Brock University

Department of Computer Science

Choosing the root node of a quadtree

Xiang Yin, Ivo Düntsch, Günther Gediga

Technical Report # CS-09-08
Mai 2009

Brock University
Department of Computer Science
St. Catharines, Ontario
Canada L2S 3A1
www.cosc.brocku.ca

Choosing the root node of a quadtree

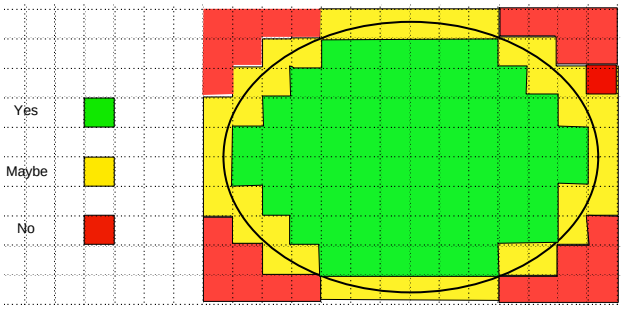
Xiang Yin, Ivo Düntsch,
Dept. of Computer Science, Brock University,
St. Catharines, Ontario, Canada, L2S 3A1
xy07to@badger.ac.brocku.ca, duentsch@brocku.ca
Günther Gediga
Dept. of Psychology, Universität Münster,
Fliednerstr. 21, D–48149 Münster
guenther@gediga.de

***Abstract*—Granular computing is closely related to the depth of the detail of information with which we are presented, or choose to process. In spatial cognition and image processing such detail is given by the resolution of a picture. The quadtree representation of an image offers a quick look at the image at various stages of granularity, and successive quadtree representations can be used to represent change. In this paper we present a heuristic algorithm to find a root node of a region quadtree which reduces the number of leaves when compared with the standard quadtree decomposition.**

I. Introduction

In the rough set model [1] objects are described by a lower approximation and an upper approximation induced by an equivalence relation θ on the base set U; both approximations are collections of equivalence classes of θ.

Fig. 1. Rough set approximation



In Figure 1, the cells represent the classes of θ. The set to be approximated is the area of the closed ellipse X; the union of all cells completely contained in X is the lower approximation $\underline{X}_\theta$, and the union of all cells that intersect X is the upper approximation $\overline{X}_\theta$. Therefore, the plane is partitioned into three disjoint regions, namely, the "Yes" region $\underline{X}_\theta$, the "Maybe" region $\overline{X}_\theta \setminus \underline{X}_\theta$, and the "No" region $U \setminus \overline{X}_\theta$.

In the present paper we shall investigate how data as in Figure 1 may be efficiently represented using as little data space as possible. Our motivation is the investigation of J. J. Gibson's [2] approach to ecological perception by two of the present authors [3]. According to Gibson, the main task of perception is to recognize the invariant parts within a variant world:

> "We perceive that the environment changes in some respects and persists in others. We see that it is different from time to time, even moment to moment, and yet that it is the same environment over time. We perceive *both* the change and the underlying non–change. My explanation is that the perceptual systems work by detecting invariants in the flux of stimulation but are also sensitive to the flux itself". [4]

If we think of U as a plane region and the cells as pixels output by a sensor, then θ fixes the granularity at which the region is perceived; each granule may consist of a class of an information system which describes the nature of the pixel – color, hue, intensity, shade, or just on/off. Table I is an excerpt of such a situation. The "container" U represents a snapshot (freeze) of the visual field which is composed of a sequence of such containers. Here we interpret 1 as "true" or "present",

TABLE I
Bitstrings of a perceived field

Cell	Status	Color			Intensity		
	On	red	green	…	high	medium	low
$\langle 0,0\rangle$	1	1	0		?		
$\langle 0,1\rangle$	1	?			0	1	0
$\langle 0,2\rangle$	0	0			0		
$\langle 0,3\rangle$	?	?			?		
…	…	…			…		

and 0 as "false" or "not present"; a question mark in a cell indicates ignorance about the cell content.

