

書画カメラを用いた珠算行動センシング

松田 裕貴[†]

[†] 奈良先端科学技術大学院大学 〒630-0101 奈良県生駒市高山町 8916-5

E-mail: †yukimat@is.naist.jp

あらまし 算盤（そろばん）を用いた数値計算，すなわち「珠算」は，「珠」を用いた数値表現を理解した上で，複数の方法・順序により「珠」を操作しなければならず，その習得には長期に渡る繰り返し学習を要する。しかしながら，珠算指導は学習者の観察に基づきミスや苦手な珠操作を発見・指導する形式が主となっており，人手に頼るほかないのが現状である。そこで本研究では，一般的な算盤を用いる珠算学習を対象とした ICT による学習支援を実現するため，書画カメラを用いて取得した画像認識に基づく算盤上の入力値推定手法を提案する。評価実験の結果，算盤上に入力された 7 桁の数値に対し正解率 95.0% での推定が可能であることを確認した。

キーワード 珠算，算盤（そろばん），画像認識，学習支援

Abacus Operation Sensing Using Document Camera

Yuki MATSUDA[†]

[†] Nara Institute of Science and Technology 8916-5 Takayama-cho, Ikoma, 630-0101 Japan

E-mail: †yukimat@is.naist.jp

Abstract Numerical calculation using an abacus requires an understanding of numerical expressions using the beads of the abacus and manipulating beads in multiple ways and in different orders. It is usually required a long period of repeated learning to acquire the skill of using the abacus. However, the teaching method of the abacus mainly relied on lecturers' observation through finding errors and poor bead operations and pointing them out, and there is no other way but to rely on human labor at this moment. In this study, we aim to realize an ICT-based learning support system for arithmetic with a common abacus. This paper proposes a method of estimating input values on an abacus based on image recognition captured by a document camera. Through the evaluation experiments, we have confirmed that the proposed method showed an accuracy of 95.0% in the estimation of 7-digit number input on an abacus.

Key words calculation with abacus, abacus, image recognition, learning support

1. はじめに

計算能力を獲得するための方法の一つとして，算盤（そろばん）を用いた数値計算，すなわち「珠算」が現代においても活用されている。算盤は，串で刺した珠を移動させその位置で数を表現することにより，四則演算を含む様々な計算を補助することが可能なツールである。物理的な「珠」の操作と，視覚的な算盤上での数値表現の組み合わせにより，一般的な計算とは異なる方法で計算を実現する。近年において直接的に算盤を使用する場面は減少しているものの，珠算学習を通じて習得できる能力や効果が注目されている [1]~[6]。しかしながら，算盤を用いた計算は，「珠」を用いた数値表現を理解した上で，複数の方法・順序により「珠」を操作しなければならず，その習得には長期に渡る繰り返し学習を要する。

一般的な学校教育・珠算教室における珠算指導では，問題へ

の正答誤答に基づく指導が基本となっている。学習者の状況に応じて，講師により実演をすることで理解を促したり，学習者の観察に基づきミスや苦手な珠操作を発見・指導したりするものの，基本的に全て人手に頼るほかないのが現状である。また，計算ミスや苦手な珠操作（学習者がつまづくポイント）には規則性・パターンがあると考えられるが，その知見は講師の中に暗黙知として蓄積されていることから，他者が再利用することができない。

以上の背景から本研究では，一般的な算盤を用いる珠算学習を対象とし，ICT による学習支援を実現することを目的としている。本稿では，学習支援を行うために必要となる学習者の「珠算行動」を把握するため，書画カメラによって取得した画像認識に基づく算盤上の入力値推定手法を提案する。評価実験の結果，算盤上に入力された 7 桁の数値に対し正解率 95.0% での推定が可能であることを確認した。

