

立体数独アプリケーションの開発

田中 貴拓¹⁾(非会員) 新谷 幹夫²⁾³⁾(非会員) 岩穴口 貴祥²⁾(非会員) 白石 路雄²⁾(正会員)

1) 株式会社ユニスティ 〒105-0012 東京都港区芝大門 1-9-9 野村不動産芝大門ビル 6階 2) 東邦
大学理学部 〒274-8510 千葉県船橋市三山 2-2-1 3) UEI リサーチ 〒113-0034 東京都文京区湯
島 3-1-3 MS ビル

Development of a 3D Sudoku Application

Takahiro TANAKA¹⁾ Mikio SHINYA²⁾³⁾ Takahiro
IWAANAGUCHI²⁾ Michio SHIRAISHI²⁾

1) UNISTY CO.,LTD., Nomura Real Estate Development SHIBA-DAIMON BUILDING 6F,
9-9 Shiba-Daimon 1-chome, Minato-ku, Tokyo 105-0012, Japan 2) Faculty of science, Toho
University, 2-2-1 Miyama, Funabashi-shi, Chiba, 274-8510 Japan 3) UEI Research, MS Bldg.
3-1-3 Yushima, Bunkyo-ku, Tokyo, 113-0034 Japan

{shinya, shira}@is.sci.toho-u.ac.jp

概要

近年、立体パズルの開発が盛んに行われ、ゲーム・携帯アプリの1分野として定着している。数独の立体化も試みが見られるが、成功しているとは言えない。本研究では、従来の立体数独アプリケーションの問題点である操作性と見づらさを改善するため、立体数独を解く上で必要な情報を平面的に表示するインターフェースを提案する。提案手法では解きたいマスをクリックするだけで必要な情報が提示される。これによりプレイヤーは煩雑な3次元操作から解放される。比較実験を行なったところ、解答速度が従来手法に対して有意に速いことが示された。余計な操作を省略し、立体数独の魅力の向上に寄与することができる。と期待する。

Abstract

3D puzzles have been recognized as a popular type of computer games. The Sudoku is a popular puzzle game and has been attempted to extend into 3D Sudoku but not commercially successful yet. We analyzed that complicated 3D operations required in current 3D Sudoku applications largely discourage players. In this study, we proposed a simple interface for the 3D Sudoku that displays necessary information to solve the puzzle with a single mouse click. The experimental results show a significant improvement on the play speed over a current game application.

1 はじめに

近年、立体パズルの開発が盛んに行われ、ゲーム・携帯アプリの1分野として定着している。たとえば2Dのパズルであるピクロスが立体化され成功している [1]。一方、数独 [2] の立体化も試みが見られるが、成功しているとは言えない。

通常の数独では縦9マス×横9マスの平面について解いていくが、立体数独は縦9マス×横9マス×奥行き9マスの立方体について解いていく。立体的に存在するマスを立体的に表示する従来の立体数独アプリケーション RealSudoku3D [3] では、解く際にマスが重なって見づらいことや、視点の操作およびマスの一時的な非表示などを必要としていて操作性が悪いことが問題であった。

本研究では操作性や見づらさを改善するため、立体数独を解く上で必要な情報(以下立体数独の基本情報)を平面的に表示するインターフェースを提案する。解きたいマスをクリックするだけで基本情報を平面的に提示する。これにより、プレイヤーは煩雑な3次元操作から解放され、余計な手間をかけることなく立体数独を解くことができる。

2 数独

数独は人気の高いペンシルパズルであり、雑誌や小冊子、携帯アプリなど様々な形で老若男女に親しまれている。数独では9マス×9マスの81マスを以下のルールに従って埋めていく。図1に示すように、数独の解答の例とルールは以下のようである。

- 全てのマスを1~9の数字で埋める (rule 1)。
- 縦一列の9マスには1~9がそれぞれ一度だけ使われる (rule 2)。
- 横一行の9マスには1~9がそれぞれ一度だけ使われる (rule 3)。
- 太線で囲われた3×3のエリアの9マスには1~9がそれぞれ一度だけ使われる (rule 4)。