Briefly, the logical background is as follows: We assume that an agent a (animal, human, robot) knows ("sees") some or all properties of some or all bits; in other words, we are setting up an epistemic system with one agent a and a knowledge operator K. The atomic sentences are of the form `attribute value = constant`, e.g. "bit is on" or "color is red". For each atomic proposition p we consider $K(p)$, $K(\neg p)$ and $\neg K(p) \wedge \neg K(\neg p)$; in other words, "$a$ knows that p", "a knows

Ivo Düntsch gratefully acknowledges support from the Natural Sciences and Engineering Research Council of Canada.

Günther Gediga is also adjunct professor in the Department of Computer Science, Brock University

that $\neg p$", and "a has no knowledge of the truth of p". We suppose that K satisfies the axiom T, i.e. $K(p) \Longrightarrow p$; in other words, we suppose that the agent's observations are correct. The natural evaluation e of the propositional variables is given by the cells under consideration.

The agent, therefore, presents a matrix consisting of r rows and s columns each of which represents $K(p)$ for an atomic proposition p; these, in turn, are grouped by the un–binarized attribute they belong to. There are some semantical considerations to take care of – e.g. if $K(p) = 1$, then the agent knows that $K(q) = 0$ for all other atomic sentences obtained from the attribute.

In the simplest case we suppose that the information given by a cell is just an on/off state, given by only one symmetric (binary) attribute which we may or may not be able to perceive. This situation is a strong simplification – in more realistic situations, each cell may contain a whole information system as in Table I. For an overview of representation of imprecision in finite resolution data and its connections to rough set theory we invite the reader to consult [5].

II. Quadtrees and rough sets

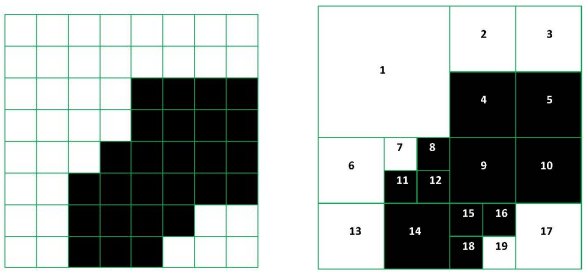
In order to reduce the storage requirements of (a sequence of) images that may be given in a hierarchical order, the design of efficient data representations has been studied extensively. As just one example, binary decision diagrams have been utilized to represent rough set information systems along with their indiscernibility relations [6]. Quadtrees, introduced in the early 1970s [7], are another tree–like data type. These have since become a major representation method for spatial data. Owing to their hierarchical nature, they are our method of choice to represent regions such as the ones in Figure 1.

The scenario for a region quadtree is as follows: We are given an a window of consisting of black cells situated in a fixed image area A of dimension $2^m \times 2^m$. A is recursively partitioned into equal sized quadrants until each quadrant consists entirely of unicolored cells. The process can be represented by a tree each non–leaf node of which has four children, corresponding to the four quadrants NW, NE, SW, SE. Each descendant of a node g represents a quadrant in the plane whose origin is g and which is bounded by the quadrant boundaries of the previous step.

We shall only concentrate on representing one of the three partition classes; since the regions Yes/Maybe/No are disjoint, it is straightforward to generalize the representation to all three regions (see e.g. [8]).

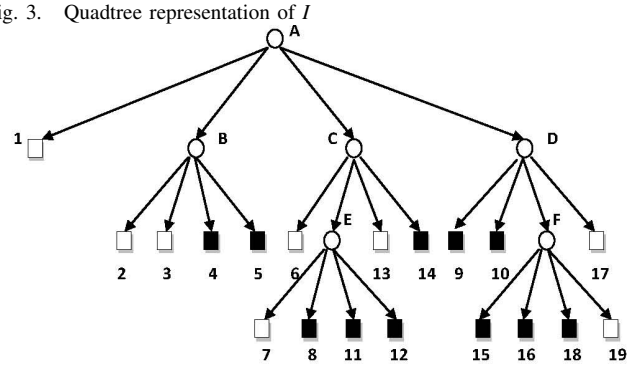
Consider, for example, the region I placed into an area of dimension $2^3 \times 2^3$, shown in Figure 2 [9]. The NW quadrant is entirely made up of white cells, thus leading to a white leaf node which terminates this branch. The NE quadrant contains cells of different color; therefore an intermediate (grey) node is placed into the center of the NE quadrant, which is divided into four quadrants. Each quadrant is homogeneous in color, and thus, four leaf nodes are produced, two of which are white and

Fig. 2. A region I of cells and its decomposition



two of which are black. The process continues until the whole image I is decomposed into square regions. The complete quadtree for the image of Figure 2 is shown in Figure 3.