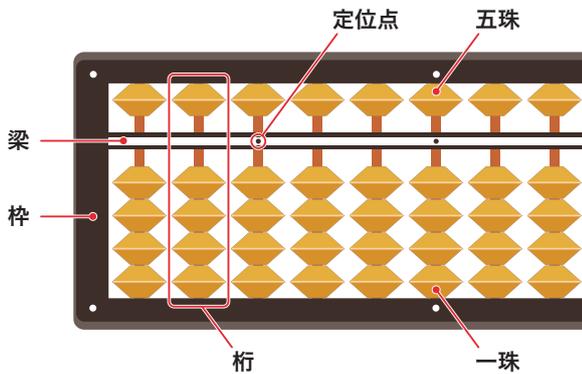


図1 算盤の部位

2. 前提知識・研究の動機

算盤は、串で刺した珠を移動させその位置で数を表現することで、四則演算を含む様々な計算を補助することが可能なツールである。ここでは、算盤の基本的な知識について概説した上で、本研究を実施する動機について述べる。

2.1 算盤での数表現

まず、算盤での数表現について説明する。図1は算盤の部位を示している。「枠」は10進数の「1桁」に対応し、5つの珠の配置によって「0~9」の数を表現することができる。それぞれの「枠」は、「梁」を挟み上段・下段に分かれ、上段の1つの珠（「五珠」）は10進数の「5」、下段の4つの珠（「一珠」）は1つにつき10進数の「1」をそれぞれ表現するために用いられる。具体的な数表現の例を図2に示す。上段・下段ともに珠の高さの半分の空間があいており、珠を上下に動かすことによって各珠の状態を表現する。上段では、珠が上がっている状態がOFF(0)、下がっている状態がON(5)を示し、逆に下段では、珠が上がっている状態がON(1)、下がっている状態がOFF(0)を示す。同一桁の珠の総和（ONの状態となっている珠の表す数値の合計）がその桁の示す数となる。なお、図1の「定位点」は、3桁ごとに梁に示されており、一の位を決める目印として用いられ、一の位として決めた定位点より右側の桁については、小数を表現するために使用される。

2.2 算盤での計算方法

算盤を用いた計算は、5の補数と10の補数の組み合わせを多用することが特徴である。具体的には、次に示す3つの操作が存在する。次では加算を例に上げて操作について紹介するが、減算についても、加算の逆操作を行うことで同様に表現することができる。

(1) 一珠のみ・五珠のみを操作する計算

一珠あるいは五珠のみを移動させることで完結する最も単純な操作である。加算の例を次に示す。まず、 $1+2$ の計算では、1つだけ一珠が上がった状態(1)から、さらに2つの一珠を上げる操作(+2)を行うことで、3の状態を表現することができる。また、 $3+5$ の計算では、3つ珠が上がった状態(3)に対し、五珠を下げる操作(+5)を行うことで、8の状態を表現することができる。

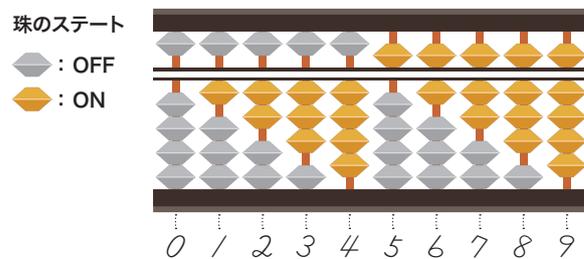


図2 算盤での数表現

(2) 一珠・五珠を操作する計算

一珠と五珠を同時に移動させることが必要となる操作である。具体的には、0~4の数値から5~9の数値への状態変化（またはその逆）が生じる操作に対応する。加算の例を次に示す。 $1+6$ の計算は、(1)の操作を組み合わせ、1つ珠が上がった状態(1)に対して、6（すなわち、5と1）を加えることで、計算結果の7を得ることができる。一方で、 $2+4$ の計算では、2つ珠が上がった状態(2)に対し4を加えるのだが、一珠は一桁につき4つであることから一珠の組み合わせのみではこの計算は表現できない。そこで、5の補数の考え方を導入する。4は $5-1$ と表現できることから、現在の入力値である2から「1を減ずる」「5を足す」という操作を同時に行うことで、 $2+4$ の計算結果である6を得ることができる。

