DOI: 10.2507/36th.daaam.proceedings.xxx

CELLULAR ENVIRONMENT AND 4-DIMENSIONAL CUBES FOR EFFECTIVE REPRESENTATION OF ROBOT IN LABYRINTH

Roman V. Khelemendik & Valentin E. Pryanichnikov

This Publication has to be referred as: Khelemendik, R[oman] & Pryanichnikov, V[alentin] (2025). CELLULAR ENVIRONMENT AND 4-DIMENSIONAL CUBES FOR EFFECTIVE REPRESENTATION OF ROBOT IN LABYRINTH, Proceedings of the 36th DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/36th.daaam.proceedings.xxx

Abstract

In robotics, one of the essential tasks is to simplify the representation of the operating environment in the form of a two-dimensional cellular field of dimension nxm. Let's consider the formalism of representing data about the operating environment in the robot's memory in the form of a labyrinth in which some neighboring cells are separated by walls (partitions) and the robot cannot move from one such cell to the next. The main task is to determine the control of the robot (trajectory planning), which ensures the transition from the current position (i_0, j_0) - forming a sequence of transitions to neighboring cells so that there is a path to the target cell (i_g, j_g). Let's divide the cellular field into enlarged 4x4 cells (or larger ones). For them, we will build correlated 4-dimensional cubes, that will set the numbering inside each 4x4 square in such a way, that the numbering of vertices in binary form allows you to move from one vertex (cell) to the neighboring one by changing only one bit. The absence of a corresponding edge for the transition means, that these cells are separated by an obstacle. The path on such a field is constructed so that the entrances and exits of the paths on adjacent 4x4 cells coincide. The separation of neighboring cells by an obstacle can be set both a priori (then this is the task of a known map), and due to the activation of the sensor system of near detection. The paper analyzes the proposed method of constructing correlated four-dimensional cubes (graphs) on which the sequence of transitions and path search is carried out in a more efficient and economical way.

Keywords: Mobile service, collaborating robots; representation of environment information; n-dimensional cubes.

1. Introduction

The movement of objects across the neighboring cells of a field on a plane is considered. We assume that the number of cells is 2^n, and the field has dimensions 2^m x 2^{n-m}. We study methods for numbering cells with Boolean sets of length n and numbers from 0 to 2^{n-1} such that adjacent (vertically or horizontally) fields differ in exactly one binary digit. This numbering helps save time and memory when formalizing movement on a discrete plane. This, in particular, allows us to change just one bit in software to move the robot one cell in a straight line; we also gain new capabilities for modeling robots in flat labyrinths.

For problems where fast and efficient transitions between states of objects are important, Gray codes ([2], [3]) are used. Their distinctive feature is that only one position changes when transitioning from one code combination to another. Gray codes have been studied in terms of Hamiltonian cycles in graphs and n-dimensional Boolean cubes ([2], [4], [5]), development of combinatorial algorithms ([6]), error correction in communication via a communication channel ([7]), and design of hard drives and databases ([7], [8]).

In this article, we propose specialized truncated n-dimensional cubes, flat rectangular fragments and their combinations for new applications of n-dimensional cubes and Gray codes, in particular, for moving robots in cellular environments. In practice, this is important for representing the working environment in the robot's memory as a labyrinth and for planning the movements of a mobile service robot such as Amur. Based on these principles, it is possible to build a description of the efficient algorithm for by passing wards by a logistics robot in a hospital.

2. Definitions and designations

A simple path in a graph or *a simple chain* of length k>1 on a graph G will be called a sequence (set) $C=<v_1,u_1,\ldots,u_{k-1},v_k>$, where v_i belongs to the set of vertices V, and u_i to the set of edges U, and none of the elements of the sequence is repeated (if there are repeating elements in this sequence, the word “simple” is omitted). Similarly, a simple cycle in a graph is a simple chain C for which $v_1 = v_k$.