Fig. 3. Quadtree representation of I



The tree consists of eight white leaf nodes and eleven black leaf nodes; there are six intermediary ("grey") nodes. As a hierarchical decomposition method, the quadtree structure lends itself well to varying granularity, since each level of the tree is a refinement of the previous one. This is well within the rough set paradigm; indeed, as early as 1982 Ranade et al [10] introduced an approximation of images which can be regarded as a special case of Pawlak's approach: each level j of the quadtree corresponds to an equivalence relation θ_j; θ_0 is the universal relation, and θ_m is the identity (the pixel level). The inner approximation of I at level j consists of all black leaf nodes obtained up to level m, and the outer approximation additionally contains the grey nodes for this level. As an example, Table II shows the upper and lower approximations of the image of Figure 2 with respect to the resolution corresponding to the levels of the quadtree.

TABLE II
Rough approximation and quadtrees

Level	Lower approximation	Boundary
0	$\emptyset$	A
1	$\emptyset$	B, C, D
2	$4, 5, 14, 9, 10$	E, F
3	$4, 5, 14, 9, 10, 8, 11, 12, 15, 16, 18$	$\emptyset$

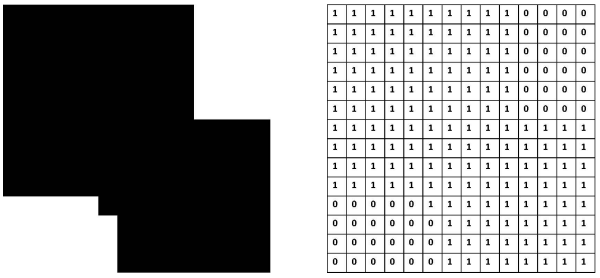
III. Choosing the frame

Efficient algorithms are available to decompose the image with the root node of the quadtree in center of a square frame: It

was shown by Proietti [8] that the quadtree decomposition of a square of side length n inside a fixed frame of side length 2^k can be done in $O(n)$ time, independent of t, thus improving an earlier result by Aref and Samet [11] which depended on t.

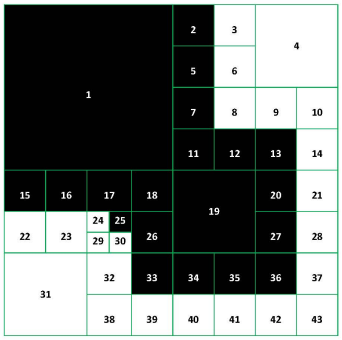
It may be argued that in the visual field, the choice of the base frame is somewhat arbitrary and, as a consequence, the root node of the quadtree representation of an image I may be to some extent variable. Indeed, if we aim to minimize the number of black nodes of a quadtree representation of I, then the choice of the root node may influence the size of the quadtree. A striking example is the case, when an optimal position with respect to a central root node is shifted by one pixel, see e.g. the discussion in [12]. We shall illustrate this with another example [13], p. 221. An image and its bounding box are shown in Figure 4.

Fig. 4. A region I and its bounding box



Following [13], we place the image into an 8×8 grid as shown in Figure 5; the corresponding quadtree can be found in Figure 6. It contains 43 leaf nodes, 20 of which are black. If we have

Fig. 5. 1st decomposition of I



the freedom to choose where the image I is placed into an area A, then a smaller number of leaf nodes can be obtained. Figure 7 shows a different placement, and Figure 8 the corresponding quadtree. This second quadtree contains only 34 leaf nodes, 14 of which are black. These observations now lead to the following question:

- Suppose we are given an image of black cells with bounding box dimension $n \times m$. Which position of the root node will minimize the number of black leaf nodes in the quadtree decomposition of A?

