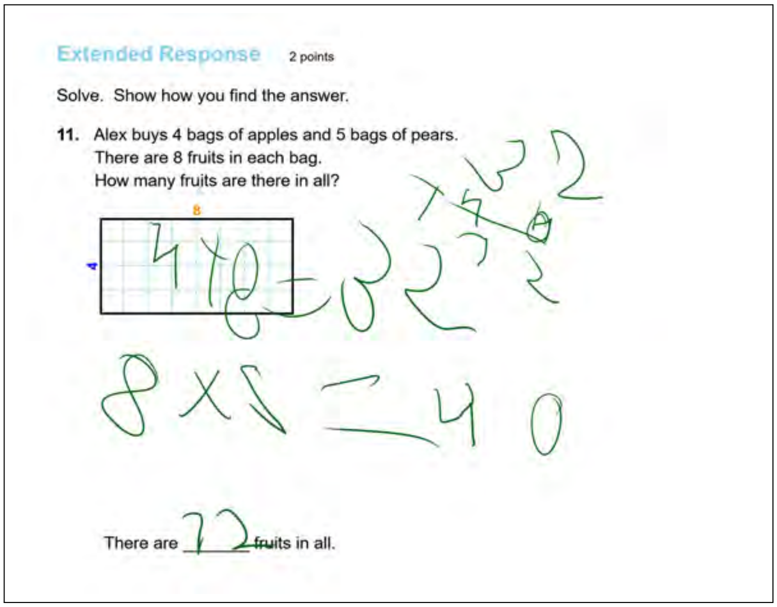


Fig. 3.2g: Ink inside and outside arrays near edges

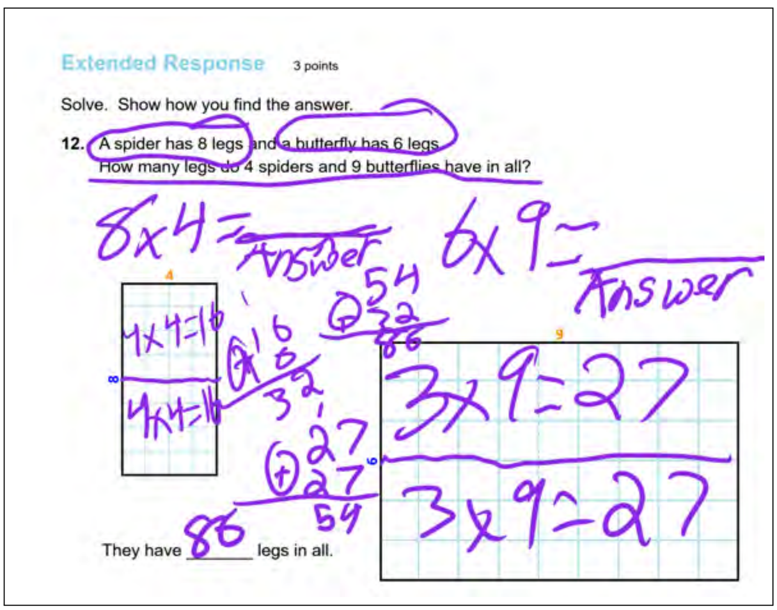


Fig. 3.2h: Extra ink intersecting array edges

Fig. 3.2: Student work illustrating challenges in identifying and interpreting skip counting

3.2.1 Representations Used

Students created representations using digital ink and three digital tools: a stamp, for drawing an image and creating multiple copies [12, 8]; a number line [1]; and an array [18]. Of the total of 264 pages of student work (22 students working 12 problems each), 74% of the pages include a final representation created using a digital tool. Approximately a third of those representations are arrays, for a total of 81 arrays, all of which are used in multiplication problems. In addition, 76 of the arrays (94%) have accompanying ink annotations, with 54 skip-counting sequences on 51 of the arrays—44 arrays with skip counting only on the right edge, 4 only on the bottom, 3 on both right and bottom. Other ink annotations include drawing a line across an array as part of a partial product strategy, e.g., as in Figures 3.2f and 3.2g; equations inside an array, e.g., as in Figures 3.2b, and 3.2f through 3.2h; and dots or lines, e.g., as in Figures 3.2c and 3.2d.

The large number of instances of skip counting on the right of an array is not surprising: The third grade curriculum being used represented a group as a row in an array, and the teacher modeled multiplication by writing a skip-counting number to the right of each row. Since the majority of skip counting in our data set was on the right of arrays—87% of the skip-counting sequences—we have focused our current machine analysis efforts on identifying and interpreting skip counting on the right.