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 9 | 5 | 3 | 4 | 2 | 6 | 8 | 7 |
| 6 | 3 | 2 | 1 | 8 | 7 | 9 | 4 | 5 |
| 8 | 7 | 4 | 5 | 9 | 6 | 1 | 3 | 2 |
| 5 | 6 | 8 | 4 | 2 | 1 | 3 | 7 | 9 |
| 4 | 1 | 9 | 7 | 6 | 3 | 5 | 2 | 8 |
| 3 | 2 | 7 | 9 | 5 | 8 | 4 | 1 | 6 |
| 2 | 4 | 1 | 6 | 7 | 5 | 8 | 9 | 3 |
| 9 | 8 | 6 | 2 | 3 | 4 | 7 | 5 | 1 |
| 7 | 5 | 3 | 8 | 1 | 9 | 2 | 6 | 4 |

図1 数独のルール

3 立体数独

本研究で扱う立体数独では、縦9マス×横9マス×奥行き9マスに関して、任意のスライスが数独の条件を満たすように数字で埋めていく。通常の数独のように紙媒体でのプレイはできないため、アプリケーションの開発が必要となる。図2に立体数独アプリケーション RealSudoku3D の実行画面を示す。



図2 9×9×9マスの立体数独 (RealSudoku3D 画面)

プレイヤーはいずれかのスライスに着目して通常の数独のように解いていくが、ある数字が入るマスの候補が複数あった時に、そのスライス以外の情報を参照することで候補を絞ることができる場合がある。この時に参照すべき情報は、候補のマスを含む他の方向のスライスにある情報である。具体的には、すなわち選択したマスを含む奥行き方向の行および表示スライスと異なる2方向のエリアの情報が選択したマスに入る数字を限定する情報が必要となる。すなわち、立体数独の基本情報は以下

の3つとなる．

- 着目するスライスの情報 (①) ．
- 選択したマスの奥行き方向の情報 (②) ．
- 選択したマスを含むブロックの情報 (③) ．

このうち③のブロックの情報は前述の2方向のエリアの情報を含んでいる．これらの基本情報を簡単な操作で提示することが立体数独では重要となる．

4 従来手法

従来の立体数独アプリケーションには、立体の表面のみを対象としたもの [4, 5] や一辺4マスのもの [6] はあるものの、 $9 \times 9 \times 9$ の立方体を対象としたものは RealSudoku3D のみである．そこで、RealSudoku3D を従来手法として比較対象とする．

RealSudoku3D は半透明のマス縦9マス×横9マス×奥行き9マスの配置で描画し、その中に数字のオブジェクトを表示している．クリックにより最も手前に見えているマスを選択し、画面下部の1~9の入力ボタンで数字を入力する．

奥にあるマスを見たり選択したりするために、表示切替ボタンによりスライスの表示/非表示を切り替える．図中の S,L,F はそれぞれスライスの方向を示している．F 方向の2番~9番のスライスをすべて非表示とした結果を図3に示す．表示切替ボタンを操作するたびに、そのボタンに対応するスライスの全てのマスが表示もしくは非表示になる．また、図中の矢印は S,L,F の各方向を示している．全体の回転や平行移動、ズームなどの操作はマウス操作で行なう．

RealSudoku3D で基本情報①~③を全て読み取るうとしたとき、表示の切り替え操作と視点の調整が複雑なため、パズルを解く際にプレイヤーに余計な手間がかかってしまう．

図4において、(a) は着目しているスライスおよび奥行き方向を見たいマスを示している．この奥行き方向を見るために表示切替ボタンで点線で示したスライスを表示する ((a) (b)) ．しかしそれだけでは、元々見ていたスライスと重なってしまい奥行き方向の情報が見られないため、ドラッグして全体を回転し、(c) のようにする ((b) (c)) ．奥行き方向の情報を確認することができたなら、ドラッグして回転を元に戻し ((c) (b))、表示切替