The set of all sets $<\alpha_1...\alpha_n>$ of binary (Boolean) values is called an *n-dimensional (Boolean) cube* of size n and is denoted by B^n. For sets of values we use the following notations: $\boldsymbol{\alpha}^n=<\alpha_1...\alpha_n>$. *The distance* $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)$ between vertices $\boldsymbol{\alpha}^n$ and $\boldsymbol{\beta}^n$ is the number of digits in which they differ: $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)=\Sigma^n_{i=1}|\alpha_i-\beta_i|$. *The number* $\eta(\boldsymbol{\alpha}^n)$ *of a set* $\boldsymbol{\alpha}^n$ is the number whose binary representation is $\boldsymbol{\alpha}^n$: $\eta(\boldsymbol{\alpha}^n)=\Sigma^n_{i=1}2^{n-i}\alpha_i$. Geometrically, the Boolean cube B^n is a graph consisting of 2^n vertices, and any two of its vertices $\boldsymbol{\alpha}^n$ and $\boldsymbol{\beta}^n$ are connected by an edge if and only if $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)=1$. *A Hamiltonian cycle* (HC) in a graph is a sequence of all its vertices, connected by edges and not repeating, in which the last vertex is connected by an edge to the first. If the vertices of B^n are denoted by binary words (sets) of length n, then the HC in B^n is also called the Gray code [2, 3]. Fig. 1 shows flat projections of n-dimensional cubes for n=1,2,3,4, as well as some HCs (arrows).

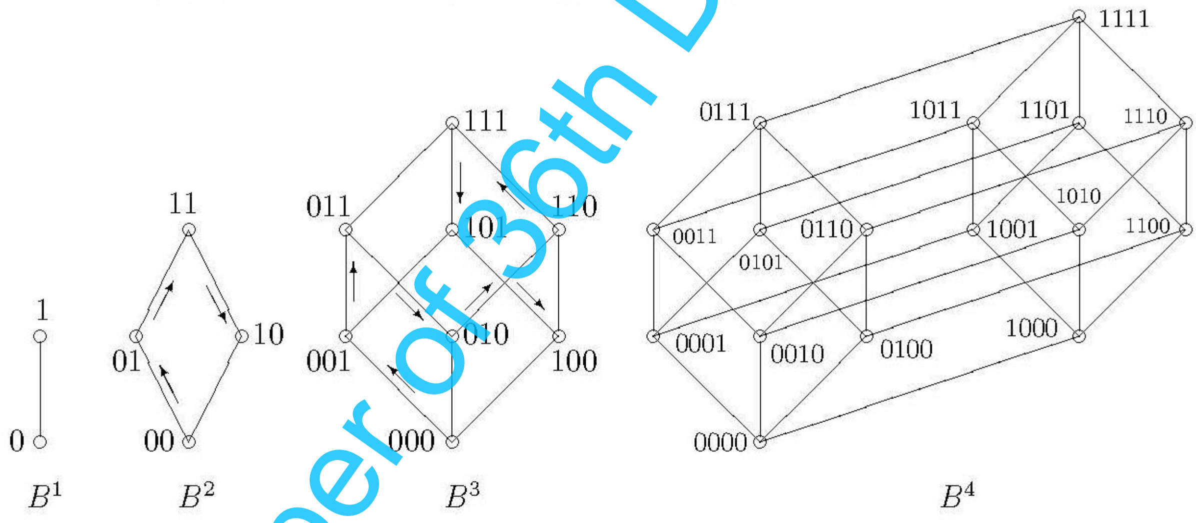

Fig. 1. n-dimensional cubes B^n, $n\leq 4$

A plane rectangular (m,n)*-fragment* ($PF_{(m,n-m)}$) of an n-dimensional cube is a subgraph B^n such, that its vertices are mapped to the cells of a rectangle of size 2^m x 2^{n-m}, the edges of the subgraph connect vertically and horizontally adjacent (neighboring) cells of this rectangle, and the remaining edges from B^n are removed. In this case, the cells of a 2^m x 2^{n-m} rectangle can be numerated with numbers representing binary codes (of vertices). Thus, the HC in Fig. 1 for B^2 is the basis for $PF_{(0,2)}$ – a rectangle of size 1 x 4, the cells of which are sequentially numbered with the numbers 0, 1, 3, 2; and if we restore the edge <00>–<10> in this graph, we obtain $PF_{(1,1)}$ – a rectangle of size 2 x 2 (square), consisting of the rows [0,2], [1,3].

