Computer Vision-based Mathematics Learning Enhancement for Children with Visual Impairments

Chenyang Zhang[1], Mohsin Shabbir[1], Despina Stylianou[2], and Yingli Tian[1]
[1]Department of Electrical Engineering,
[2]Department of Secondary Education
The City College of New York, NY, 10031, USA
{czhang10, mshabbi00, dstylianou, ytian}@ccny.cuny.edu

***Abstract*—Children with visual impairments face disproportionate challenges in learning mathematics. In this paper, we explore how to apply computer vision techniques to enhance mathematics learning by fusion of multisensory information for children with visually impairments. The multisensory information includes visual and audio information. Our research focuses on the mathematics education for children with visual impairments in early ages (Kindergarten – Grade 3) by using arithmetic racks. A computer vision-based algorithm is developed to detect the numbers and positions of arithmetic rack beads and the detection results are then provided to students with visual impairments as speech guidance. Preliminary results demonstrate the effectiveness and efficiency of the proposed method for detecting the number, color, position, and moving directions of the arithmetic rack beads under different situations.**

Computer Vision; Math Learning; Children with Visual Impairement; Math Learning Enhauncement

I. INTRODUCTION

The study of mathematics can be challenging for many children. More so, children with visual impairments face even more disproportionate challenges in learning mathematics. Indeed, few students with visual impairments are currently participating in advanced mathematics classes in secondary schools and teachers of students with visual impairments report that they encounter continuing difficulties in providing materials and equipment for mathematics instruction [7]. Nearly 5 million preschool-aged children and about 12.1 million children ages 6-17 have visual impairments, according to the Braille Institute [12]. Clearly, innovative tools and techniques are needed to help these children with visual impairments learn mathematics more easily -- and perhaps multiply their career opportunities when they reach adulthood. Yet, little research and development efforts are directed towards addressing these challenges faced by children (and adults) with visual impairments and their teachers.

Our group has developed many algorithms for assistive technology to help blind people [4, 8-10, 12-17]. This paper attempts to investigate and develop computer vision-based techniques and models to enhance mathematics education for children with visual impairments in early ages (Kindergarten – Grade 3) by fusion of multisensory information: visual (camera), and hearing (audio). The visual information will be captured by a camera and the feedback will be provided to the visually impaired student by audio.

II. TEACHING MATHEMATICS CONCEPTUALLY BY USING ARITHMETIC RACKS

Work by cognitive psychologists indicates that even young children (as early as infancy) can often perceive small amounts (one to five) as a unit without a need for counting [1, 2]. *Subitizing* – viewing groups of objects as a unit – has been studied in six month babies: babies become habituated to a display of three objects and treat a new display of four objects as novel [11]. Mathematics educators capitalize on children's ability to subitize by first having young children work within the "structure of 5". Children are offered groups of objects from 1 to 5 and asked to determine the cardinality (size) of the group.

As shown in Figure 1, an arithmetic rack is a model that was specifically built to encourage and build on this innate ability of humans to subitize. It is a calculating frame, consisting of two rows of ten beads – a set of five red beads and a set of five white beads on each row. The five-structure of the arithmetic rack supports the development of part-whole relations in early number sense.

Figure 1.An arithmetic rack and a child working with it.

Educators suggest that early on, children are shown small amounts (less than five) on the arithmetic rack, with little reaction time (about 2 seconds) and are encouraged to recognize the amount [3]. Once children are comfortable with subitizing within the first five numbers on the arithmetic rack, teachers begin to show them quantities larger than five: For example, a number 7 (5 red beads and 2 white). By doing this, children are encouraged to use the "five structure" that is built in the arithmetic rack to develop understanding of quantities up to 10. They also begin to understand the decomposition of number (numbers can be seen as combinations of smaller quantities – e.g., 7=5+2) and the hierarchical inclusion in the number system (whole numbers grow exactly by one). In this manner, the arithmetic rack supports the gradual development of the "ten structure" where children comfortably work within the first 10 numbers.

Gradually, children are introduced to the second row of the arithmetic rack and are encouraged to work within the first 20 numbers understanding the structure that extends beyond ten. Children can then work with double numbers (e.g., 3+3), or near doubles (e.g., 3+4, which can then be seen on the arithmetic rack as 3+(3+1) or, 7+7= (5 red+ 2 white)+(5 red + 2 white)) as well as the building of tens. Work with this model helps children automatize basic facts, but also builds the skill to look for the structure, the relations and regularity in numbers [6].