(3) 二桁を操作する計算（繰り上がり・繰り下がり）

二桁に渡っての珠移動を必要とする操作である。具体的には、一桁で表現可能な0~9の数値から10以上の数値への状態変化（またはその逆）、つまり繰り上がり・繰り下がりが生じる操作に対応する。加算の例を次に示す。 $3+8$ の計算では、3つ珠が上がった状態(3)に対し8を加えるのだが、一桁の範囲では表現できないため、10の補数の考え方を導入する。8は $10-2$ と表現できることから、現在の入力値である3から「2を減ずる」「10を足す」という操作を同時に行うことで、 $3+8$ の計算結果である11を得る。さらに、5の補数を同時に考えなければならない場合もある。例えば $6+8$ の計算では、8は上述したとおり10の補数を考え、1の位から2を減じなければならないが、6から2を減ずるには5の補数($5-3$)を考える必要がある。つまり、 $6+8$ の計算結果である14を得るためには「3を足す」「5を減ずる」「10を足す」という3つの操作を同時に行うことが必要となる。

2.3 珠算学習支援の実現に向けた課題

算盤を用いた計算は、前述の通り5の補数と10の補数を常に考えながら操作を行わなければならない、一般的な計算とは大きく異なる。初学者が珠算能力を獲得するためには反復的かつ継続的な珠算学習を要するとされる。しかしながら、珠算指導は学習者の観察に基づきミスや苦手な珠操作を発見・指導する形式が主となっており、人手に頼るほかないのが現状である。このことから、学習者による算盤の操作状況（珠算行動）のセンシングに基づき効果的な指導を提供することは、珠算学習の効率を向上させる上で必要と考えられる。

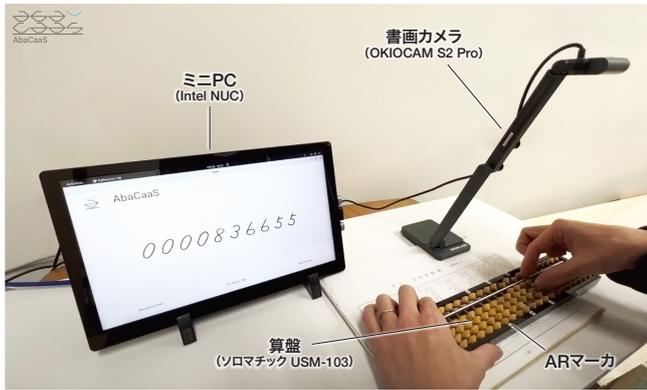


図3 AbaCaaS のシステム構成

3. 関連研究・サービス

北川らは、カメラを用いた盤面推定システムおよび珠の上に映像を重畳表示することで操作方法等を伝えるプロジェクションマッピングシステムからなる珠算学習支援システムを提案している [7]. 盤面の推定には、透明な機の裏側（下部）から算盤を RGB カメラで撮影し、算盤の珠の形状によって生じる明暗差を利用してオブジェクト検出する方法を用いることで、オクルージョンが生じない状況での推定を実現している. なお、明暗差を強調するために、カメラと同様に機の裏側に LED ライトを設置する必要がある.

新川らは、珠算教育向け Learning Management System (LMS) を提案しており、フラッシュ暗算、読上算、見取り算などといった学習ソフトを LMS 上で管理し、また個人成績や学習の進捗状況と組み合わせることによって、ソフトウェアを搭載した PC であれば場所を問わず学ぶことを可能とした [8].

スマートフォンやタブレット端末の画面上に算盤を再現することで、珠算学習を支援するアプローチも存在する. 齋藤らはスマートフォン上の学習支援システムのプラグインの一つとして電子算盤の機能を提案している [9]. 画面上に算盤を再現することで基本的な算盤の操作ができ、また電子算盤を用いて計算した過程を数式として表示することも可能となっている. Digika 社は、算盤の UI・操作方法を応用した暗算学習指導を提供する「そろタッチ」というサービスを提供している [10]. タブレット端末上に算盤に類似したインターフェースを再現し、珠に対応するボタンを両手で操作することによって計算する方式を採用している.

4. 提案システム・AbaCaaS

市販の算盤を用いた一般的な珠算学習を対象とし、書画カメラを用いることで珠算行動のセンシングを実現するシステム、AbaCaaS (Abacus Operation Sensing Using Camera as a Sensor) を提案する. 本稿では、珠算行動センシングの中でも特に重要と考えられる「算盤の盤面上にどのような数値を入力しているのか?」という点に着目し、AbaCaaS システムを用いた算盤の入力値推定手法を提案する.