3.2.2 Machine Analysis

The process of developing skip counting analysis routines started with the questions: How do math educators describe skip counting? How could a machine reproduce that description?

The action of writing the skip-counting sequence of 8, 16, 24, 32, shown in Figure 1, would be described by math educators as skip counting by 8 from 8 to 32 along the right side of a 4 by 8 array. Written in terms of a coding scheme developed by the first two authors and two additional math education researchers, the expression for this action is ARR skip [4x8: 8, 8-32, right].[1] To reproduce this expression, machine analysis routines must output a sequence of numbers by identifying and interpreting the germane digital ink strokes, then summarize the sequence. Our current machine analysis routines perform the first of these tasks—outputting the sequence; the summarization step is discussed in the Current and Future Work section.

To meet the challenges in identifying and interpreting digital ink skip-counting sequences, CLP's machine analysis routines use knowledge about the structure of arrays and the nature of skip counting. Skip-counting numbers to the right of an array, for example, are likely to be written very close to an array, be aligned with

[1] The coding process and results of human analysis of the data set are described in a forthcoming paper.

consecutive rows of the array, and be about the same height as a row. Using these observations, the machine analysis routines can (1) use the structure of an array to group strokes associated with each row, (2) use the expectation of strokes aligned with consecutive rows to reject ink strokes that are not part of skip-counting sequences, and (3) use the expectation of a sequence of numbers and the values we would expect to see based on the array dimensions to improve handwriting recognition results. Employing these techniques, CLP is able to produce human-like coded expressions for skip-counting sequences. In the example discussed above, CLP produces the expression ARR skip [4x8: "8, 16, 24, 32", right].

CLP's first step in identifying and interpreting skip-counting sequences is, for each array on a page, to search all ink strokes on a page, using the array's location and size to reject strokes that are unlikely to be in a skip-counting sequence and returning a group of ink strokes that could be: It rejects any strokes that are much smaller than an array cell, e.g., stray dots, then uses the height of the array to set a bounding box, which extends from half a row height above to half a row height below the array, 80 pixels to the right of the right edge, and 40 pixels inside the right edge.[2] It rejects any strokes that have less than 50% of their height inside the bounding box or whose weighted center is not inside the bounding box. It prunes the remaining strokes by rejecting strokes that are too large (twice the row height), then prunes a second time by calculating an average stroke height for the remaining strokes and rejecting strokes that again are too large (twice the average stroke height). Using a factor of two when determining thresholds for stroke rejection works well with the students' ink annotations in our data set——no necessary strokes are ever rejected, and any unnecessary strokes are pruned in later steps. At this point, CLP has a set of candidate skip-counting strokes.

CLP's next step is to partition the candidate ink strokes into groups, where each group potentially corresponds to a skip-counting number. The partitioning is done by determining with which row ink strokes are aligned. This task would seem straightforward except for two challenges: Small strokes may be aligned incorrectly with an adjacent row, e.g., the small top stroke of a 5 may be drawn in the row above the body of the 5; and strokes may overlap more than one row, e.g., as with the "27" in Figure 3.2c. The first of these challenges turned out to be easily met by our solution to the second challenge: The region tested for stroke overlap is not just a single row, but instead extends from half a row above to half a row below the row being checked. Each stroke then is grouped with the row whose extended region it overlaps the most. For cases in which a stroke overlaps two rows equally, the stroke is included in both rows.

Each row's group of ink strokes then is sent to a handwriting recognizer, which returns a list of possible interpretations.[3] CLP achieves high recognition rates by choosing number interpretations over non-numbers and, when presented with multi-

[2] Left and right bounds were determined empirically using the examples in the current data set. Further testing on a larger corpus may refine the dimensions.

[3] CLP is written in C# and currently runs in the Windows 8 and 10 operating systems; it uses the Microsoft English Handwriting Recognizer introduced in the Windows 8 (`https://msdn.microsoft.com/en-us/library/ms840450.aspx`).

digit options, choosing one that has the same number of digits as the value expected based on the array dimensions. In the student work shown in Figure 3.2a, for example, for the last number in the sequence on the left, the handwriting recognizer returns choices of 32 and 320. CLP chooses 32 by using the dimensions of the array to determine that 32 is in the expected sequence of numbers. (Using the preceding interpreted numbers to predict expected sequence is discussed in the Current and Future Work section.) CLP also improves recognition rates by using heuristics to correct commonly misinterpreted strokes that occur in our data set: "&" is replaced with 8, ">" is replaced with 7, "Z" and "z" are replaced with 2, "|" and "l" (lowercase L) are replaced with the number 1.