Minimizing the number of black leaf nodes will result in a better (lossless) compression of the original image, and we

Fig. 6. 1st quadtree representation of I

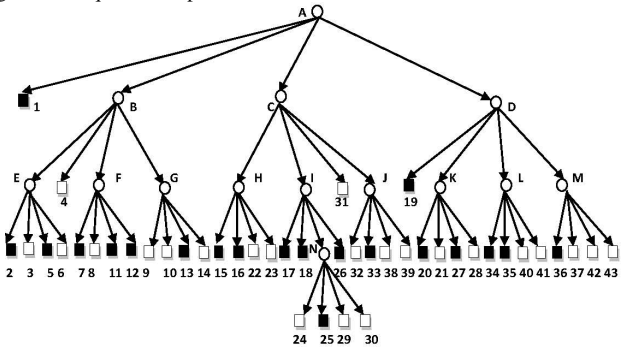
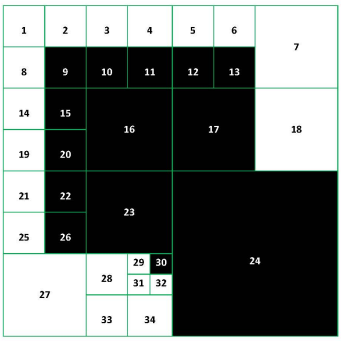


Fig. 7. 2nd decomposition of I



will use this number as a metric. In the next section we will outline a heuristic algorithm which decreases the number of black nodes required to represent an image.

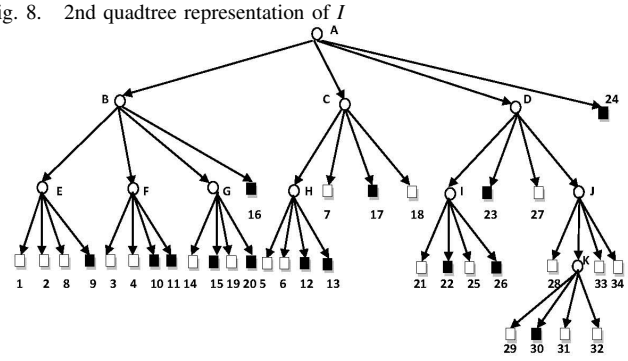
IV. Choosing a root node

For brevity, we call a (closed) square of size $2^k \times 2^k$ containing only black cells a *block*. We say that two blocks s,t are *in contact* if $s \cap t \neq \emptyset$, i.e.

- s and t contain a common cell (Figure 9(a)), or
- s and t contain a common edge of a cell (Figure 9(b)), or
- s and t contain a common corner point (Figure 9(c)).

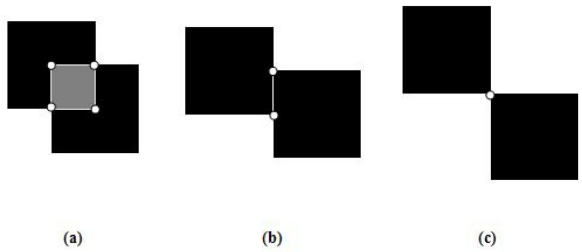
A set S of blocks is called a *contact set*, if $\bigcap S \neq \emptyset$, i. e. the blocks in S have at least one common point of contact.

Fig. 8. 2nd quadtree representation of I



Furthermore, if A is a point (i.e. the intersection of two orthogonal edges), the *neighbor number* n_A *of* A is the sum of the size of all blocks of which A is a corner point.

Fig. 9. Blocks in contact

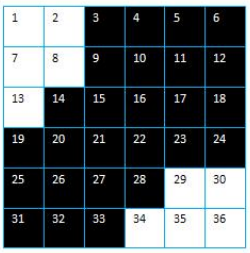


We use three criteria for choosing a root node A:

1) A block of maximal size will be one leaf node in the finale quadtree representation.
2) The size of neighbors of A which are blocks is maximized.
3) The choice of A minimizes the size of the image space $2^t \times 2^t$ containing I.

Suppose that the input consists of a binary array I with a bounding box of dimension $n \times m$; an entry "1" indicates a black cell and an entry "0" indicates a white cell. In describing the procedure we will use the image of Figure 2 as a running example. First we find the bounding box of I (Figure 10).