III. Detecting Numbers and Positions of Arithmetic Rack Beats based on Camera Visual Information

In this paper, we develop a computer vision-based framework to detect and track the numbers and positions of arithmetic rack beads to facilitate the development of this number sense for children with visual impairments. The flowchart of our framework is illustrated in Figure 2. A camera is used to monitor the arithmetic rack while the blind student is learning.

Our proposed framework includes two main steps: calibration and synchronization. The first step is to calibrate state frames in an input video sequence captured by a camera and to compute the initial state and properties of a "virtual" arithmetic rack (i.e. a digital arithmetic rack stored in computer). The second step is to generate the synchronization between state frames of the camera-captured input video and the current state of the "virtual arithmetic rack" in a sequential manner. The details of each step in Figure 2 will be described in Sections III.A to III.E. Experimental results are demonstrated and discussed in Section IV. Section V concludes our work.

As shown in Figure 1-left and Figure 3-left, the arithmetic rack has two rows of beads: top and bottom rows. Each row has ten beads in total with five in red and five in white. All the beads can be moved separately or jointly to show different arithmetic equations. To automatically detect and track arithmetic rack beads by computer, an abstract and informative digital arithmetic rack is generated to help visual impaired children.

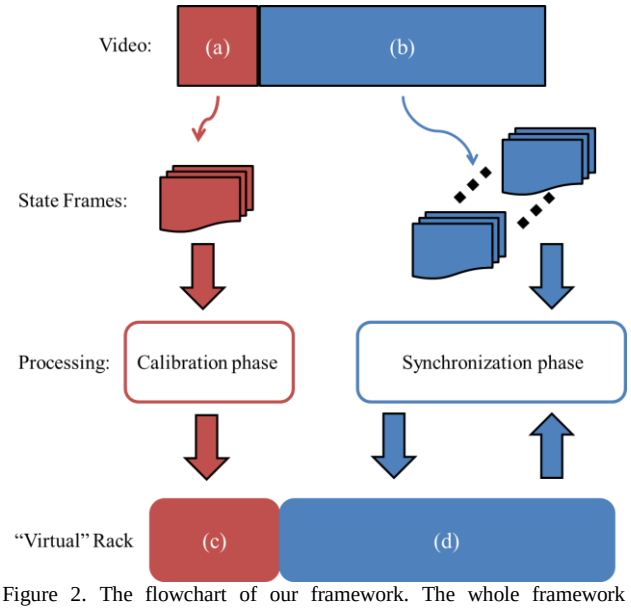

Figure 2. The flowchart of our framework. The whole framework includes two main steps: 1) calibration, which is used to take (a) calibration of the input video sequence to compute (c) the initial state and properties of "virtual" arithmetic rack (i.e., a digital arithmetic rack stored in computer); 2) synchronization, which is used to synchronize between (b) the input video and the current state of (d) "virtual arithmetic rack" in a sequential manner.

A. *Arithmetic Rack Modeling*

As shown in Figure 3, we create a vitual arithmetic rack (Figure 3-right) and synchronize the bead movements based on the video the real arithmetic rack captured by a camera (Figure 3-left). Unlike the traditional arithmetic rack that uses the same shape and material for both white and red beads, we use different shapes and materials (plastic cubes for red beads and wood balls for white ones), which are more suitable for visually impaired students to touch and feel while maintaining the structure of our framework. In the virtual arithmetic rack, we use '@' to represent red beads and 'o' for white ones.

The digital arithmetic rack is quite simple and just for illustration. It is represented by several variables which thoroughly measure and describe the states and properties of the real arithmetic rack: average area of bead, variance of area of bead, color of bead, position of bead (left side or right ride), row of bead, and the number of beads.

B. *Task description*

In order to transfer the visual information to digital information which can be understood by computer, we utilize image processing technologies together with computer vision technologies to transfer the state frames to a certain representation. This is a brief and informative form that can be easily used to synchronize the real arithmetic rack and the digital one. In brief, our tasks in this paper are to automatically detect and track the number, color, position, and movement of beads and to synchronize the movements to the virtual arithmetic rack stored in computer.

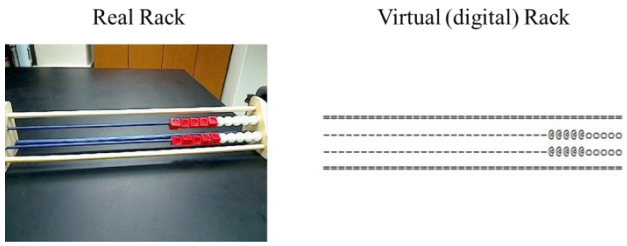

Figure 3. Illustration of a real arithmetic rack used in our work and the visualization of our synchronized arithmetic rack..