CLP's final step is to use a set of rules based on knowledge of skip-counting patterns to reject any sequences that are likely to not be skip counting. Sequences are rejected by checking their interpreted values and finding that one or more of the following is true: the first two rows do not have values, fewer than two rows have values, more than two rows share the same value, the first row does not have a value and more than 50% of the rows do not have values (which handles arrays such as the one on the right in Figure 3.2f), fewer than 34% of rows with values have numeric values, there is more than one gap of one row between values, or there is a gap of more than one row between values.[4]

3.3 Evaluation Results and Discussion

We evaluated CLP's machine analysis of skip-counting sequences in two ways: comparing CLP's results against human analysis results, and comparing CLP's results with results from two versions of machine analysis routines that use little or no knowledge of arrays or skip counting in identifying and interpreting sequences.

3.3.1 Comparison With Human Analysis Results

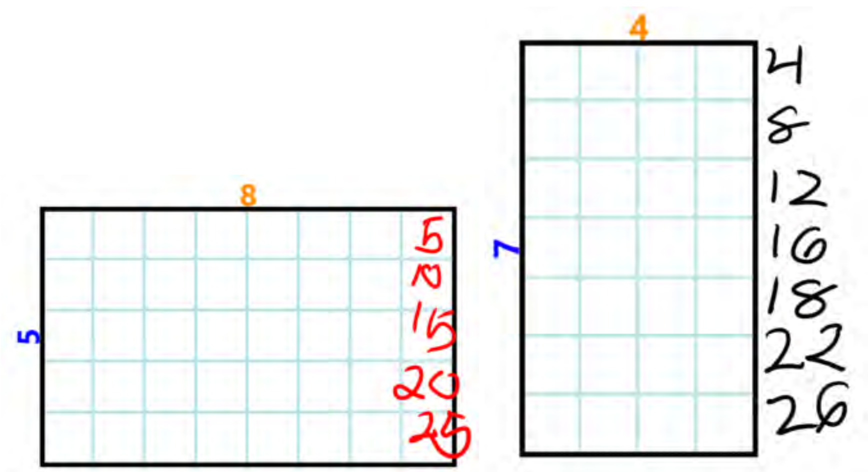
Using the analysis routines described in the previous section, CLP currently can identify skip counting along the right side of an array very accurately: It identified all 47 skip-counting sequences on the right of arrays in our data set and did not falsely identify any non-skip-counting ink as skip counting. It produced no interpretation errors in 37 of the sequences, thus reporting completely correct sequences in 79% of the sequences. It correctly identified 279 numbers out of a total of 296 numbers in all skip-counting sequences—94% of the numbers.[5]

The degree of accuracy achieved by the analysis routines means that CLP can present reliable information to a teacher about students' skip-counting strategies:

[4] The threshold cutoffs were empirically determined.

[5] We mean number here, rather than digit, e.g., 16 is the number 16, not the "numbers" 1 and 6.

It can identify which students used the strategy and present information about the accuracy of the students' sequences. It can identify, for example, that the student whose work is shown in Figure 3.3a (and also in Figure 3.2e) skip counted by the wrong dimension—5 instead of 8—which results in an answer of 25 for 5 x 8. This error is fairly common and may indicate some misunderstanding on the student's part about the difference between the size of a group and the number of groups or about how each quantity is represented in the array. Shown in Figure 3.3b is an example of student work containing an arithmetic mistake: The student has added 4 to all elements except 16, which produces an incorrect answer of 26 rather than 28 for 7 x 4. This information suggests that the student has a relatively robust understanding of skip counting, but has made a careless arithmetic error, information that is not available just from knowing that the student's answer is 26.