3. Construction of the cellular environment and PF (plane fragments) for the first values of n.

Obviously, for n=1 there is only one PF, namely $PF_{(0,1)}$ with a corresponding rectangle of size 1 x 2, and “almost identically” (indistinguishably) numbered cells – in the order [0,1] or [1,0]. In the case of n=2, as shown above, there are only two PFs: $PF_{(0,2)}$ (strip) and $PF_{(1,1)}$ (square), for which the indistinguishable cell numberings are given above. We move on to the case of n≥3. The statement is obvious:

Statement 1. *Every* HC *in* B^n *uniquely corresponds to some* $PF_{(0,n)}$, *and vice versa*.

If in HC for B^n we can recover (n-1) edges, then we get $PF_{(1,n-1)}$.

Statement 2. *For* $n\geq3$, *not every* HC *in* B^n *has a corresponding* $PF_{(1,n-1)}$.

Indeed, for $n\geq3$ there is HC (2,0,4,5,1,3,7,6). However, to obtain $PF_{(1,2)}$, we can restore edges (2,6) and (5,1), but we cannot restore edges (0,7) and (4,3), since they are not in B^3. Similar, as well as more complex, situations arise when n>3.

A sequence of numbers $\{a_k\}$ of length s is called ↑-unimodal (↓-unimodal) if there exists k such that $a_1 < a_k > a_{k+1} > \ldots > a_s$ ($a_1 > a_k < a_{k+1} < \ldots < a_s$), $1<k<s$. Thus, the sequences (0,1,3,2) and (6,7,5,4) are ↑-unimodal. Obviously, if the sequence $\{a_k\}$ ↑-unimodal (↓-unimodal), then its inverse (taken in reverse order) sequence is ↓-unimodal (↑-unimodal). By combining two oppositely directed unimodal sequences (0,1,3,2) and (4,5,7,6), we obtain $PF_{(1,2)}$, consisting of rows with these sequences (Fig. 2). Note that the ↑-unimodal sequence is simply unimodal ([1], p. 178), so in our definition, unimodality is somewhat specified (clarified) for this sequence, whether it is increasing or decreasing.

4	5	7	6
0	1	3	2

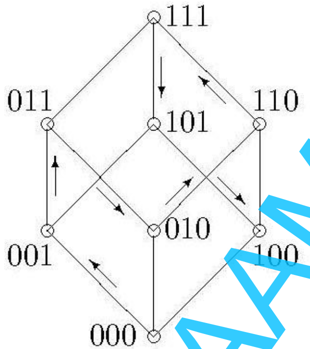

Fig. 2. $PF_{(1,2)}$, B^3 and HC in B^3

Further, unimodal sequences of numbers in rows (columns) will be called unimodal rows (columns). It is easy to see that both unimodal rows form a GC in the B^3 subgraphs, which are B^2 cubes (squares) and represent boolean functions depending only on the second and third variables.

Statement 3. *For* B^4 *there exists a* $PF_{(2,2)}$, *such that the sequences of all its rows and columns are unimodal.*

One of these $PF_{(2,2)}$ and its corresponding B^4, from which the edges unnecessary for the 4x4 square (a truncated four-dimensional cube) are removed, are shown in Fig. 3.