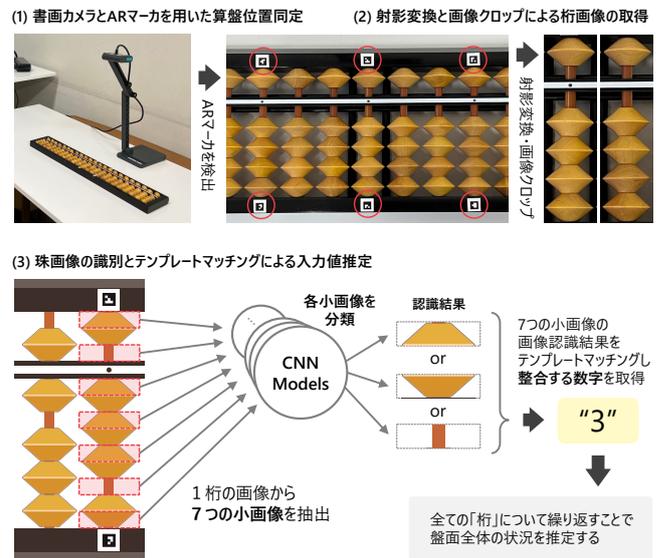


図4 AbaCaaS の入力値推定ワークフロー

4.1 システム構成

システム構成を図3に示す. AbaCaaS は、AR マーカを貼付した市販の算盤と、算盤の盤面を俯瞰的な視点から撮像する書画カメラ、そして画像処理・珠算行動認識処理を実行するミニ PC から構成される. 書画カメラは机と並行した角度で直下を撮像するように配置することを想定するが、算盤との位置関係は算盤が画角に収まる範囲内であれば自由に設定することができる (算盤を固定する必要はない) ものとする.

本論文においては、書画カメラは OKIOLAB S2 Pro (OKIOLABS 社製、解像度：1920×1080px)¹⁾、算盤はソロマチック USM-103 (雲州堂社製、23 桁ツゲ玉)²⁾、ミニ PC は NUC8i5BEH (Intel 社製、CPU：Intel Core i5-8259U、RAM：16GB、GPU：Intel Iris Plus Graphics 655、OS：Ubuntu 22.04.1 LTS)³⁾を使用する. なお、システムのソフトウェア開発環境に関する詳細な情報については付録 1. に示すとおりである.

4.2 珠算行動センシング手法

ここでは、書画カメラから得られる映像データを用いて算盤領域を検出し盤面上に入力されている数値を推定する手法について述べる. センシングの手順を図4および以下に示す.

- (1) 書画カメラと AR マーカを用いた算盤位置同定
- (2) 射影変換と画像クロップによる桁画像の取得
- (3) 珠画像の識別とテンプレートマッチングによる入力値推定

以降では、各手順について詳述する.

4.2.1 書画カメラと AR マーカを用いた算盤位置同定

通常、算盤は計算を行う過程で移動することから、書画カメラから得られた映像フレームから算盤領域を検出しなければならない. しかしながら、算盤による計算操作を妨げないようにするため、算盤の加工は最低限に留める必要がある. そこで、

(注1) : <https://www.okiolabs.com/product/okiocam-s2-pro/>

(注2) : <https://ec.unshudo.co.jp/ソロマチック 23 桁ツゲ玉-7/>

(注3) : <https://www.intel.co.jp/content/www/jp/ja/products/sku/126148/intel-nuc-kit-nuc8i5beh/specifications.html>

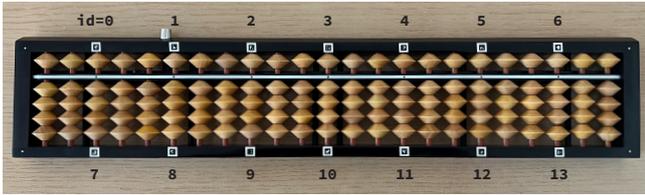


図5 AR マーカの貼付

算盤の枠に貼付したAR マーカ [11] を映像フレームから検出することで位置同定を行う方法を採用する。

AR マーカを貼付した算盤の外観を図5に示す。OpenCVのAR マーカ生成・検出モジュールであるArUco [12] を用い生成したIDの異なるAR マーカ(辞書サイズ:3×3, 印刷サイズ:5×5 mm)を, 算盤の各定位点の上部・下部の枠にそれぞれ7枚ずつ貼付した。なお, 市販されている一般的な算盤は珠の大きさや形状は概ね同一となっていることから, 他の算盤を使用することも可能である。