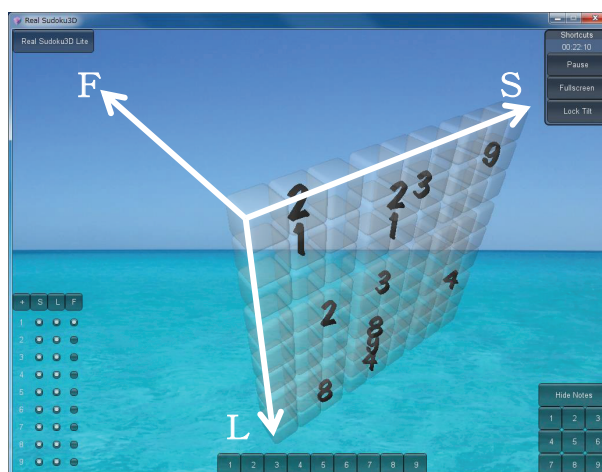


図3 表示切替とスライスの方向

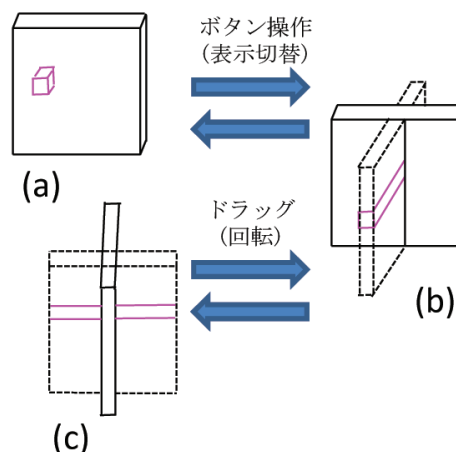


図4 RealSudoku3D の操作例

ボタンでスライスの表示も元に戻す ((b) (a)) ことで、パズルを解くことを続けることができる．

このように煩雑な操作が必要となり、プレイヤーがゲームに集中することが阻害されてしまう．基本情報を表示するこれらの操作を簡単に行なえるインターフェースが必要である．

5 提案手法

5.1 設計方針

本研究の目的は操作性と見づらさを改善した立体数独アプリケーションの開発である．そのために、3章で述べた基本情報①, ②, ③を全て簡単な操作で提示できるよう設計する必要がある．本アプリケーションでは情報を平面的に表示する．これにより、視点の調整操作を不要

とする。

ボタン操作で着目するスライスの方向と番号を選択し、またマスをクリックすることでそのマスを選択する。これらの操作によって表示する基本情報を切り替える。

また、情報がワンクリックで切り替わるため、プレイヤーは自分がどこを見ているのか分からなくなる可能性がある。それを防ぐために全体図を表示する。

5.2 画面構成

前述の設計方針に基づきアプリケーションを作成した。アプリケーションの開発は統合開発環境 Unity[7]で行った。図5に作成したアプリケーションの画面を示す。

基本情報①, ②, ③をそれぞれ main view(画面中央), depth view(画面上部), block view(画面左部) 内に表示する。また全体図を 3D view(画面右部) 内に表示する。なお、各ビューは異なる背景色で塗られている。

操作はマスの選択とボタンで行われる。ボタンには、(a) 選択中のマスに数字を入力/削除するための 1~9, delete キーからなる入力ボタン, (b) 着目するスライスの番号を変えるための forward および backward キーからなるスライス番号変更ボタン, (c) 着目するスライスの方向を変えるための x, y, z からなるスライス方向変更ボタン, がある。マスの選択は main view 内のマスをクリックすることでできる。