8	10	14	12
9	11	15	13
1	3	7	5
0	2	6	4

1000	1010	1110	1100
1001	1011	1111	1101
0001	0011	0111	0101
0000	0010	0110	0100

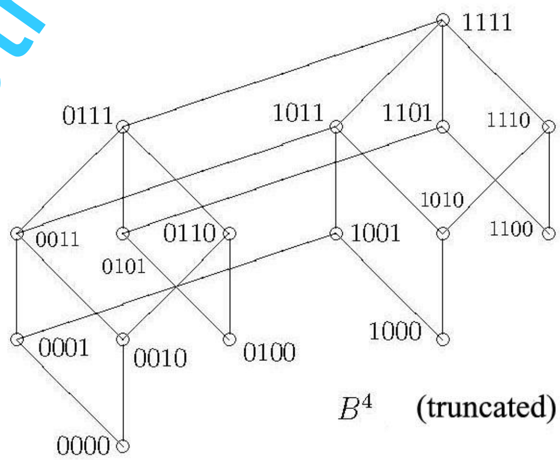

Fig. 3. $PF_{(2,2)}$ and truncated B^4

It is easy to see, that in this 4x4 square, all rows are unimodal in the second and third digits, and the columns are in the first and fourth. More specifically, from this we get, that the horizontal movement between the second and third verticals corresponds to a change in the second digit, and the movement between the first and second or third and fourth verticals corresponds to a change in the third digit. Similarly, a vertical movement between the second and third horizontals corresponds to a change in the first digit, and a movement between the first and second or third and fourth horizontals corresponds to a change in the fourth digit.

The weight $\omega(\boldsymbol{\alpha}^n)$ *of a boolean set* $\boldsymbol{\alpha}^n$ is the number of its unit components: $\omega(\boldsymbol{\alpha}^n) = \Sigma^n_{i=1}\, \alpha_i$. Then the weights in the unimodal rows (x_2 and x_3) and columns (x_1 and x_4) also form a unimodal sequence: 0, 1, 3, 2. Thus, instead of the $PF_{(2,2)}$ shown in Fig. 3, other 4x4 squares can be constructed using forward and reverse unimodal sequences of weights for various pairs of digits. Note that to obtain $PF_{(1,3)}$ from $PF_{(2,2)}$, it is sufficient to rotate 270 degrees counterclockwise relative to the midpoint of the left side of the 4x4 square of the two upper rows. Then in $PF_{(1,3)}$ we obtain rows consisting of the concatenation of the unimodal rows (13,15,11,9,1,3,7,5) and (12,14,10,8,0,2,6,4), and in the truncated cube B^4 we remove 4 edges corresponding to the "gaps" (due to rotation), and restore (from the full B^4) the edge (8,0).

Continuing this process (cutting along the central horizontal line, rotating 270 degrees and restoring one edge), from $PF_{(1,3)}$ we obtain $PF_{(0,4)}$.

4. Construction of the cellular environment and PF for n≤6.

To go from n=4 to n=5, simply duplicate the $PF_{(2,2)}$, mirroring it relative to the vertical centerline, increasing each number by 16, and positioning this PF to the right of the original. Then, for all adjacent cells in both $PF_{(2,2)}$, the condition $r(\boldsymbol{\alpha}^n,\boldsymbol{\beta}^n)=1$ remains satisfied, and for the fourth and fifth columns, horizontal adjacency means a difference of only one digit – the most significant one (Table 1).

Recording cells by natural numbers							
8	10	14	12	28	30	26	24
9	11	15	13	29	31	27	25
1	3	7	5	21	23	19	17
0	2	6	4	20	22	18	16
Recording cells as binary numbers							
01000	01010	01110	01100	11100	11110	11010	11000
01001	01011	01111	01101	11101	11111	11011	11001
00001	00011	00111	00101	10101	10111	10011	10001
00000	00010	00110	00100	10100	10110	10010	10000

Table 1. Cell numbering in $PF_{(2,3)}$ (rectangle of size 2^m x 2^{n-m}, m=2, n=5 , i.e. the size of rectangle is 4x8).