(a) Skip counting by the wrong dimension (b) Arithmetic error from 16 to 18

Fig. 3.3: Skip counting mistakes that CLP can identify

3.3.2 Effect of Using Array and Skip Counting Knowledge

In order to evaluate the effect of using knowledge of array structure and skip-counting patterns, we compared the results of CLP's current machine analysis routines with two other versions of the analysis routines, each of which used less knowledge in attempting to identify and interpret skip-counting sequences. All versions prefer numerical to nonnumerical characters when the handwriting recognizer offers a choice. Version 1 employs only a bounding box along the right edge of each array, interpreting as a single string all strokes that fall within the box. Version 2 collects strokes using a bounding box and information about an array's row height, which allows it to improve results by rejecting strokes that are much shorter or much taller than the row height; it too interprets the collected strokes as a single string. Version 3—our current version, described in Section 3.2.2—uses a bounding box and array row height to collect strokes, interprets groups of strokes aligned with rows, and uses heuristics to improve results. Version 3 returns a sequence of strings, each of which corresponds to a number, with one number per row of an array.

Table 3.1 presents results for how often each routine interpreted a sequence correctly, identified skip-counting sequences when it should not have (false positives), and failed to identify skip-counting sequences (false negatives). As is evident from the table, the more knowledge of arrays and skip counting used in the analysis routines, the better the results. Especially of note are the large counts of false positives for versions 1 and 2: These versions have no reliable way to distinguish skip-counting ink from non-skip-counting ink, so they consider all ink near the right edge of an array to be skip counting. In addition, versions 1 and 2 reject few or no ink strokes, so they will not miss any skip-counting sequences. Version 3, on the other hand, rejects quite a few ink strokes, but is still able to identify all skip-counting sequences.

Table 3.1: Results for different machine analysis routines

	Version 1 **bounding box**	**Version 2** **bounding box** **+ row height**	**Version 3** **all knowledge**
# sequences interpreted correctly (of 47)	8 (17%)	10 (21%)	37 (79%)
# false positives	19	18	0
# false negatives	0	0	0

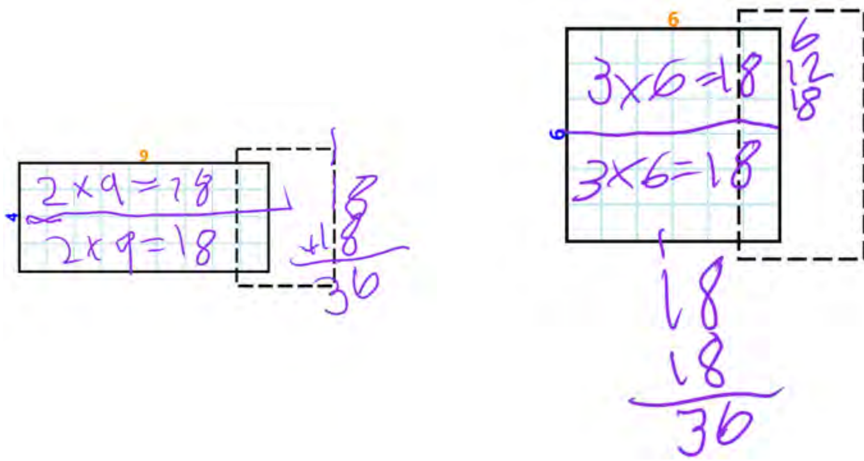
As an additional measure of the interpretation accuracy for each of the versions, we calculated a *character error rate* (CER), which is a ratio of the number of errors in a string to the number of expected characters and is commonly used in handwriting recognition evaluation [20, 15]. To quantify the number of errors, we used *edit distance* (ED), which measures dissimilarity between two strings by counting the minimum number of steps needed to transform one string into the other [4]. We used Levenshtein edit distance [10], which sums insertions (I), deletions (D), and substitutions (S) required for the transformation: I + D + S. Dissimilar strings yield higher edit distances. CER is then ED ÷ N, where N is the number of expected characters in the string used as a reference. In our case, the reference string is created by concatenating the expected skip-counting sequence numbers, each separated by a single space, which is the delimiter used in the strings produced by versions 1 and 2.

Version 3 outperformed the other two versions, producing much lower CER values for all 47 skip-counting sequence on the right of arrays in our data set. Table 3.2 presents the average CER results for the 47 sequences and for the 10 sequences for which version 3 produced interpretation errors. Again, the use of knowledge of arrays and skip counting significantly improved version 3's performance over the other two versions.

Table 3.2: Character Error Rate (CER) values for different machine analysis routines

	Version 1 bounding box	**Version 2 bounding box + row height**	**Version 3 all knowledge**
Average CER	27	23	3
Average CER for v3 sequences with errors	43	39	18

The examples shown below illustrate the differences in interpretations across the three versions of machine analysis routines. For the work shown in Figure 4 (also shown in Figure 3.2f), the bounding boxes for both arrays collect extraneous ink—–the "+1" outside the 4x9 array and the "8" in each equation inside the 6x9 array—causing versions 1 and 2 to produce interpretations with very high error rates. Version 3, however, is able to reject the extraneous ink and produce correct interpretations for both arrays.