C. *Calibration*

In this section, we describe the algorithm to calibrate a real arithmetic rack to generate the virtual arithmetic rack for detecting and tracking the number, color, and position of beads.

State Frame. We define a current frame of a camera video as "state frame" when the current frame is relative static comparing to its temporal neighbor frames after the beads are moved to different positions. This assumption is suitable to our tasks because when an educator is showing the equation represented by the arrangement of the beads, the rack is usually static; in addition it is usually different from the previous arrangement.

Extracting state frames from an input video can also benefit to our tasks. By comparing the current state frame with the previous state frame, our algorithm is able to acquire the information which contains "what is the current arrangement of the beads", as well as "how many\ which color\ which row beads did the educator move". For example, if the previous state frame describes "there are FIVE RED beads on the top-left side and no white ones" and current state describe "there are TWO RED beads on the top-left and no white ones", from which we can infer an equation "5-3=2" besides each state themselves.

In our work, we first start from how to recognize and infer such information from state frames and then how to extract such state frames from an input video.

What to Learn in Calibration. Basically, we apply frame difference to capture the changed region between two state frames. The position (x-y coordinates) and area (how large is the moved region) in images can be used to infer which row and how many beads are moved. Then combining the color information, we can determine the color and direction of the moved beads.

Due to the perspective effect, an object with the same size appears larger in the image when it is near to the camera and smaller when it is far from the camera. Furthermore, the camera is not constraint to be perfectly in front of the arithmetic rack and the bead sizes in images are not required to be the same (in terms of number of pixels in the state frame). Such effect is manifested as shown in Figure 4: the red region is larger than the blue region, because the left-side is slightly closer to the camera than the right-side.

Both the left-side area and right-side area will be stored with scalar values. Moreover, in our experiments, we observe that different colors result in different size of motion area since we use different shapes for beads with different colors. So we stored the areas of beads of different colors separately.

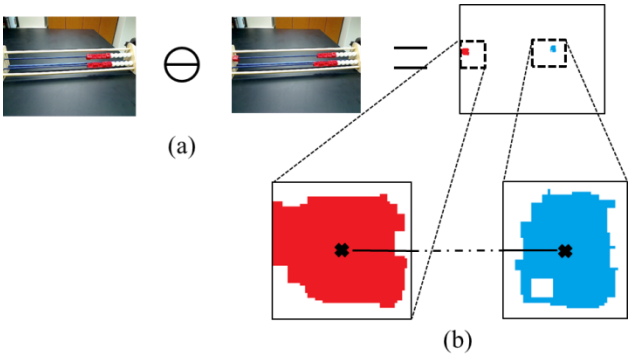

Figure 4. Illustration of the calibration phase. (a) Using frame difference to capture motion area between consecutive state frames. A red bead is moved from right side to left side on the top row, thus there are two regions of motion, which correspond to the initial position and terminal position. (b) Due to view angles difference and perspective effect, the two areas are not exactly the same where the one closer to camera is larger than the other. Centroids of both regions are used to estimate the straight line corresponding to the top row.

Our calibration phase includes four steps, one of each color beads on each row, *i.e.*, 1) red beads on top row, 2) white beads on top row, 3) red beads on bottom row, and 4) white beads on bottom row.

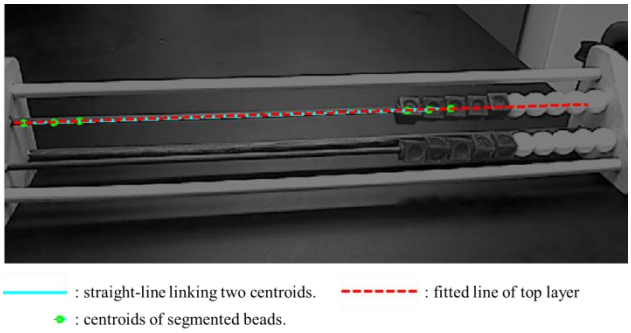

Figure 5. An example result of row information calibration by fitting a straight line. Each pair of centroids (green dots) generated as in Figure 4(b) can be used to fit a straight line in cyan color. The final straight line is generated using the mean of three lines, as can be seen as red dashed line.