When moving from n=5 to n=6, we duplicate the constructed $PF_{(2,3)}$ and perform similar actions of mirror symmetry relative to the horizontal midline, increasing the numbers by 2^5=32, and positioning the new PF above the original. An example of such a $PF_{(3,3)}$ and the associated truncated cube B^6 are shown in Table 2 and Fig. 4. This $PF_{(3,3)}$ is not exactly an extension of the previously constructed $PF_{(2,3)}$, since in the starting $PF_{(2,2)}$ we chose oppositely directed unimodalities: rows are ↓-unimodal, and columns are ↑-unimodal.

Recording cells by natural numbers							
34	32	36	38	54	52	48	50
35	33	37	39	55	53	49	51
43	41	45	47	63	61	57	59
42	40	44	46	62	60	56	58
10	8	12	14	30	28	24	26
11	9	13	15	31	29	25	27
3	1	5	7	23	21	17	19
2	0	4	6	22	20	16	18
Recording cells as binary numbers							
100010	100000	100100	100110	110110	110100	110000	110010
100101	100001	100101	100111	111101	110101	110001	110011
101011	101001	101101	101111	111111	111101	111001	111011
101010	101000	101100	101110	111110	111100	111000	111010
001010	001000	001100	001110	011110	011100	011000	011010
001011	001001	001101	001111	011111	011101	011001	011011
000101	000001	000101	000111	011101	010101	010001	010011
000010	000000	000100	000110	010110	010100	010000	010010

Table 2. Cell numbering in $PF_{(3,3)}$

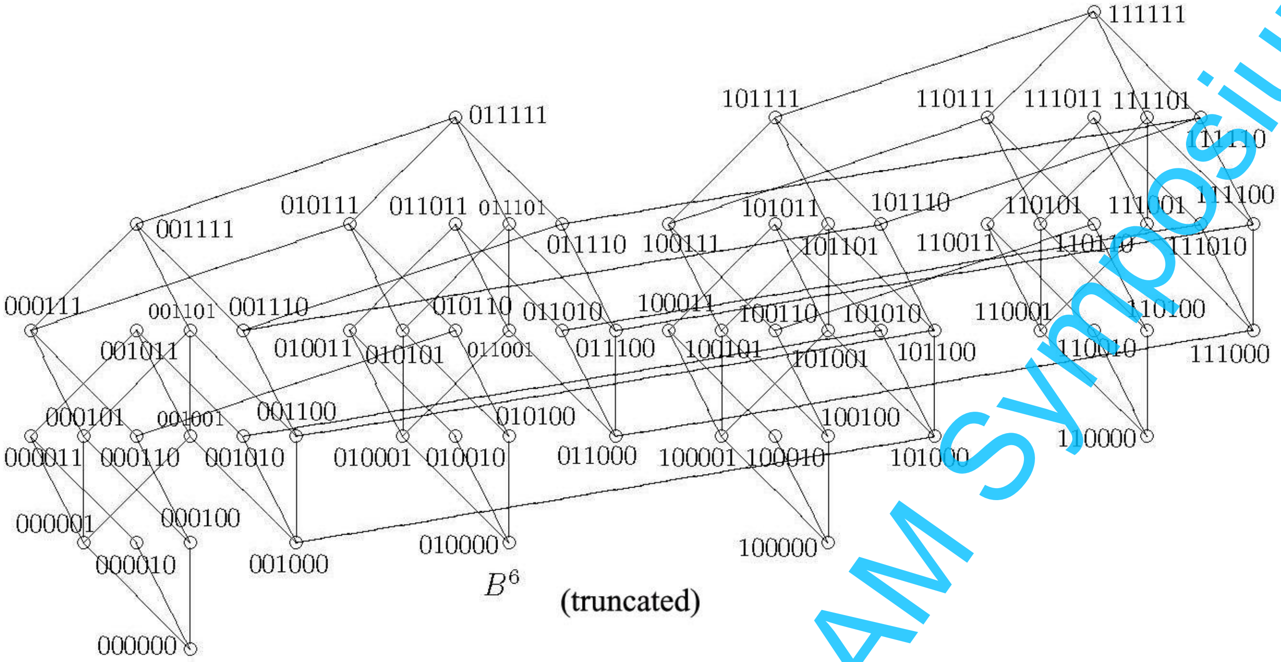

Fig. 4. Truncated six-dimensional Boolean cube

5. Construction of the cellular environment and PF for n>6.

We will construct the $PF_{(4,4)}$ for n=8, and its lower half will also be the $PF_{(3,4)}$ for n=7. First, we will slightly modify the original $PF_{(3,4)}$ for n=4, according to which we will construct the $PF_{(3,3)}$ for n=6 (see Table 3) as was done above.