4.2.2 射影変換と画像クロップによる桁画像の取得

次に, 検出されたAR マーカから算盤の領域画像を抽出する。算盤を使用する手によるオクルージョンや光の反射などの影響により, 必ずしも全てのAR マーカが検出されるとは限らないため, まず検出されたAR マーカの位置関係から欠損しているAR マーカの位置を補完する。これらのAR マーカで囲われた領域が算盤領域となるが, 画像内における角度や位置が異なるため, 射影変換を行うことによって正位置からの画像に変換する。得られた画像を分割・クロップすることにより, 桁画像を取得する。

4.2.3 珠画像の識別とテンプレートマッチングによる入力値推定

最後に, 得られた桁画像を認識することで, 各桁の入力値を推定する。ここでは, 算盤の構造が持つ特性に着目し, 解くべき課題を単純化することにより高性能な入力値推定を実現する。

前述の通り, 算盤では1桁あたり上段・下段合わせて5つの珠を上下に移動させることで数値を表現するが, 珠を動かすスペースはおおよそ珠ひとつの半分の高さとなっている。つまり, 図4左下に示すように, 1桁の画像から7つの小画像を取得することが可能である。各小画像は「珠の上半分」「珠の下半分」「軸」のいずれかに必ず該当し, かつ7つの小画像の順列は入力される数値ごとにユニークである。

そこで, 7つの各小画像を上記の3クラスに分類する画像認識モデルを構築するとともに, 得られた認識結果の順列と, 事前に準備した0~9の数値それぞれの小画像順列の辞書とのテンプレートマッチングを行うことで, 対象となる桁についての入力値推定結果を取得する。この手続きをすべての桁について繰り返すことによって算盤に入力されている数値を推定する。なお, 本論文で構築した画像認識モデルは3層からなるCNNモデルであり, 独自に作成した合計4,845サンプル(上:1,925サンプル, 下:1,540サンプル, 軸:1,386サンプル)の学習データセットのみを使用してモデルをトレーニングした。トレーニ



図6 評価用データセットの例

表1 算盤の入力値推定モデルの評価結果

項目	値
評価用データセットサンプル数	300
正解サンプル数	285
誤認識サンプル数	15
射影変換の失敗サンプル数	11
小画像認識の失敗サンプル数	4
全サンプルにおけるAccuracy (正解率)	95.0%
射影変換の失敗サンプルを除いた場合のAccuracy (正解率)	98.6%
平均推論時間	0.176 sec

ングには, TensorFlow⁴およびそのライブラリのKerasを使用した。また, 構築したモデルは軽量化のため, TensorFlow Lite Converterを使用し整数量子化(int8)するとともにTensorFlow Lite形式に変換した。

5. 評価実験

提案する手法の性能を評価するため, 構築した算盤の入力値推定モデルを用いた実験を行った。実際の利用ケースを想定し, ランダムに生成した7桁の数値(0000000~9999999)を算盤で入力し, 書画カメラで画像を取得することで, 評価用データセットを作成した。なお, このデータセット内の画像は, 学習用データセットには一切含まれていない。評価用データセットの例を図6に示す。これらの画像は, 算盤に貼付したAR マーカが少なくとも上段・下段についてそれぞれ2つ以上画像内に含まれていること, 推定対象となる算盤の入力部分が画像内に全て含まれていること, 手によるオクルージョンが発生していない画像であることを条件とするが, 算盤の角度や位置は任意に設定できるものとした。なお, 入力する数値は図5に示すid=4, 11の定位点を1の位に設定した。

評価結果を表1に示す。評価用データセットの全300サンプルのうち, 285サンプルについて7桁の入力数値推定に成功しており, 非常に高い性能(Accuracy:95.0%)での推定が可能であることを確認した。推定に失敗した15サンプルについて内訳を確認したところ, 11サンプルについてはAR マーカの