In each step, we move one bead on each row once a time from right to left three times and compute the mean area (μ) and corresponding variance (σ^2), as well as the straight line (l) fitted on each row

In addition to the area information which helps to determine how many beads are moved, a straight line (l) is used to determine in which row the beads are moved (as shown in Figure 5). In each pair of state frames, we generate a tuple $[A\ B\ C]^T$ (a straight line is known to be represented using equation $Ax+By+C=0$, which homogenous representation is $[x\ y\ 1]\ [A\ B\ C]^T=0$) to represent a line. Hence after k pairs of calibration, we using the mean tuple of $\{[A_1\ B_1\ C_1],\ldots[A_k\ B_k\ C_k]\}$ as $[\bar{A}\ \bar{B}\ \bar{C}]$.

D. Synchronization

In this section, we describe the details of how we synchronize the state frames and the virtual arithmetic rack. We first describe the transition model between two state frames and then the estimation of each argument in the model respectively.

State Transition. An example of state transition is as shown in Figure 6. We use the term "state transition" to represent how one state frame turns to the next state frame. A state transition can be obtained from two consecutive state frames by estimating three answers to three questions: "which row are the moved beads on?", "how many beads are moved?" and "which direction are they moved to?" The virtual arithmetic rack can thus be synchronized using the estimated answers, as shown in the right of Figure 6: we first move one red bead from right to left on top row and repeat it, thus two state transitions are completed.

Parameters estimation. There are three basic parameters to estimate as shown in the right part of Figure 6, *i.e.*, "which row (top\bottom)", "how many beads (0 to 10)" and "which moving direction (right to left or left to right)". In practice, since it is more convenient to treat red beads and white beads separately, we will estimate "how many beads (0 to 5)" for each color instead of estimating their total number directly. Our tasks are summarized in Figure 7.

Given the motion area by calculating frame difference of two consecutive state frames, we first generate a binary map as shown in Figure 7-left by some processing on hue channel in HSV model to overcome the effect of illumination changes.

First, determining which row the moved bead(s) are on is quite straight-forward. The intuition is the row which is closer to the beads should be the row where the moved beads are on. Since we have obtained two candidate straight line l_{top} and l_{bottom} for both rows (top and bottom)denoted by its homogeneous parameters $[A\ B\ C]$, we can determine the row of the bead(s) by comparing the distances between the centroids of motion area and both candidate straight lines, we can find the nearest line as the estimated row:

$$L = argmin_{\{l\in\{top,bottom\}\}}[x_c, y_c, 1][A_l, B_l, C_l]^T \quad (1)$$

where subfix c denotes the centroid of motion area.

To determine the number of moved beads, as in Figure 7-left, the motion area provides a strong evidence. The intuition is that beads of larger number occupy more area than smaller number of beads. We employ multiple Gaussians to construct the function to determine the number of beads, as shown in Figure 8 and given by:

$$N = argmin_{n\in\{1:5\}} N(a, \mu_n, {\sigma_n}^2) \quad (2)$$

where a is the number of pixels of the motion area and $\{\mu_n, {\sigma_n}^2\}$ is the mean and variance of the n[th] Gaussian represented by $N(a, \mu_n, {\sigma_n}^2)$.

Since in calibration we have generated the mean and variance of area for single bead, $\{\mu_1, {\sigma_1}^2\}$, we can derive $\mu_n = n\,\mu_1$ and ${\sigma_n}^2 = n^2\,\sigma^2$. However, in practice $n^2\,\sigma^2$ is too large and generate too ambiguous distributions. Thus we choose ${\sigma_n}^2 = n\,\sigma^2$ instead, as illustrated in Figure 8, which works well in our experiments. In other words, for a given number of pixels of motion area, we estimate the most possible number of beads corresponds to, while the number goes higher and the certainty of such estimation goes lower.