Fig. 10. Bounding box of I



The cells of the bounding box are scanned from left to right, row by row, starting at the upper left hand corner of the bounding box of I. The CORN[1] algorithm proceeds as follows:

1.
repeat
Progress to the next unvisited black cell b_i and record its position. {In the example, b_1 is cell 3.}

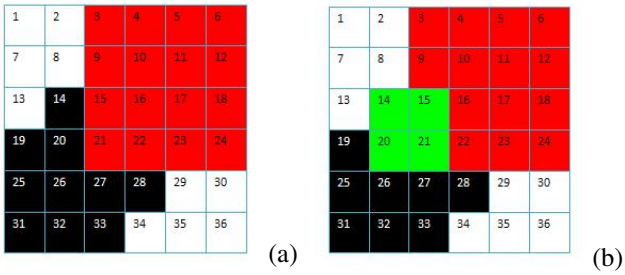
Find a block s_i of maximal size that
1) contains b_i, and
2) maximizes the number of unvisited cells.

{s_1 is shown in Figure 11(a) and s_2 in Figure 11(b).}

Change color of the cells of s_i into, say, red.
until all cells have been visited.

[1]**C**hoosing an **O**ptimal **R**oot **N**ode - we are aware that the name is somewhat optimistic.

Fig. 11. Maximal squares containing b_1, b_2



{At this stage, we have a set $B = \{b_1, \ldots, b_n\}$ of points and a set of blocks $S = \{s_1, \ldots, s_n\}$. In our example, the resulting squares are shown in Table III; the cell b_i is shown in bold face.}

TABLE III
MAXIMAL SQUARES

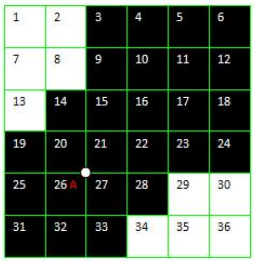
s_1 : {**3**,…,6,9,…12,15,…,18,21,…,24}
s_2 : {**14**,15,20,21}
s_3 : {**19**,20,25,26}
s_4 : {26,**27**,32,33}
s_5 : {21,22,27,**28**}
s_6 : {25,26,**31**,32}

2. Find the largest blocks, say, $s_1, \ldots, s_k$.

3. Find all maximal contact sets $S_1, \ldots, S_m$ from S containing at least one of $s_1, \ldots, s_k$ and for each S_j the set C_i of points in the intersection.

{This can be integrated in the previous part. In the example, there is only one contact set, namely, $S_1 = \{s_1, s_2, s_3, s_4, s_5, s_6\}$; the common point of contact is the north–west corner of cell 27, see Figure 12. Its neighbor number n_A is $1+16+4+1 = 22$ (NW,NE,SW,SE).}

Fig. 12. Common contact C_1



4. Order the points in $\bigcup\{C_i : 1 \le i \le m\}$ by their neighbor numbers in decreasing order, say $A_1, A_2, \ldots, A_k$ such that $n_{A_1} \ge n_{A_2} \ge \ldots \ge n_{A_k}$.

5. Let t be the smallest number such that $n, m \le 2^t$; in other words, a block with side length t is a smallest block that can contain I.
6. Set $i = 0, j = t$.
repeat
repeat

Increase i by 1.
Decompose the image with A_i as a root node. If the resulting block has side length 2^j then choose A_i as root node and stop.
until $i = k$. {At this stage, none of the A_i will allow using 2^j as a side length, so we try the next smallest block.}
Increase j.
until FALSE
The final quadtree decomposition of our example is shown in Figure 13

