

THE TESSERAL PROCESSOR FOR IMAGE PROCESSING BASED ON HIERARCHICAL DATA STRUCTURES

Marcel HARAKAL
 Department of Informatics and Computers
 Military Academy of Liptovský Mikuláš
 P. O. BOX 76
 031 01 Liptovský Mikuláš
 Slovak Republic
 E-mail: harakal@valm.sk
 and
 Ján CHMÚRNY
 Military Academy of Liptovský Mikuláš
 031 01 Liptovský Mikuláš
 Slovak Republic

Abstract

This paper describes the design and hardware implementation of the Tesseral Processor (TP) with Programmable Logic Devices (PLD). The TP can be used for image processing based on hierarchical data structures as Linear QuadTree (LQT).

Keywords

quadtree representation, linear quadtree, tesseral addressing, tesseral arithmetic, tesseral arithmetic unit, tesseral control unit, tesseral processor

1. Introduction

Image representation is an important technique in the domains of computer graphics and image processing. From the large range of many representation techniques the hierarchical data structures as quadtree [8, 10] and octree [7] seem to be the most important. The quadtrees are used to represent and to process two-dimensional (2-D) image. Octrees are three-dimensional (3-D) extension of quadtrees.

The quadtree representation is based on hierarchical decomposition of a square picture into four quadrants (Fig. 1). This process is then repeated for each sub-quadrant until a quadrant is reached that is homogeneous. The quadtree is a tree in which each node is either terminal (leaf) or internal (Fig. 2). The terminal node represents a coherent quadrant where every pixel has the same color.

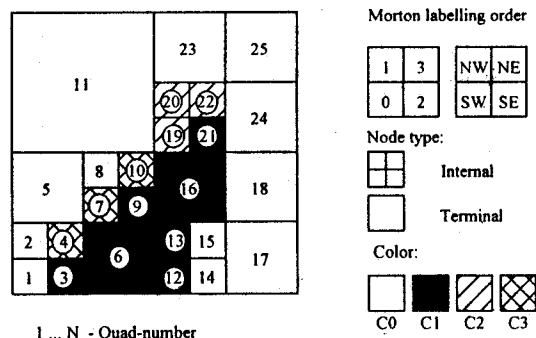


Fig. 1. Recursive decomposition of the picture into quadrants

Internal nodes are not coherent quadrants and are subdivided into further quadrants. The atomic node represents the smallest possible quadrant, corresponding to individual pixels in the picture. Atomic nodes are always leaves and all large nodes are called molecular. An internal node (called a parent) is sub-divided into four other nodes, these are called its children. The root node represents the whole picture. The depth of a node in the quadtree is the number of its ancestors. The root node has the depth 0. The atomic depth a of picture specifies how big picture can be represented. The picture depth n implies the picture size $2^n \times 2^n$.

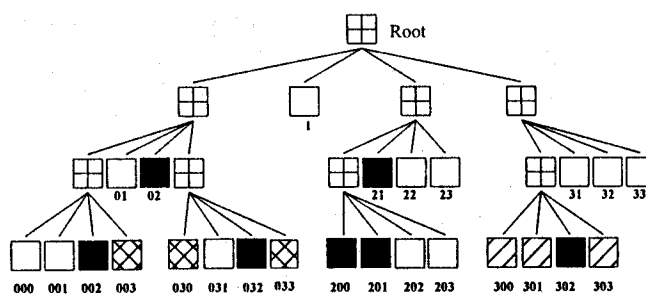


Fig. 2. The quadtree and quaternary labelling of quadtree nodes

A well-known form for encoding of pictures is the linear quadtree [2]. The linear quadtree is a hierarchical data structure consisting of a pointerless list which contains only terminal nodes in the quadtree. Internal nodes are omitted as of no interest. The terminal nodes in the linear quadtree are represented by locational codes [4] which identify their position in the tree. Because each terminal node is individually identified, it is possible to remove all nodes of a particular color (the background color). Nodes include the identity of each nodal information which has certain advantages. The code may be disordered without the

loss of meaning. For image processing by LQT we may use tesseral theory [1].