Recording cells by natural numbers							
32	33	37	36	52	53	49	48
34	35	39	38	54	55	51	50
42	43	47	46	62	63	59	58
40	41	45	44	60	61	57	56
8	9	13	12	28	29	25	24
10	11	15	14	30	31	27	26
2	3	7	6	22	23	19	18
0	1	5	4	20	21	17	16
Recording cells as binary numbers							
100000	100001	100101	100100	110100	110100	110001	110000
100010	100011	100111	100110	110110	110111	110011	110010
101010	101011	101111	101110	111110	111111	111011	111010
101000	101001	101101	101100	111100	111101	111001	111000
001000	001001	001101	001100	011100	011101	011001	011000
001010	001011	001111	001110	011110	011111	011011	011010
000010	000011	000111	000110	010110	010111	010011	010010
000000	000001	000101	000100	010100	010101	010001	010000

Table 3. Unimodal cell numbering in $PF_{(3,3)}$

The main difference between this PF and the previous $PF_{(3,3)}$ (see Table 2) lies precisely in the lower left 4x4 square: previously, the rows were unimodal in digits 2 and 3, and the columns in digits 1 and 4, but now the rows are unimodal in even digits (2, 4), and the columns in odd digits (1, 3). When moving to the new $PF_{(3,3)}$, unimodality in row concatenations is also preserved for even digits, and unimodality in column concatenations is preserved for odd digits. Continuing the transition from n=6 to n+1 in width, and then to n+2 in height, as was previously done for n=4, we arrive at $PF_{(4,4)}$ – a 16x16 square consisting of 16 4x4 squares (Table 4).

The cells in this field contain two-digit numbers written in hexadecimal notation. The first number represents the position of a 4x4 square, and the second represents the cell's position within the 4x4 square (Table 5). The cells of this field contain two-digit numbers written in hexadecimal notation. In this case, the first number characterizes the position of the 4x4 square, and the second characterizes the location of the cell already in the 4x4 square (Table 5, the symbols * denote arbitrary hexadecimal numbers, and $*^4$ – four arbitrary binary digits).

80	81	85	84	94	95	91	90	D0	D1	D5	D4	C4	C5	C1	C0
82	83	87	86	96	97	93	92	D2	D3	D7	D6	C6	C7	C3	C2
8A	8B	8F	8E	9E	9F	9B	9A	DA	DB	DF	DE	CE	CF	CB	CA
88	89	8D	8C	9C	9D	99	98	D8	D9	DD	DC	CC	CD	C9	C8
A8	A9	AD	AC	BC	BD	B9	B8	F8	F9	FD	FC	EC	ED	E9	E8
AA	AB	AF	AE	BE	BF	BB	BA	FA	FB	FF	FE	EE	EF	EB	EA
A2	A3	A7	A6	B6	B7	B3	B2	F2	F3	F7	F6	E6	E7	E3	E2
A0	A1	A5	A4	B4	B5	B1	B0	F0	F1	F5	F4	E4	E5	E1	E0
20	21	25	24	34	35	31	30	70	71	75	74	64	65	61	60
22	23	27	26	36	37	33	32	72	73	77	76	66	67	63	62
2A	2B	2F	2E	3E	3F	3B	3A	7A	7B	7F	7E	6E	6F	6B	6A
28	29	2D	2C	3C	3D	39	38	78	79	7D	7C	6C	6D	69	68
08	09	0D	0C	1C	1D	19	18	58	59	5D	5C	4C	4D	49	48
0A	0B	0F	0E	1E	1F	1B	1A	5A	5B	5F	5E	4E	4F	4B	4A
02	03	07	06	16	17	13	12	52	53	57	56	46	47	43	42
00	01	05	04	14	15	11	10	50	51	55	54	44	45	41	40