Fig. 13. CORN decomposition

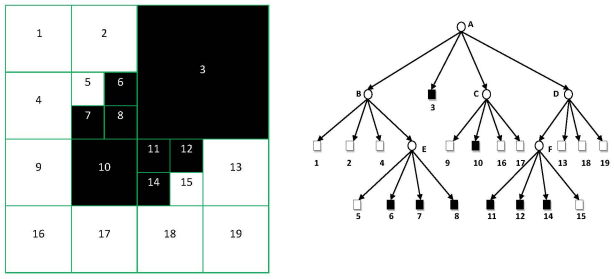


Fig. 14. Example 1

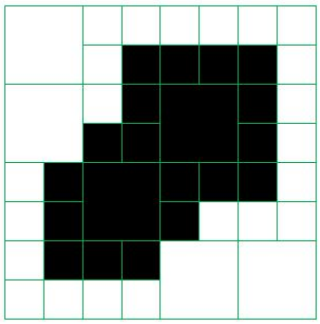
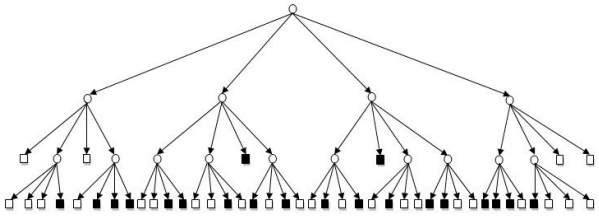


Fig. 15. Example 1



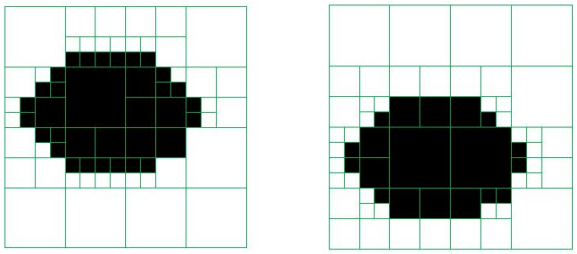
A brief indication about the reduction performance of the choice of the root node is shown in Table IV. Traditionally, the root node is placed in the center of the chosen image area or the image to be considered. Example 1 puts the root node in the center of the black region of Fig. 2 (Fig. 14, 15). The CORN algorithm reduces the number of relevant leaf nodes by more than half.

Figure 16 shows the decompositions of the inner approximation of the ellipse in Figure 1, and Fig. 17 and 19 show the decompositions of a disconnected region.

TABLE IV
LEAFREDUCTION

		Black	White	Total
Example 1	Center (Fig. 14,15)	20	26	46
	CORN (Fig. 13)	8	11	19
Example 2	Center (Fig. 5, 6)	20	23	43
	CORN (Fig. 7,8)	14	20	34
Example 3	Center (Fig. 16(a))	37	30	67
	CORN (Fig. 16(b))	19	45	64
Example 4	Center (Fig. 17,18)	23	35	58
	CORN (Fig. 19)	5	11	16

Fig. 16. Quadtree decompositions of Figure 1



(a) (b)

V. OUTLOOK

The quadtree representation offers a hierarchical decomposition of visual information which can be implemented in a system or a robot. More generally, the quadtree serves well as a representation of a hierarchy of approximation spaces in the sense of Pawlak. Due to the fact that the node building process is triggered by the difference of information and the geometrical structure of the plane, we are able to use the quadtree not only as a coding scheme, but as a simple pyramid representation of image data. Finding the optimal root node

Fig. 17. Standard quadtree decomposition of a disconnected region

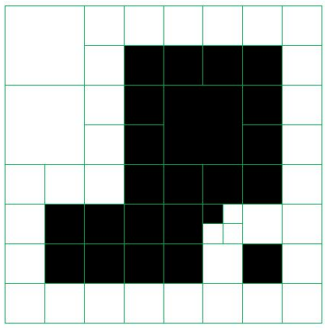


Fig. 18. Standard quadtree decomposition of a disconnected region

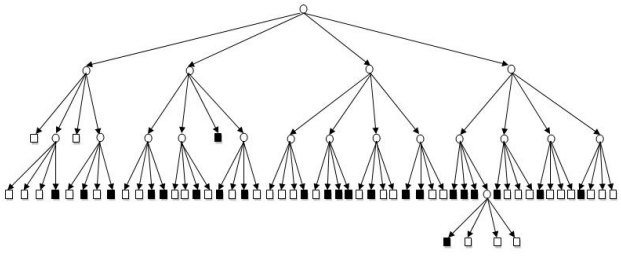
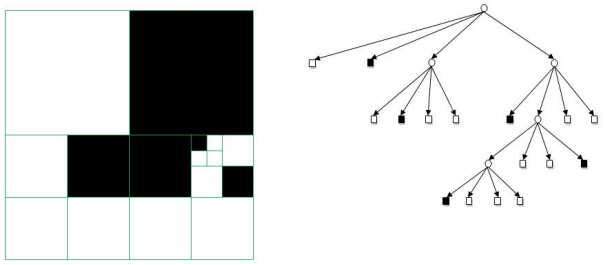