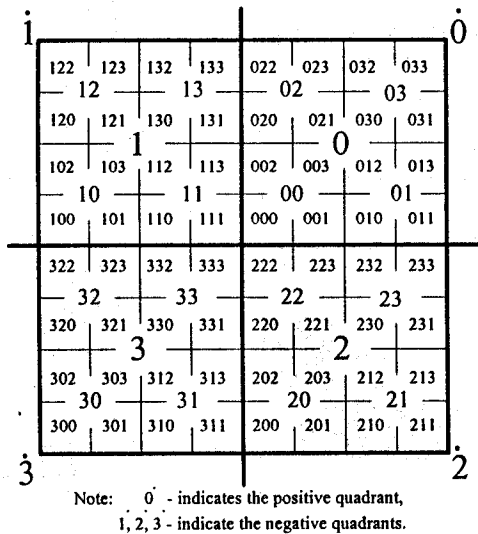


Fig. 3. The tesseral addressing structure

2. The Tesseral Theory

The tesseral theory [1] is very interesting for computer processing of data in both 2 and 3 dimensions. The tesseral theory is based on an approach to image processing as of groups of tesseral elements. The tesseral elements are identically shaped elements. In the processes whereby tesseral elements are divided or combined into larger elements, new elements with the same shape are produced. The theory contains addressing theory and tesseral arithmetic. Addressing theory is important for labelling or ordering quadrants in linear quadtree [1, 3, 4]. The importance of tesseral arithmetic lies in the possibility of mapping spatial geometric transformations into arithmetic operations. Conventional computer architectures do not support tesseral arithmetic operations directly as single CPU (Central Processor Unit) instructions. This means that tesseral operations must be realised using software which often involves a large number of sequential operations. The tesseral arithmetic for rapid real-time image processing is ideally suited for implementation in hardware.

2.1 Tesseral addressing

In modern computer systems the picture representation is based on raster images, that is two-dimensional arrays of pixels. Each pixel has a colour value and Cartesian co-ordinate value. For co-ordinate value we may use a tesseral address [1]. The word tesseral derives from the Latin "tesserae" - small tile used in the construction of mosaic patterns. The whole raster graphics is a form of tessellation. Tessellation is the division of n-dimensional space into identically shaped and sized

primitive elements. Useful tessellations for 2-dimensional space use squares and hexagons, for 3-dimensional space use cube [7].

Tiling is the process whereby the tesseral elements are combined into larger elements with the same shape. The original elements are called atomic tiles, the larger ones molecular tiles. The molecular tiles can be themselves combined into even bigger ones. Tesseral image T_0 is a group of molecular and atomic tiles.

In the tessellation and tiling processes we must determine how many tiles are to be grouped together to form a larger one and determine their position in space. Rather than labelling atomic tiles by a Cartesian co-ordinate value, we may use a tesseral address. The tesseral address is a quaternary (base 4) number giving the position of the atomic tile in the tiling hierarchy (Fig. 3). Molecular tiles in the tiling hierarchy may be addressed by giving their size and the address of any constituent atomic tile. The tesseral addressing requires an ordering on the atomic tiles in Morton order. The mapping between tesseral addresses and Cartesian co-ordinates of the same objects is very simple by bitwise interlace. Tesseral to Cartesian address conversion is performed by considering each successive tesseral digit in its binary form and partitioning the result such that the first digit of each pair goes to form the y and the second digit the x Cartesian co-ordinates. Example:

$$1304 = 011100_2 \Rightarrow (2, 6) = (010_2, 110_2)$$

$$\begin{array}{r}
 010 \Rightarrow y = 2 \\
 \nearrow \uparrow \nwarrow \\
 1304 = 011100 \\
 \searrow \downarrow \swarrow \\
 110 \Rightarrow x = 6
 \end{array}$$

The practical upshot of tesseral addressing is that tesseral operations directly on Cartesian co-ordinates or Cartesian operations directly on tesseral co-ordinates can be performed, with very simple bitwise operations. Tesseral addressing, besides, may be used for labelling linear quadtree nodes [1, 5].

2.2 Tesseral Arithmetic

The importance of tesseral arithmetic lies in the possibility of spatial operations (translation, scaling, rotation) mapping into arithmetic operations. For example, in the case to be described in the paper, addition produces a translation of the tesseral image and multiplication results into scaling and rotation. The tesseral arithmetic is different than linear arithmetic. The definitions of tesseral arithmetic operations are in [5]. Addition of two tesseral numbers T_a and T_b results into a translation of the first number T_a by an amount equal to the vector from 0 to the second number T_b (the vector $0T_b$). Subtraction of tesseral numbers T_a and T_b is a translation of the first number T_a by an amount equal to the vector from the second number T_b to the number 0 (the vector T_b0). Performing a subtraction involves finding the

negative of the tesseral number T_b to be subtracted and then addition of this number to T_a .