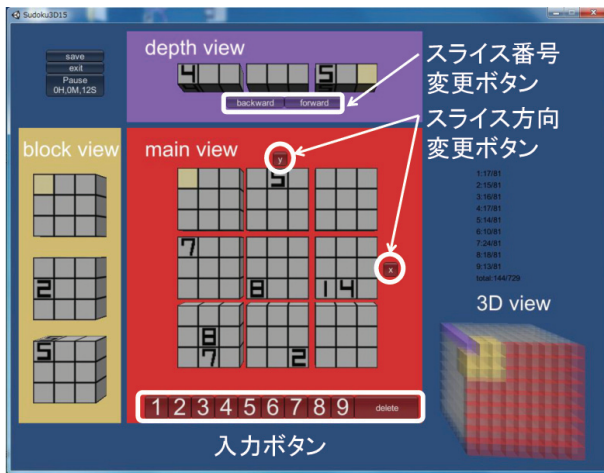


図5 画面構成

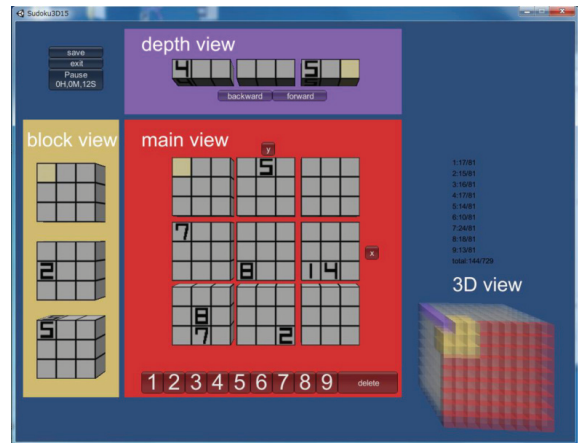
main view では基本情報①, すなわちプレイヤーが着目するスライスを表示している。また、クリックによりマスを選択することができる。main view 下部には入力

ボタンがある。また、スライス方向変更ボタンも main view 内に表示される。着目するスライス方向に応じて、他の2方向への変更ボタンが表示される。着目するスライス方向と変更後の方向の関係に応じて、変更後にカメラが移動する方向にボタンが配置される。

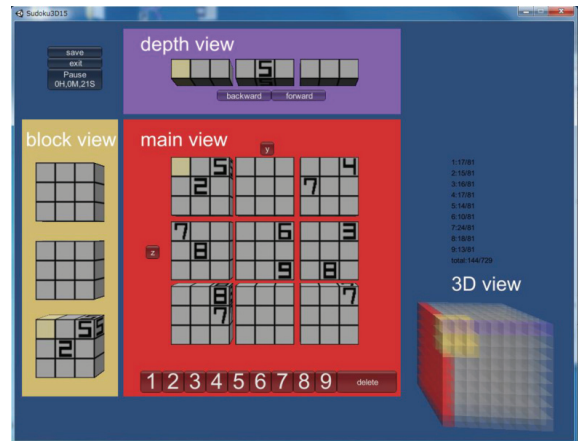
depth view では基本情報②, すなわちプレイヤーが選択中のマスの奥行き方向の情報を表示している。depth view 下部にはスライス番号変更ボタンがある。

block view では基本情報③, すなわちプレイヤーが選択しているマスを含むブロックの情報, を表示している。着目するスライスと同じ方向にスライスし、手前の中から順に上から表示している。

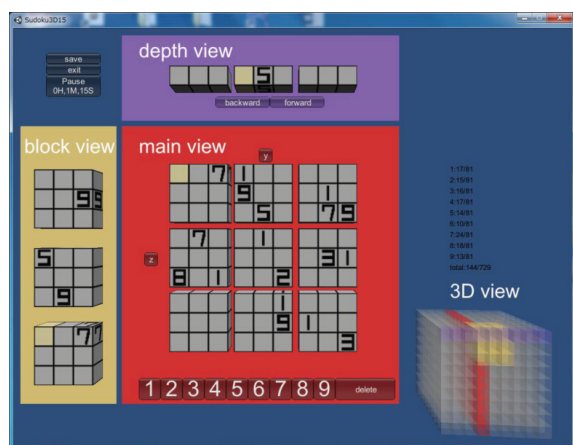
3D view では $9 \times 9 \times 9$ マスの絶対位置を表示し、その向きは変わらない。着目しているスライス, 行, ブロックの絶対位置を提示するために、それぞれに含まれるマスを main view, depth view, block view の背景色で描画する。



(a)



(b)



(c)

図 6 スライス方向/番号変更と画面の遷移

5.3 基本情報の提示

提案手法では着目するスライスとマスを選択するだけで、それぞれの view に基本情報が表示される。

以下、典型的な操作例における各 view の画面遷移の例を示す。尚、マスの位置を 1 から 9 までの整数座標 $(1,1,1)$ から $(9,9,9)$ により表すこととする。また、「*」により 1 から 9 の任意の自然数を表す。