Table 4. Unimodal cell numbering in $PF_{(4,4)}$

8*	9*	D*	C*	*8	*9	*D	*C
A*	B*	F*	E*	*A	*B	*F	*E
2*	3*	7*	6*	*2	*3	*7	*6
0*	1*	5*	4*	*0	*1	*5	*4
Binary numbering				Binary numbering			
1000*^{4}	1001*^{4}	1101*^{4}	1100*^{4}	*^{4}1000	*^{4}1001	*^{4}1101	*^{4}1100
1010*^{4}	1011*^{4}	1111*^{4}	1110*^{4}	*^{4}1010	*^{4}1011	*^{4}1111	*^{4}1110
0010*^{4}	0011*^{4}	0111*^{4}	0110*^{4}	*^{4}0010	*^{4}0011	*^{4}0111	*^{4}0110
0000*^{4}	0001*^{4}	0101*^{4}	0100*^{4}	*^{4}0000	*^{4}0001	*^{4}0101	*^{4}0100
Big square				Little square			

Table 5. Numbering of cells of the large and small square in $PF_{(4,4)}$

Thus, any horizontal movements correspond only to a change in even digits, and any vertical movements correspond only to a change in odd digits. In this case, a change in the 1st (2nd) digit is a transition between horizontals (verticals) 8 and 9; a change in the 3rd (4th) digit is a transition between horizontals (verticals) 4-5, 12-13; a change in the 5th (6th) digit is a transition between horizontals (verticals) 2-3, 6-7, 10-11, 14-15; a change in the 7th (8th) digit is a transition between horizontals (verticals) 1-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13-14, 15-16.

The algorithm for constructing a PF from n=2k to n+1 in width, and then to n+2, also works in the general case n>8.

6. Comparison of traditional and multidimensional representations of the cellular environment.

It is common to represent a robot's position in a cellular environment as a pair of coordinates <x,y>, where x is the vertical axis and y is the horizontal axis. The simplicity and clarity of this representation are beyond doubt. Moving to an adjacent cell then means adding or subtracting one from the x or y variable. It should be noted, that repeated execution of such simple arithmetic operations in computing devices with a multidimensional representation of the cellular environment requires significantly less time and memory. As an example, consider the case of n = 8 and a path from cell <1,1> to cell <16,16>. Storing the current position of an object (robot) requires one byte for both cellular representations. Any shortest path from <1,1> to <16,16> consists of 15 elementary vertical and 15 horizontal movements. In the multidimensional representation, an elementary movement results in a speed savings of some δ, since changing one bit is faster than adding or subtracting one. For 30 movements, this already yields a savings of 30δ. We achieve memory savings if we want to remember a specific route from <1,1> to <16,16>. With the traditional representation, we need to remember a sequence of coordinates (bytes), i.e., 30 bytes, whereas in the multidimensional representation, 30 bits are sufficient. At the same time, the hierarchy of bits by seniority allows us to obtain instant detailed information about the robot even when only one bit changes: for example, the inversion of one of the most senior bits immediately means that the robot has moved to the other half of the square (vertical or horizontal).

7. Formalism of representing a labyrinth in the robot's memory.

A labyrinth in a cellular environment is a PF in which transitions between some adjacent cells may be forbidden. Let's first consider a field of size $2^{n/2}$ x $2^{n/2}$, which we represent as a $PF_{(n,n)}$, as shown above. A specific labyrinth is then defined by enumerating all pairs of adjacent cells between which transitions are forbidden. Since the total number of possible transitions already at n=8 amounts to hundreds, listing forbidden pairs of adjacent cells as pairs of their coordinates is only suitable for a small number of prohibitions; moreover, it is not visually clear. Using a multidimensional representation of the cellular environment, forbidden pairs are specified by specifying the bits responsible for the transitions. Thus, for n=8, a set of values β^8=<β_1,…, β_8> is first formed such that β_j=1 if and only if, for the case of at least one change in the j-th digit, there is a transition prohibition. Then, an additional set ξ is formed, allowing us to specify all such prohibitions. The representation of prohibitions becomes clear if the labyrinths are represented as truncated cubes in which prohibited edges are colored red (or removed). Further refinement of prohibitions and formalization of routes can be carried out using Boolean functions of a small number of variables and propositional logic.