(a) 4x9 array
v1: "4"
v2: "1 -"
v3: no skip counting

(b) 6x9 array
v1: "in 88 8"
v2: "b 12 '8"
v3: "6" "12" "18"

Fig. 3.4: Two examples in which version 3's interpretations contain no errors

For the work in Figures 3.4a and 3.4b, version 3's interpretation contains no errors. For the work shown in Figure 3.5 (also Figure 3.2c), version 3's interpretation contains three errors in the sequence of seven numbers for the skip-counting by 9 from 9 to 63. Unlike the interpretations for versions 1 and 2 for this example, however, the majority of numbers are correct, and as discussed in next section, CLP can infer values for the misinterpreted numbers.

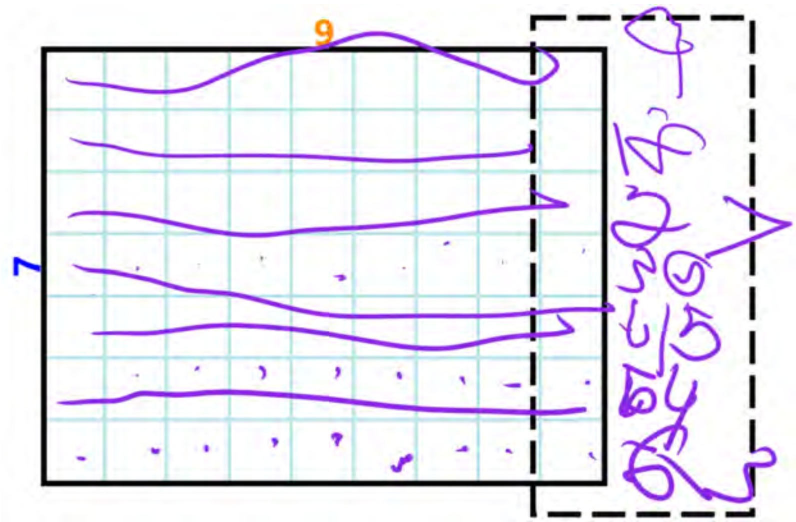
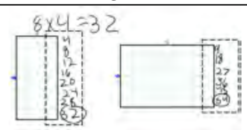
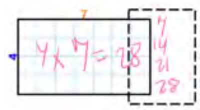
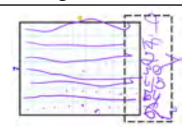
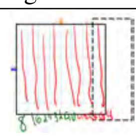
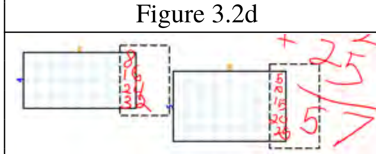
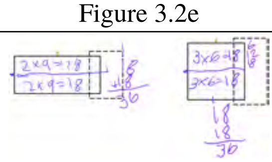
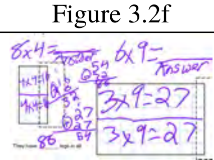
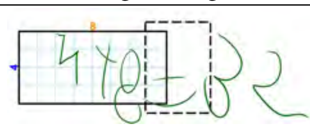


Fig. 3.5: Comparison of interpretations for different machine analysis routines
v1: "℅ 52 30s 05 tubby –863"
v2: "I 27 30s '6/5 tby 363",
v3: "9" "18" "27" "30" "05" "i4" "63"

To further illustrate differences in interpretations across the three versions, Table 3.3 presents the bounding boxes, interpretations, and CER values for the student work examples shown in Figure 3.2.