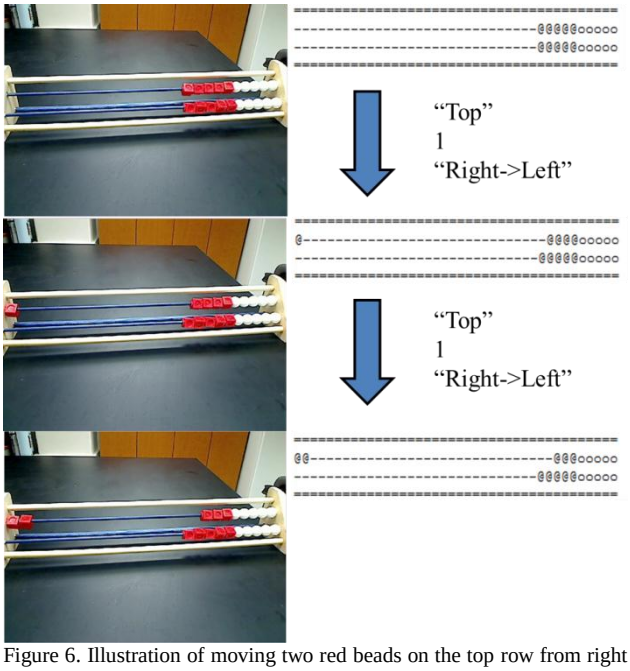

Figure 6. Illustration of moving two red beads on the top row from right to left one at a time. There are three parameters "which row", "how *many beads*" and "*which direction*".

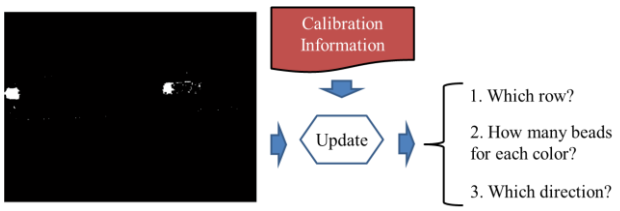

Figure 7: Summary of the tasks in estimation step.

However, the moving direction of the beads cannot be determined from the binary map of motion area. We further employ color information to solve it. Since we have already known the possible color of the beads, either red or white, we apply a purity-based strategy to manifest such information.

The definition of purity is the ratio of red pixels plus white pixels over all pixels, given by:

$$P = \frac{n_{red} + n_{white}}{n_{whole}} \quad (3)$$

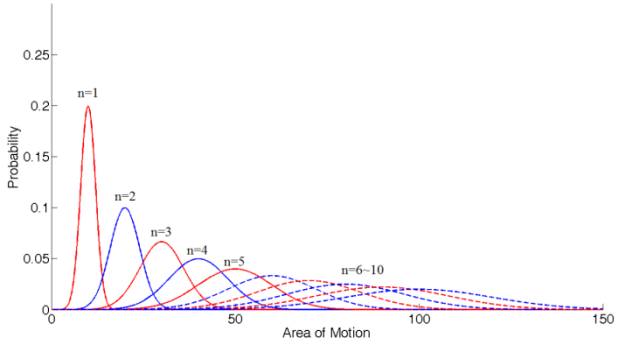

Figure 8. Illustration of probability model we used to determine the number of beads by an array of Gaussian distributions. In this work, we only use an array of 5 Gaussians (solid curves), which gives reasonable distinguish power compared with more Gaussians (dashed lines).

Then by applying a threshold on the purity, we can know which of the two segments (as shown in Figure 7-left) corresponds to the terminal position of the beads. The intuition is that the area of terminal position should be beads so its purity should be higher and the area of initial position should be background so its pixels should not be the same as beads so its purity is lower.

The estimation of the moving direction of beads can be obtained by checking the purity of each motion area. In addition, we can compute the area in number of pixels for the beads of each color occupied. Then we can use equation 2 to estimate the number of beads of each color separately.

Once all the above estimations are finished, we then synchronize the virtual arithmetic rack with the real arithmetic rack by updating the virtual one with all the three arguments we have estimated, as shown in Figure 6-right.

E. State frame capture

Although we only employ state frames to generate updating information, it is easy to create the state frames from an input video with a rule-based algorithm. As illustrated in Figure 9, for each frame of the video sequence, we generate two difference maps (binary map of frame-difference) by compare the frame with both the most recent state frame and previous frame. Condition 1 is that if the current frame has a reasonable difference with current state frame (last static state), which measures if current frame is different enough from last state. Condition 2 is that if current frame has limited difference with previous frame which means if current frame is static. Only if both conditions are met will we update the current state frame and otherwise just continue.

IV. EXPERIMENTS AND DISCUSSION

A. Experiment Setup

In our experiments, we use a Logitech™ web-camera with a Carl Zeiss lens. The real arithmetic rack has two rows of wooden racks with ten plastic red beads in cube shape and ten wooden white beads in sphere shape.