(注4): <https://www.tensorflow.org/>

表2 小画像の認識に失敗したサンプル

入力数値	正解の数値・順列	小画像の分類結果	推定結果
0504 ¹⁵²	1 上軸 上軸下下	上軸 上軸下軸下	X
	2 上軸 上上軸下	上軸 上上軸軸下	X
0159 ⁷⁸¹	9 軸下 上上上軸	軸下 上上下下軸	X
	7 軸下 上上軸下	軸下 上下軸下	X
	8 軸下 上上上軸下	軸下 上下上軸下	X
0156 ⁷⁹⁷	7 軸下 上上軸下	軸下 上上軸軸下	X
0937 ³⁹⁰	0 上軸 軸下下	上軸 軸下下軸下	X

誤検知に起因して手順 (2) の射影変換に失敗していることがわかった。本提案システムにおいては、AR マーカの検知に対し、検出できなかった AR マーカの補完処理を行うのみにとどまっていたため、今後 AR マーカの誤検知に関するフィルタリング処理を追加することでこの問題への対処が可能であると考え (なお、射影変換の失敗サンプルを除外した場合は、Accuracy は 98.6% である)。また、残りの 4 サンプルについては、表 2 に示すように小画像の分類に失敗しているものの、テンプレートマッチングで整合する数値が「無い (X)」という出力が得られている。なお、小画像の分類失敗の要因については発生条件・規則性などを見出すことはできなかった。

以上の結果について、複数の観点から考察する。まず、小画像の認識性能という観点からは、14,700 件の小画像 (= 300 サンプル × 7 桁 × 7 小画像) のうち 14,692 件の認識に成功している (Accuracy : 99.9%) こと、テンプレートマッチングによって整合する数値が存在しない (X) という出力を得られていることから、時間的に隣接するフレームを認識し結果を統合することで、入力数値の誤推定を排除することが可能であると考えられる。連続的な認識という観点からは、平均推論時間 (評価対象の 7 桁を含む 19 桁の入力値推定の所要時間) は表 1 に示すとおり 0.176 sec となっていることから、今回使用したミニ PC の性能では約 5~6 fps 程度での継続的な推論が可能といえる。初学者の算盤の入力スピードを想定するならば、より安価で低い性能のデバイス (たとえば Raspberry Pi⁵ など) を利用できる可能性がある。また、算盤の画像内での位置が与える影響という観点について、画像内の算盤の中心位置とその推定結果を図 7 に示す。画像の中心から外れるほど、珠の小画像が正面から得たものとは異なるため、誤推定が発生しやすい可能性が示唆された。また、今回の評価実験では対象外とした、珠を操作する手によるオクルージョンについても考慮する必要がある。書画カメラは、算盤を使用する人の正面に設置され、使用者の視界に近い映像が取得できる想定される。珠を操作する過程で、手は左から右へと徐々に移動していくことから、最も右の桁を入力した後に全ての桁についての情報を得られると考えられる。

6. まとめ

珠算学習は、算盤の「珠」を用いた数値表現の理解、操作方法の習得が必要であり、長期に渡る繰り返し学習を要することから、本研究では、一般的な算盤を用いる珠算学習を対象とし