5.3.1 スライス方向の選択

図 6(a) では、最前面左上 $(1,1,1)$ のマスが選択され、 z 方向（正面）のスライスが指定されている。したがって、main view では最前面正面 ($z=1$ 面, $(*,*,1)$)、depth view では奥行方向 $(1,1,*)$ の列、block view では $(1,1,1)$ から $(3,3,3)$ のブロックに対する z 方向スライスが 3 枚表示されている。また、3D view では、他の view で表示されているスライス・奥行・ブロックを各 view の色（赤、青、黄色）で表示している。さらに各 view において、注目マスは黄色で表示されている。

ここで、スライス方向変更ボタンを押し、 x 方向（左側）のスライスを指定する。各 view の表示は図 6(b) に示すよう遷移する。main view では右側面 $(1,*,*)$ 、depth view では $(*,1,1)$ 列、block view では $(1,1,1)$ から $(3,3,3)$ のブロックの x 方向スライスが表示される。またこれに伴い、3D view における着色コマも変化する。

5.3.2 スライス面の移動

図 6(b) の状態から forward ボタンを 3 回押し、 $x=4$ のスライスに移動する。main view では右側面 $(4,*,*)$ 、depth view では $(*,1,1)$ 列、block view では $(4,1,1)$ から $(6,3,3)$ のブロックの x 方向スライスが表示される。

また、注目マスは $(4,1,1)$ に移動し、各 view における注目マス表示も移動している。

以上のように、従来法においては 4 章の図 4 で示したような煩雑な操作が必要であったが、提案手法ではワンクリックで基本情報が提示される。これにより、プレイヤーは簡単かつ少ない回数での操作で立体数独をプレイすることができる。

6 ユーザテスト

今回作成した立体数独アプリケーションのユーザーテストを行ない、正答マス数、習熟に要する時間、ゲームとしての達成感などを調査した。

6.1 正答マス数

6.1.1 実験方法

6 名の被験者に本アプリケーションと RealSudoku3D をプレイしてもらい、制限時間内に何マス解くことができるかを比較した。まず各アプリケーションの使い方と立体数独の解き方の練習をしてもらった後、実験としてそれぞれのアプリケーションを 15 分ずつ、交互に 3 回ずつ解いてもらった。

実験で用いる問題は、RealSudoku3D の問題プールから 6 問を選び出し、同じ問題を提案手法でも出題できるようにした。3 問ごとにグループ分けし、問題のグループとアプリケーションの組み合わせを被験者ごとに代えて実験を行なった。

6.1.2 実験結果

各被験者 (A-F) の平均解答マス数を図 7 に示す。また、右側に 6 名の平均と標準偏差を示している。なお、実験において誤答はなかった。

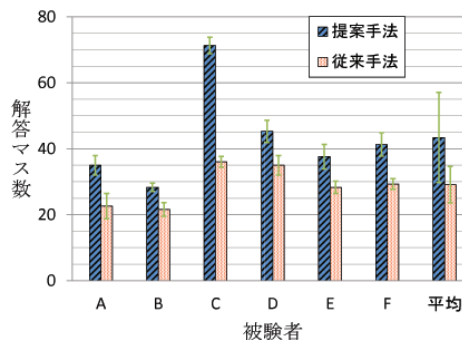


図 7 平均解答マス数

図に示すように、提案手法の方が解答マス数が多かつ

た．この差が有意であるか t 検定により確認したところ，提案手法の方が従来手法よりも解答マス数が多いことが示された ($t=3.35$, $p=1.01\%$) ．

以上の実験結果より，提案手法の方が余計な操作を省略し，必要な情報をプレイヤーに提供できていると考えられる．

6.2 操作習熟時間とゲーム性

6.2.1 実験方法

従来手法として挙げた RealSudoku3D と，今回作成した立体数独アプリケーションの操作に慣れるまでの時間を測定する実験を行った．被験者 8 名に本アプリケーションと立体数独の操作説明をした後自由にプレイしてもらい，操作に十分慣れたと思うまでの時間を計測した．また，実験の後ユーザにインタビューし，「ゲームとして面白いのはどちらか」，「立体数独を解いた達成感がより得られたのはどちらか」，という二つの質問に答えてもらった．尚，被験者は 21 歳～24 歳の男性で，3 次元数独をプレイするのは初めてであった．