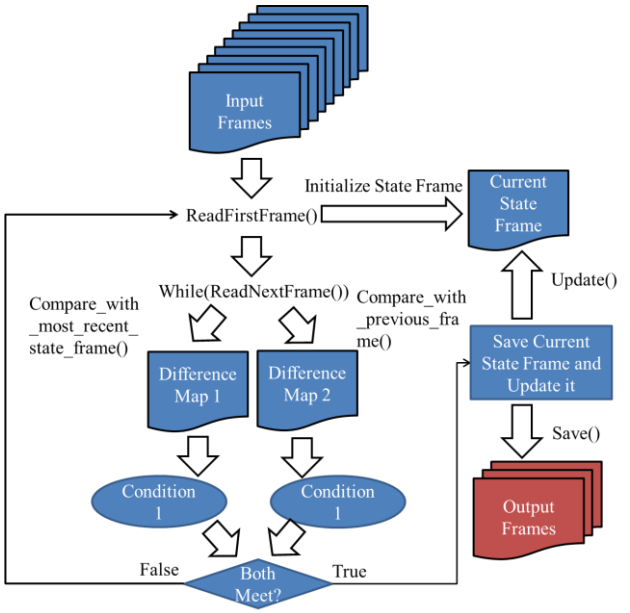

Figure 9. Algorithm flowchart of extracting State Frames (as shown in Red) from input video frames. Condition 1 and Condition 2 are discussed in text.

B. Dataset

We capture four sets of state frames (one for each task as describe in following paragraphs) to evaluate our synchronization framework discussed from Section III.A to Section III.D, one of which is used for calibration and others are for four different tasks. In addition, a video sequence is also captured to test our state-frame extraction framework discussed in Section III.E.

The calibration set contains 4 subsets, each for one color beads (red or white) in one row (top or bottom). Each subset contains three movements, once moving only one bead of corresponding color on corresponding row.

Here we define 4 tasks of different combinations of the movement of beads:

- Task 1 is for only moving red beads on the top row;
- Task 2 is for moving both red and white beads only on the top row;
- Task 3 is for moving only red beads on both top and bottom rows;
- Task 4 is more general for moving beads of both colors on both rows freely.

There is no constraint on the number of moving beads each time. For each task, we capture a video contains 10 movements. Thus there are 40 movements in total of 4 videos for the above 4 tasks. We evaluate our algorithm by compare the detection results with the ground-truth of how many movements are correctly generated. We also

collect a video over 6 minutes to test our state frame extraction algorithm as shown in Figure 9 by moving one bead each time (so by moving each bead twice, we have 20x2=40 state transitions in total).

C. Synchronization results

In our test, there are eleven state frames including the initial one; in total ten state transitions in each task. We define the event (state transition) detection accuracy as the ratio of events correctly labeled over the total number of events. We calculate the event detection accuracy in terms of the three problems (*i.e.,* "which row", "which direction" and "how many beads"), noticing in problem "how many beads", we do not discriminate between colors.

TABLE I: PERFORMANCE OF OUR ALGORITHM ON FOUR TASKS IN TERMS OF EVENT DETECTION ACCURACY (%)

	Row	**Direction**	**# Beads**
Task1	100%	100%	90%
Task2	100%	100%	100%
Task3	100%	100%	90%
Task4	100%	100%	90%
Overall	**100%**	**100%**	**92.5%**

As shown in Table I, our algorithm correctly detects all the events in the tasks of determining "which row" and "which direction". As for determining "how many beads?" there is one failure case in tasks 1, 2, and 4 that the system wrongly classifies "3 red beads" to "2 red beads". Some of the results are as shown in Figure 11.

To validate our frame extraction framework (see Figure 9), we recorded a video with 11,774 frames, where 40 frames (41 including initial frames, consist 0.3% of all frames) of which are state frames. By applying algorithm as shown in Figure 9, we correctly extract all the 40 state frames (both detection recall and precision are 100%), which enable our future work to be extended to a real time prototype. Some examples of the extracted state frames versus the non-state frames are shown in Figure 10.

D. Result discussion and future work

In this paper, we attempt to employ computer vision and image processing technologies to enhance mathematic learning for visually impaired children using an arithmetic rack. Currently, our main assumption is to keep the arithmetic rack relatively static through the processing. This may issue problems when the whole arithmetic rack is dramatically moved handled by visually impaired children. More severe occlusion and jittering will be handled in our future work, which is also a very challenging problem in many computer vision-based applications.

Another potential improvement is to involve context information in this modeling. In our work, we estimate the parameters independently from the difference binary map of two consecutive state frames, *i.e.,* we calculate every probability of each possible parameter set and select the highest one. But in reality the context information may help to discard some logical wrong solutions where their probability may not be that low in our calculation. Consider an example of beads alignment on the top row, there are two red beads on the left and the rest three red beads and five white beads are on the right of the arithmetic rack. If we move six beads from right to left, they must be three red ones plus three white ones. In other words, if we are sure there are three white beads are moved, it should give more confidence to "three red beads are moved" than "two red beads are moved" even though the two probabilities are almost the same, which is common due to beads occlusion and light reflection (as the failure case in Table 2 Task 4).