Table 3.3: Interpretations and CER values for the student work shown in Figure 3.2

Arrays + Bounding Boxes	Interpretations	CER
	v1 4x8: “in’s 160 2 24 28 320”	43
	v1 6x9: “98 27 36 45 540”	19
	v2 4x8: “4 F 12 16 20 24 28 32”	5
	v2 6x9: “918 2] 56 45 s☺”	31
	v3 4x8: “4” “8” “12” “16” “20” “24” “28” “32”	0
Figure 3.2a	v3 6x9: “9” “18” “27” “36” “45” “54”	0
	v1: “17% 8211 d8”	70
	v2: “17% 11 d8”	60
Figure 3.2b	v3: “7” “14” “21” “28”	0
	v1: “% 52 30s 05 tubby -863”	79
	v2: “I 27 30s ’6/5 tby 363”	68
Figure 3.2c	v3: “9” “18” “27” “30” “05” “i4” “63”	16
	v1: “.....”	n/a
	v2: “to”	n/a
Figure 3.2d	v3: “not skip counting”	n/a
	v1 4x8: “8. 6 24 32”	20
	v1 5x8: “is 205 25”	62
	v2 4x8: “£ 6, 22/ 39”	60
	v2 5x8: “15 15 20 25”	23
	v3 4x8: “8” “16” “24” “32”	0
Figure 3.2e	v3 5x8: “5” “10” “15” “20” “25”	0
	v1 4x9: “4”	n/a
	v1 6x6: “in 88 8”	71
	v2 4x9: “1 -”	n/a
	v2 6x6: “b 12 ’8”	14
	v3 4x9: “not skip counting”	n/a
Figure 3.2f	v3 6x6: “6” “12” “18”	0
	v1 8x4: “\Eigen”	n/a
	v1 6x9: “no strokes in boundary”	n/a
	v2 8x4: “-161 K f”	n/a
	v2 6x9: “no strokes in boundary”	14
	v3 8x4: “not skip counting”	n/a
Figure 3.2g	v3 6x9: “not skip counting”	0
	v1: “_-”	n/a
	v2: “5”	n/a
Figure 3.2h	v3: “not skip counting”	n/a

3.4 Current and Future Work

We currently are focusing on several areas of development. To simplify our methods for aligning ink strokes with rows, we will investigate the idea of first grouping ink strokes that are aligned with each other, then aligning those groups with rows, as demonstrated by [21].

Our current machine analysis routines produce accurate results for identification and interpretation of skip counting along the right side of an array. They also reliably identify skip counting along the bottom of an array by using a bounding box and evaluating a single interpreted string with respect to an expected string determined by the array dimensions. The routines, however, do not yet attempt to produce a skip-counting sequence. We are working on interpretation techniques that will make use of ink alignment with columns, similar to CLP's alignment with rows. Interpretation of skip counting along the bottom of an array, however, poses the added challenges of lack of clear separation of sequence numbers, which is possible with one number written per row; and the limited horizontal space available with narrow arrays, which can cause numbers to not be aligned with columns.

We are implementing routines that can use knowledge of a skip–counting interval to infer a pattern when a skip-counting sequence contains handwriting recognition errors. Currently, skip-counting intervals are determined using the dimensions of an array. For the 7x9 array shown in Figure 3.5, for example, a skip-counting sequence on the right would be expected to contain 7 numbers, start at 9, have an interval of 9, and end at 63. Using this information, the routines can infer that in the interpreted sequence 9, 18, 27, 30, 05, i4, 63, the 30 may be 36, the 05 may be 35, and the i4 may be 54. It then can report to the teacher both the original interpreted sequence and a summary: skip counting by 9 from 9 to 63 on the right, i.e., ARR skip [7x9: 9, 9-63, right]. We also are working on an alternate method for determining a skip-counting sequence interval—subtracting pairs of consecutive numbers in a student's sequence. This technique has the advantage of being able to report to a teacher when a skip-counting sequence is not related to an array, and it takes advantage of not just general information about array structure, but also specific information about what a student actually wrote.

An interesting variation on the array skip-counting representations in our current data set is shown in Figure 3.6. In this work, created during the five-week trial, a student has used a skip-counting strategy for division, building up to the dividend by creating single-row arrays and skip counting along the side of each array until she reaches the dividend. We are working to extend CLP to handle this case, as our current machine analysis routines assume skip counting occurs only on multi-dimensional arrays.