6.2.2 実験結果

表 1 習熟時間とゲーム性の評価．「 T_r, T_p 」では，従来法および提案法に対する習熟時間の平均値・分散を示している．「 $T_p < T_r$ 」では提案法の習熟時間が従来法より短い被験者の人数を示している．また，「Q1」では「提案法の方がゲームとして面白い」と回答した人数，「Q2」では「提案法の方が立体数独を解いた達成感が得られる」と回答した人数を示している．尚，被験者数は 8 人である．

| 項目 | 平均 | 標準偏差 | 人数 | p 値 |
|-------------|-------------|-----------|----|-------|
| T_r | 19 min 44 s | 5.493 | - | - |
| T_p | 13 min 55 s | 13.651 | - | - |
| $T_r - T_p$ | 349.875 s | 522.674 s | - | 0.100 |
| $T_p < T_r$ | - | - | 6 | 0.035 |
| Q1 | - | - | 5 | 0.144 |
| Q2 | - | - | 6 | 0.035 |

実験の結果を表 1 に示す．各アプリケーションの平均習熟時間は，RealSudoku3D は 19 分 44 秒，本アプリケーションは 13 分 55 秒という結果になった．また，8 人中 6 名は，提案手法に対する習熟時間が従来法より短い．二項検定を行ったところ，有意水準 5 % で有意であ

ることが認められた．

また，本アプリケーションの方がゲームとして面白いと答えたのは 8 人中 5 人，立体数独を解いた達成感がより得られたと答えたのは 8 人中 6 人であった．この二つの質問項目の結果についてそれぞれ二項検定を行った．立体数独を解いた達成感がより得られたのはどちらか，という質問の結果に関しては，有意水準 5 % で有意差が認められた．ゲームとして面白いのはどちらか，という質問の結果に関しては，有意水準 15 % しか得られず，被験者数 8 名では有意差が認められなかった．

実験後，被験者に感想を自由に述べてもらったところ，RealSudoku3D は操作性が悪くゲームに集中できないという意見が多く出た．このことから，操作性の良さが立体数独を解いた達成感を高めたのではないかと推測される．一方，提案法に対しては，メモ機能などの不足を指摘するコメントがあり，今後の課題としたい．

7 むすび

本研究では，立体数独の基本情報をワンクリックで平面的に表示するインターフェースを提案した．これにより，プレイヤーは煩雑な操作から解放され，数独を解くことに集中してプレイすることができるようになった．評価実験では提案手法の方が速く解けることが示された．操作性や見づらさを改善し，簡単な操作で基本情報を提供できたと考えられる．また，習熟時間も従来法より短く，「立体数独を解いた達成感が得られる」との評価が得られた．今後，間違えた場合の救済措置やメモ機能についても検討していきたい．

参考文献

- [1] 立体ピクロス:<http://www.nintendo.co.jp/ds/c6pj/>
- [2] 株式会社ニコリ: “ポケット数独初級篇”，ソフトバンククリエイティブ (2006)
- [3] RealSudoku3D:<http://www.realsudoku3d.com/>
- [4] Sudokube:<https://play.google.com/store/apps/details?id=com.highwaynorth.sudokube>
- [5] SudokuBallDetective:<http://www.play-asia.com/sudoku-ball-detective-paOS-13-49-jp-70-3g11.html>
- [6] iSudokube:<http://iwire.jp/appch/apps/346490571/iSudokube/>

[7] Unity: <http://japan.unity3d.com/>

田中 貴拓



2015年東邦大学理学部情報科学科卒業。同年株式会社ユニスティ入社。



1974年生まれ。2003年東京大学大学院総合文化研究科博士課程修了。博士(学術)。2005年東邦大学理学部情報科学科専任講師、2013年から准教授。専門はコンピュータグラフィックス。

新谷 幹夫



1979年、早稲田大学工学部応用物理学科卒。1981年、同大学工学研究科物理及び応用物理学専攻修士課程了。同年、日本電信電話公社武蔵野電気通信研究所入所。視覚系の心理物理学的研究、文字認識の研究、コンピュータグラフィックスの研究などに従事。2001年、東邦大学理学部情報科学科教授。博士(工学)。

岩穴口 貴祥



2015年東邦大学理学部情報科学科卒業。同年東邦大学大学院理学研究科入学。

白石 路雄