Fig. 19. CORN decomposition of a disconnected region



is an essential tool to offer a good representation of the given image data.

Due the hierarchical geometrical structure of the quadtree in NW-NE-SW-SE-blocks, we are enabled to analyze the information of the data structure at any stage of the hierarchy. The information presented at any stage of the hierarchy are rough sets: If we code 1 for "black", 0 for "white" and ? for "branch" (i.e. "grey") the rough set hierarchy of the first three levels of Figure 8 is given by

```
Level 0:  ?
Level 1:  ???1
Level 2:  ???1 ?010 ?10? 1111
...
```

The quadtree-rough-set representation offers a quick look at the image at an early stage using only a little amount of information. Successive quadtree representation can be used to represent change.

Once again: As the information can be used at any level of the representation, the change information is organized hierarchical rough sets as well.

Our next tasks will be to optimize successive quadtree representations (SQTR) – in particular, to investigate the properties of the CORN algorithm – and to develop a rough logic for change in SQTR. The tasks will be focused on the detection of global chance (e.g. moving environment) and local chance (e.g. moving objects) within successive quadtree representations using the hierarchy of rough set representation. Solving these problems will enable us to formalize machine vision algorithms using a suitable (rough) logic based on quadtrees. As patterns of "approaching" or even "frames to contact" are then (roughly) definable, a first step towards an ecological perception using rough quadtree representations is feasible.

REFERENCES

[1] Z. Pawlak, "Rough sets," *Internat. J. Comput. Inform. Sci.*, vol. 11, pp. 341–356, 1982.

[2] J. J. Gibson, *The ecological approach to visual perception*, 2nd ed. Hillsdale: Lawrence Erlbaum, 1986.

[3] I. Düntsch and G. Gediga, "Towards a logical theory of perception," 2009, in preparation.

[4] J. J. Gibson, "On the new idea of persistence and change and the old idea that it drives out," in *Reasons for Realism – Selected Essays of James J. Gibson*, E. Reed and R. Jones, Eds. Hillsdale: Lawrence Erlbaum Associates, 1982, pp. 393–396.

[5] M. Worboys, "Imprecision in finite resolution spatial data," *Geoinformatica*, vol. 2, no. 3, pp. 257–280, 1998.

[6] A. Muir, I. Düntsch, and G. Gediga, "Rough set data representation using binary decision diagrams," *Revista Real Academia de Ciencias, Serie A*, vol. 98, pp. 197–213, 2004. [Online]. Available: http://www.cosc.brocku.ca/Department/Research/TR/cs0403.pdf

[7] R. Finkel and J. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, pp. 1 – 9, 1974.

[8] G. Proietti, "An optimal algorithm for decomposing a window into maximal quadtree blocks," *Acta Informatica*, vol. 36, no. 4, pp. 257–266, 1999.

[9] C. A. Shaffer and H. Samet, "Optimal quadtree construction algorithms," *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 402–419, 1987.

[10] S. Ranade, A. Rosenfeld, and H. Samat, "Shape approximation using quadtrees," *Pattern Recognition*, vol. 15, pp. 31–40, 1982.

[11] W. G. Aref and H. Samet, "Efficient processing of window queries in the pyramid data structure," in *Proc. of the 9th ACM Symposium on Principles of Database Systems*, 1990, pp. 265–272.

[12] C. R. Dyer, "The space efficiency of quadtrees," *Computer Graphics and Image Processing*, vol. 19, no. 4, pp. 335–348, 1982.

[13] H. Samet, "The quadtree and related hierarchical data structures," *Computing Surveys*, vol. 16, pp. 187–260, 1984.