8. Conclusion

The key provision of the article is to propose a new logical representation for describing the operating environment of mobile service robots in the internal memory of a top-level control microprocessor. The practical significance of the results lies in the possibility of maximum data compression, close to the theoretical limit with a very promising method for determining the neighborhood on a two-dimensional cellular field. The most important conclusion is that the approach found opens up new tools for creating effective on-board software for collaborative robots. It should also be noted, that the work was carried out mainly at the Keldysh Institute of Applied Mathematics of the Russian Academy of Sciences as part of research related to the “Intelligent robotronics” project (see also the works [9], [10], [11], [12]).

9. References

[1] Yablonskii S.V. (1986). Vvedenie v diskretnuiu matematiku / M.: Nauka, 1986. — 384s.

[2] Perezhogin A. L., Bykov I. S. Review of constructions and properties of Gray codes // Mathematical Issues of Cybernetics, no. 20, A collection of papers, ed. by O.B.Lupanov, Moscow: Fizmatlit, 2022, pp. 41-60. URL: http://library.keldysh.ru/mvk.asp?id=2022-41 DOI: 10.20948/mvk-2022-41

[3] Gray F. Pulse code communications.—Patent no. 2632058 US.

[4] Gilbert E. N. Gray codes and paths on the n-cube //The bell system technical journal.— 1958.—V. 37, No 3.—P. 815–826.

[5] Perezhogin A. L. On the spectrum of Hamiltonian cycles in the n-cube // Journal of Combinatorial Theory, Series B.—2021.—V. 151.—P. 435–464.

[6] Reingold E. M., Nievergelt J., Deo N. Combinatorial Algorithms: Theory and Practice.—Prentice Hall College Div., 1977.

[7] Goddyn L., Lawrence G. M., Neme h E. Gray codes with optimized run lengths // Utilitas Mathematica.—1988.—V. 34.—P. 179–192.

[8] Faloustos C. Gray codes for partial match and range queries // IEEE Transactions on Software Engineering.—1988.—V. 14, No 10.—P. 1381–1393.

[9] Davydov D.V., Eprikov S.R., Kirsanov K.B., Pryanichnikov V.E. (2017). Service Robots Integrating Software and Remote Reprogramming, Proceedings of the 28th DAAAM International Symposium, pp.1234-1240, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-11-2, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/28th.daaam.proceedings.172.

[10] Pryanichnikov, V[alentin]; Kolesov, Y[uriy]; Tarasov, R[adomir] & Plotnikov, A[leksey] (2021). Integration Software for a Distributed Group of Mobile Service Robots, Proceedings of the 32nd DAAAM International Symposium, pp.0311-0315, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978- 3-902734-33-4, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/32nd.daaam.proceedings.045

[11] Aryskin A.A., Bogdanovich A.V., Davydov O., Khelemendik R., Petrakov M., Pryanichnikov V.E., (2019). Control and Interaction Algorithms for Industrial and Service Robots, Chapter 33 in DAAAM International Scientific Book 2019, pp.377-384, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-24-2, ISSN 1726-9687, Vienna, Austria DOI: 10.2507/daaam.scibook.2019.33

[12] Khelemendik R., Pryanichnikov V. E. (2020). Contradictory Reasoning Connecting with the Problems of Intelligent Robotronics, Proceedings of the 31st DAAAM International Symposium, pp.0276-0281, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-29-7, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/31st.daaam.proceedings.038