V. CONCLUSIONS

In this paper, we have proposed an effective computer-vision framework to enhance early mathematics learning for visually impaired children. Preliminary experimental results demonstrate that our framework is effective in determining the row, moving direction and the number of beads of each color with a calibration phase. We have developed a rule-based algorithm to extract state frames from input frame sequence which makes it easy to extend our algorithm and framework to a real time prototype system.

Our future work will focus on developing a more robust and efficient prototype system for mathematics learning enhancement and address the significant human interface issues associated with mathematics learning for children with visual impairments.

ACKNOWLEDGEMENT

This work was supported by NIH 1R21EY020990, DTFH61-12-H-00002, NSF grants IIS-0957016, EFRI-1137172, Microsoft Research, and a CITY SEEDs grant.

REFERENCES

[1] Carey, S. (2008). Math Schemata and the Origins of Number Representations. *Behavioral and Brain Sciences* 31 (6):645-646.

[2] Dehaene, S. (1997). *The number sense: How the mind creates mathematics*. Oxford University Press.

[3] Fosnot, C. & Dolk, M. (2002). *Young Mathematicians at Work: Constructing Fractions, Decimals and Percents.* Portsmouth, NH: Heinemann,

[4] Hasanuzzaman, F., Yang, X and Y. Tian, Robust and Effective Component-based Banknote Recognition by SURF Features, Wireless and Optical Communications Conference (WOCC), 2011.

[5] Rapp, D. & Rapp, A. (1992). A Survey of the Current Status of Visually Impaired Students in Secondary Mathematics. *Journal of Visual Impairment and Blindness, 86, 2,* 115-17.

[6] Science Daily (April 16, 2010). *New Teaching Tools Aid Visually Impaired Students in Learning Math.*

[7] Treffers, A. (1991). Realistic Mathematics Education in the Netherlands, 1980-1990. In L. Streefland (Ed.) *Realistic Mathematics Education.* Utrecht: Utrecht University.
[8] Tian, Y., Yi, C., and Arditi, A.: Improving Computer Vision-Based Indoor Wayfinding for Blind Persons with Context Information, 12th International Conference on Computers Helping People with Special Needs (ICCHP), 2010.
[9] Tian, Y., Yang, X. and Arditi, A.: Computer Vision-Based Door Detection for Accessibility of Unfamiliar Environments to Blind
[10] Persons, 12th International Conference on Computers Helping People with Special Needs (ICCHP), 2010.
[11] Tian, Y. and Yuan, S.: Clothes Matching for Blind and Color Blind People, 12th International Conference on Computers Helping People with Special Needs (ICCHP), 2010.
[12] Wynn, K. (1998). Psychological foundations of number: numerical competence in human infants. *Trends in Cognitive Sciences 2, 296-303.*
[13] Yang, X., Yuan, S. and Tian, Y.: Recognizing Clothes Patterns for Blind People by Confidence Margin based Feature Combination, International Conference on ACM Multimedia, 2011
[14] Yang, X., Tian, Y., Yi, C. and Arditi, A.: Context-based Indoor Object Detection as an Aid to Blind Persons Accessing Unfamiliar Environments, International Conference on ACM Multimedia, 2010.
[15] Yuan, S., Tian, Y. and Arditi, A.: Clothing Matching for Visually Impaired Persons, Technology and Disability 23, Page1-11, 2011. The online version if the article is available at: http://dx.doi.org/10.3233/TAD-2011-0313
[16] Yi, C. and Tian, Y.: Text String Detection from Natural Scenes by Structure-based Partition and Grouping, IEEE Transactions on Image Processing, Vol. 20, Issue 9, 2011. PMID: 21411405.
[17] Yi, C. and Tian, Y.: Text Detection in Natural Scene Images by Stroke Gabor Words, The 11th International Conference on Document Analysis and Recognition (ICDAR), 2011.
[18] Yi, C. and Tian, Y.: Assistive Text Reading from Complex Background for Blind Persons, The 4th International Workshop on Camera-Based Document Analysis and Recognition (CBDAR), 2011