As mentioned earlier, CLP stores both a final representation and a re-playable interaction history, which captures a student's process of creating and interacting with the representation. We are working on routines that will analyze an interaction history in order to provide additional insights into a student's mathematical thinking—insights often not possible when looking at only a final representation. We have found, for example, that students often try several representations before settling on

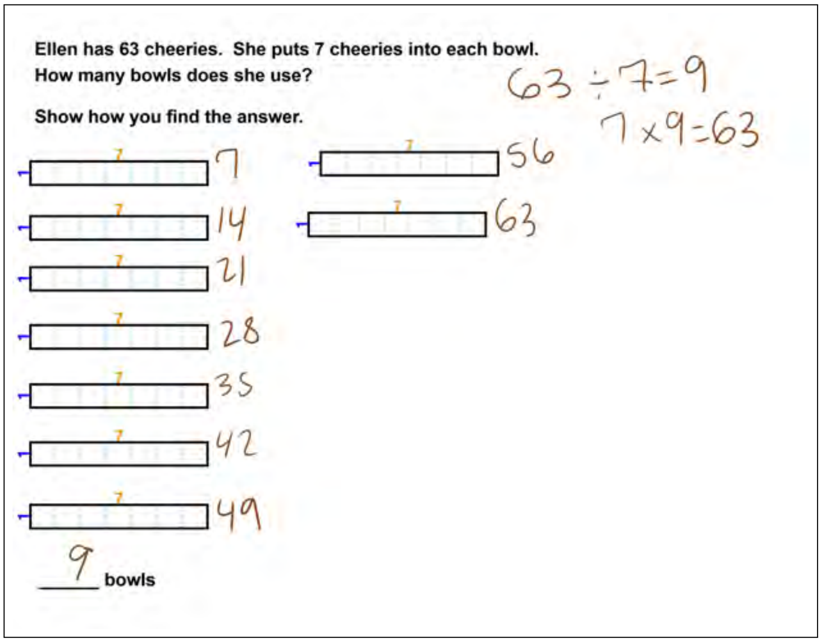
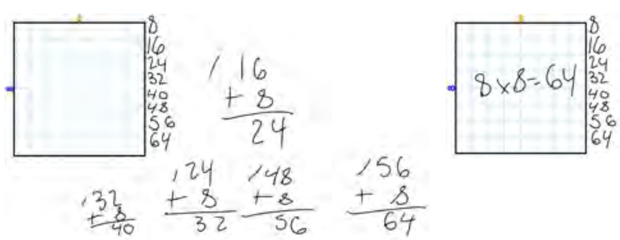


Fig. 3.6: Student's use of multiple arrays and skip counting

a final one to submit to their teacher. We also have found that students often do arithmetic, which they erase before submitting their work. In the example shown in Figure 3.7, for example, the final representation does not give any indication that the student wrote 8 and 16 alongside the array, then did arithmetic for 24, 32, and 40, wrote 48 alongside the array, then did arithmetic for 56 and 64. This interaction indicates that the student is comfortable with her 8 times table up to 16, then must use arithmetic for larger numbers, except for 40 + 8.



(a) Arithmetic used to compute sequence

(b) Arithmetic erased in final representation

Fig. 3.7: Interaction history reveals student's use of arithmetic to complete sequence

Finally, we are working on machine analysis routines for other types of array ink annotations, such as the use of a line to indicate a partial product strategy as shown in Figures 3.2f and 3.2g. We also are working on machine analysis routines for other types of representations such as a number line [1]; stamp, which enables students to draw an image and make multiple copies [12, 8]; division template, which is an interactive visual representation for the process of division [18, 9]; and bin, which supports what are called dealing out strategies [11, 22]. With an extended set of machine analysis routines, we will be able to increase our data set to include 8000 additional pieces of student work—work from the full five-week trial in the third grade class whose assessment we are currently using, plus work from a previous five-week trial in a fourth grade class.

3.5 Conclusion

This paper reports progress made on the development of machine analysis routines that accurately interpret elementary math students' visual representations created using a combination of freehand drawing and a digital array tool. Using the final assessment from a five-week unit on multiplication and division in a class of 22 third grade students, the routines, which are part of our Classroom Learning Partner (CLP) software, are able to identify all final array representations that are accompanied by a handwritten sequence of numbers that exhibit a common problem-solving strategy called skip counting. The majority of the skip-counting sequences occur along the right side of arrays, and CLP is able to correctly interpret 94% of the handwritten numbers in the sequences in our data set, with 79% of the sequences interpreted completely correctly. The key to this success is to use knowledge about the structure of arrays and the nature of skip counting to bias the machine analysis routines. The effect of using such knowledge is evident from our evaluation of two versions of routines that do not use this knowledge—both suffer from high error rates in the interpretation of the handwritten skip-counting sequences and from the inability to distinguish skip-counting ink from non-skip-counting ink. With our analysis routines' current accurate identification and interpretation of skip-counting, CLP will be able to present reliable information about students' skip-counting strategies. Such information can give teachers and others valuable insights into students' use of visual representations and the mathematical thinking that is revealed. Our current work focuses on expanding the machine analysis routines to operate on additional types of representations and on not just a final representation, but on an interaction history that captures a student's process of creating and manipulating representations. With these extensions, we hope to expand the usefulness of information we can provide to teachers about their students' mathematical reasoning, to math education researchers about the ways in which visual representations support student learning, and to software designers about ways in which design interacts with machine analysis.