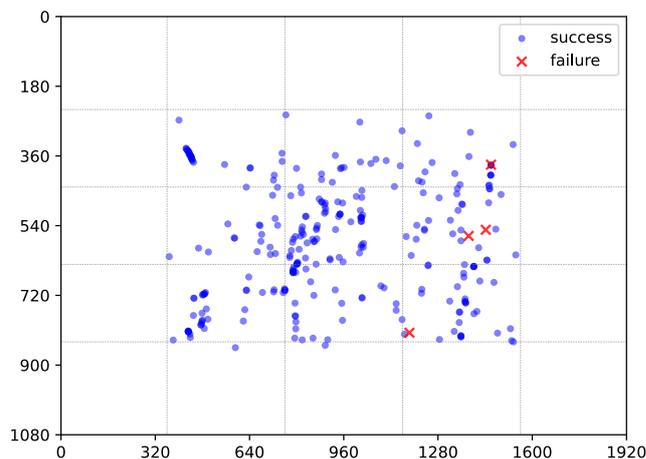


図7 算盤の中心位置と推定結果の正誤

た ICT による学習支援を実現を目指している。本稿では、学習支援を行うために必要となる学習者の「珠算行動」を把握するため、書画カメラによって取得した画像認識に基づく算盤上の入力値推定手法を提案した。AR マーカを貼付した算盤を画像から検出するとともに、桁を構成する珠画像をそれぞれ画像認識し、得られたテンプレートマッチングすることによって、算盤上に入力された 7 桁の数値に対し正解率 95.0% での推定が可能であることを確認した。今後は、操作する手によるオクルージョンが生じる状況下における継続的な認識手法へと拡張するとともに、珠算行動に合わせた適切な指導を行う情報提示システムの開発を行う。

謝辞

本研究の一部は、KDDI 財団調査研究助成 (2021 年度) の助成を受けて行われたものです。

文献

- [1] S. Amaiwa and G. Hatano, "Effects of abacus learning on 3rd-graders' performance in paper-and-pencil tests of calculation," *Japanese Psychological Research*, vol.31, pp.161-168, 1989.
- [2] S. Amaiwa, "The Effects of Abacus Learning on Solving Arithmetic Problems: A Comparative Study of Elementary / Junior High School Students at Upper Level and Inexperienced Students," *Journal of the Faculty of Education, Shinshu University*, vol.96, pp.145-156, 1999.
- [3] C. Wang, T. Xu, F. Geng, Y. Hu, Y. Wang, H. Liu, and F. Chen, "Training on Abacus-Based Mental Calculation Enhances Visuospatial Working Memory in Children," *Journal of Neuroscience*, vol.39, no.33, pp.6439-6448, 2019.
- [4] C. Wang, "A Review of the Effects of Abacus Training on Cognitive Functions and Neural Systems in Humans," *Frontiers in Neuroscience*, vol.14, no.913, pp.1-12, 2020.
- [5] Y. Lu, M. Li, Z. Cui, L. Wang, Y. Hu, and X. Zhou, "Transfer Effects of Abacus Training on Cognition," *Current Psychology*, vol.42, pp.6271-6286, 2023.
- [6] Y. Hu, F. Geng, L. Tao, N. Hu, F. Du, K. Fu, and F. Chen, "Enhanced White Matter Tracts Integrity in Children With Abacus Training," *Human Brain Mapping*, vol.32, no.1, pp.10-21, 2011.
- [7] 北川珠莉, 鈴木優, "珠の位置認識と操作手順の重畳表示を用いたそろばん学習支援システム," *情報処理学会インタラクシオン 2022*, pp.759-762, 2022.
- [8] 新川晃司, 川崎健志, 澤田一樹, 二石芳裕, 箕宗徳, 渡邊一衛, "そろばん教育における Web 学習支援システムの開発," *成蹊大学理工学研究報告*, 第 48 巻, pp.75-79, 2011.

(注5) : <https://www.raspberrypi.org/>

- [9] 齋藤謙太, 佐々木整, 水野一徳, “携帯電話を利用した学習支援ツールの開発,” 情報科学技術フォーラム講演論文集, 第8巻, pp.653–654, 2009.
- [10] 株式会社 Digika, “そろタッチ,” <https://www.sorotouch.jp/>. (accessed 2023-04-10).
- [11] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” Pattern Recognition, vol.47, no.6, pp.2280–2292, 2014.
- [12] OpenCV, “ArUco markers and boards detection for robust camera pose estimation,” https://docs.opencv.org/4.7.0/de/d67/group__objdetect__aruco.html. (accessed 2023-04-10).

付 録

1. 開発環境

本論文で提案したシステムは, Python (バージョン: 3.10.6) を用いて実装されている. 使用した主要な Python ライブラリは次に示すとおりである.

```
numpy==1.23.5
opencv-python=4.6.0.66
opencv-contrib-python=4.6.0.66
Pillow==9.3.0
keras==2.11.0
tensorboard==2.11.0
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
tensorflow==2.11.0
tensorflow-estimator==2.11.0
tensorflow-io-gcs-filesystem==0.29.0
```