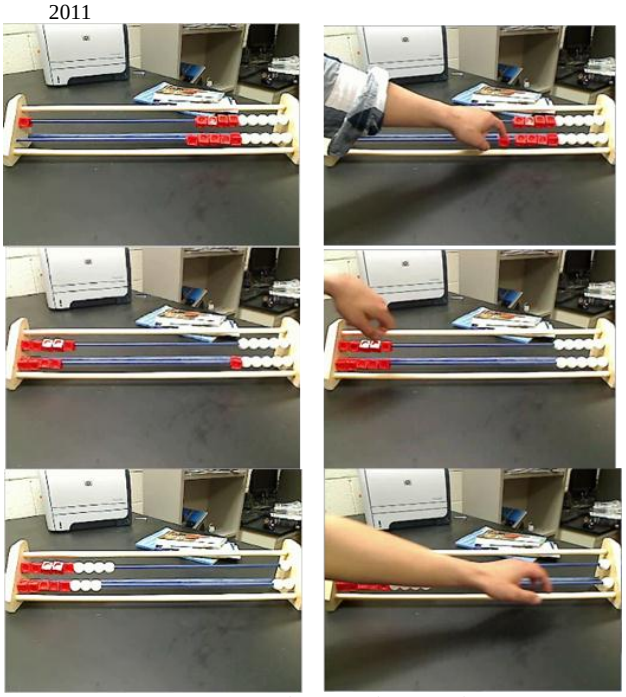

Figure 10. Some examples of the state frames (left column) and non-state frames (right column).

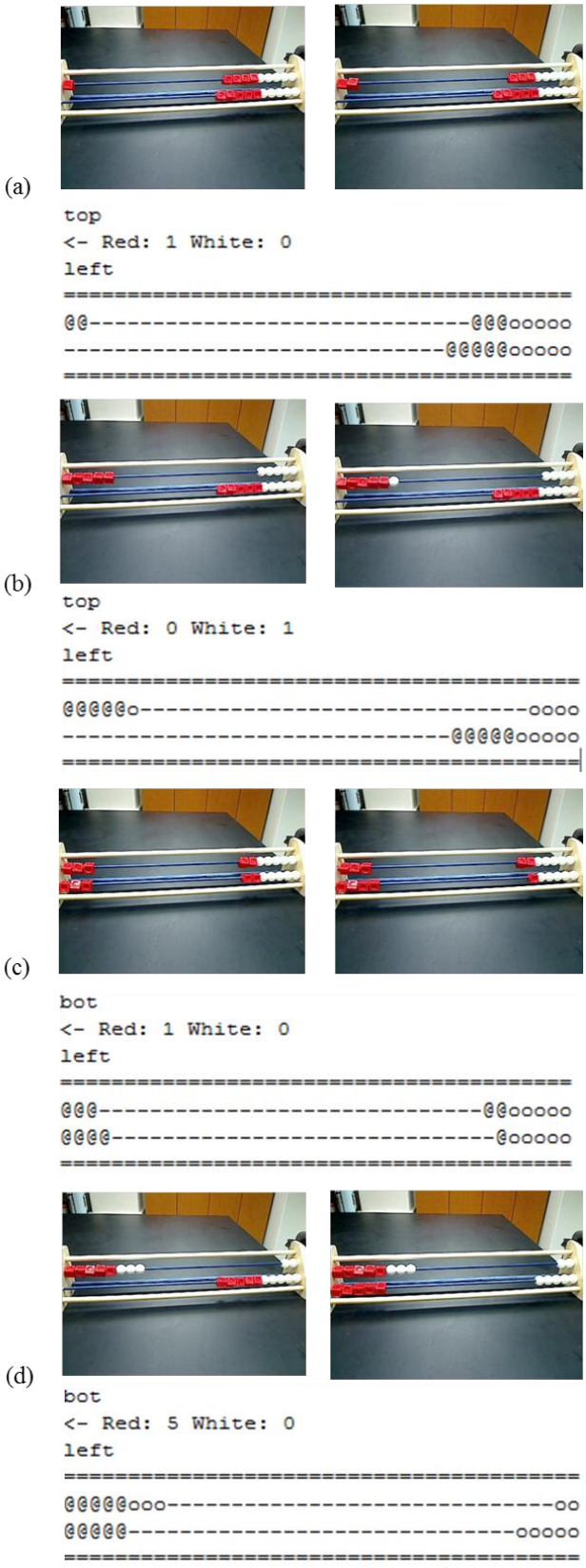

Figure 11. (a) (b) (c) and (d) are sample results for task 1 to 4, respectively. In each task, two images on the top row indicate two consecutive state frames and bottom row is the synchronization result generated from the two state frames showing "which row", "how many beads of" and "which direction".