3.6 Acknowledgements

This research is funded by the NSF *INK–12: Teaching and Learning Using Interactive Ink Inscriptions* project, DRL–1020152 (Koile), DRL-1019841 (Rubin). We gratefully acknowledgment contributions from MIT's CLP research group members past and present, and from math education researchers Lily Ko and Marlene Kliman at TERC. We also thank Randall Davis, Jonathan Grudin, Tracy Hammond, YJ Kim, and Marlene Kliman for valuable feedback on drafts of this paper.

References

1. Benz, E.: Design and implementation of a number line tool for classroom learning partner (2014). Unpublished UROP project
2. Brizuela, B.M., Gravel, B.E.: Show me what you know: Exploring student representations across STEM disciplines. Teachers College Press (2013)
3. Cuoco, A.: The roles of representation in school mathematics (reston, va: Nctm) (2001)
4. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM **7**(3), 171–176 (1964)
5. Fosnot, C.T., Dolk, M.L.A.M.: Young mathematicians at work. Heinemann Portsmouth, NH (2001)
6. Kim, H.H.: Analysis of children's sketches to improve recognition accuracy in sketch-based applications. Ph.D. thesis, Texas A&M University (2012)
7. Koile, K., Rubin, A.: Animated mathematical proofs in elementary education. In: The Impact of Pen and Touch Technology on Education, pp. 67–79. Springer (2015)
8. Koile, K., Rubin, A.: Machine interpretation of students' hand-drawn mathematical representations. In: The Impact of Pen and Touch Technology on Education, pp. 49–56. Springer (2015)
9. Koile, K., Rubin, A.: Tablet–based technology to support students' understanding of division. In: Revolutionizing Education with Digital Ink, pp. 71–89. Springer (2016)
10. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady, vol. 10, pp. 707–710 (1966)
11. Markovits, A.: Machine interpretation of elementary math students' hand–drawn bin representations (2016). URL `http://ink-12.mit.edu/documents/AMarkovits_UAP.pdf`. Bachelor of Science Thesis. Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.
12. Maynard, E.D.: Using a structured vocabulary to support machine understanding of student work (2013). URL `http://ink-12.mit.edu/documents/2013.02_MaynardE_MEng.pdf`. Master of Engineering Thesis. Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.
13. NCTM: National council of teachers of mathematics. Principles and standards for school mathematics. Reston, VA (2000)
14. NGA: National Governors Association Center for Best Practices (NGA center). Common core state standards for English language arts &Literacy in history/social studies, science, and technical subjects. Washington, DC: National Governors Association Center for Best Practices and the Council of Chief State School Officers (2010)
15. Read, J.C.: A study of the usability of handwriting recognition for text entry by children. Interacting with Computers **19**(1), 57–69 (2007)
16. Read, J.C., MacFarlane, S., Horton, M.: The usability of handwriting recognition for writing in the primary classroom. In: People and Computers XVIII—Design for Life, pp. 135–150. Springer (2005)

17. Rubin, A., Storeygard, J., Koile, K.: Supporting special needs students in drawing mathematical representations. In: The Impact of Pen and Touch Technology on Education, pp. 57–66. Springer (2015)
18. Simon, E.: Design and implementation of a division tool for elementary math education (2014). Bachelor of Engineering Thesis. Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.
19. Smith, S.P.: Representation in school mathematics: Children's representations of problems. A research companion to principles and standards for school mathematics pp. 263–274 (2003)
20. Soukoreff, R.W., MacKenzie, I.S.: Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 113–120. ACM (2003)
21. Stahovich, T.F., Lin, H.: Enabling data mining of handwritten coursework. Computers & Graphics **57**, 31–45 (2016)
22. Yang, D.: Identifying bins-dealing strategies in Classroom Learning Partner (2015). URL `http://ink-12.mit.edu/documents/DYang_UAP.pdf`. Bachelor of Science Thesis. Electrical Engineering and Computer Science Department, MIT, Cambridge, MA.