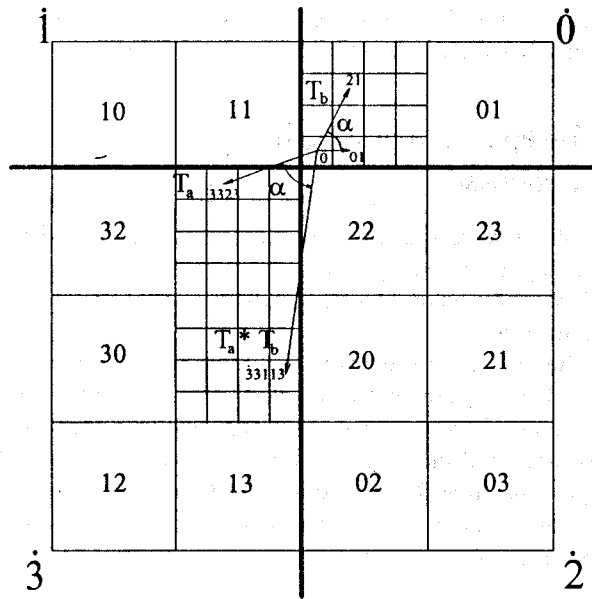


Fig. 4. Geometry of tesseral addition

Multiplication of tesseral numbers $T_a * T_b$ results into tesseral number, which is interpretation of vector OT_a , rotated through an angle α and multiplied (scaling) by coefficient S . S and α are results of transformation of vector 01 into vector OT_b . For example (Fig. 4) the multiplication of tesseral numbers $323 * 21 = 33113$ is shown. Tesseral division is an inverse of multiplication and can be defined as a rotation in the "other way" and scaling down rather than up. Division of tesseral numbers T_a / T_b results into tesseral number, which is interpretation of vector OT_a rotated through an angle β and multiplied by coefficient S . S and β are results of transformation of vector OT_b into vector 01 . Division also gives rise to fractional addresses on the Euclidean plane below the atomic level of the tiling. Combinations of these basic operations can be used to all standard image operations (scaling, zooming and rotation).

3. The Tesseral Processor

The tesseral arithmetic for rapid real-time image processing is ideally suited for implementation in hardware as new architecture - tesseral processor [6, 8, 9]. The TP is synchronous digital system. The architecture of TP (Fig. 5) consists of 2 basic parts: Tesseral Control Unit (CL_REG) and Tesseral Arithmetic Unit (TAU).

The CL_REG consists of internal memory and Control Unit (CU). The internal memory of CL_REG consists of a member of registers in which information can be stored. The SREG register, CT register and CREG register are status and control registers. The NL, NC, SC and RV registers are general registers for tesseral operands. The CU of CL_REG consists a set logic circuits that make it possible to perform certain operations on the information stored in the registers and control of TAU.

The TAU performs tesseral arithmetic operations (addition, subtraction, multiplication and division) on input data directly. The TAU block diagram is shown in Fig. 6. It consists of 4 basic parts: a non standard multiplexer (MUX), two full adders with carry in and carry out (S1 and S2) and internal Control Logic (CL).

The methods for performing tesseral arithmetic use "rule based methods"[5]. In the first step of this method tesseral operands are split into register pairs T_x and T_y in binary form. These two register pairs represent the Cartesian components of the tesseral number. The second step of the method is based on linear arithmetic operations

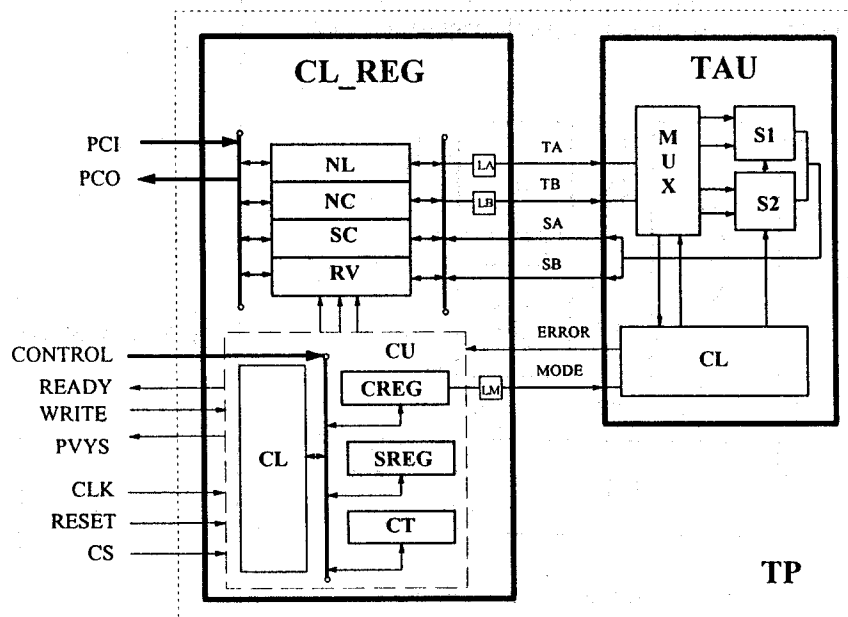


Fig. 5. The tesseral processor block diagram

and these convert the result to tesseral number. Addition of tesseral operands is accomplished by performing normal

binary addition with each $T_{ax} + T_{bx}$, $T_{ay} + T_{by}$ register pair and then registers can be recombined to yield the tesseral sum. In subtraction, the second operands T_{bx} , T_{by} are 2' complemented (T_{bx}^{\oplus} , T_{by}^{\oplus}) and added to the register pair $T_{ax} + T_{bx}^{\oplus}$ and $T_{ay} + T_{by}^{\oplus}$. Tesseral multiplication is performed using 4 rules:

- Rule 1: Multiplication by 0 - Both register pairs are set to 0.
- Rule 2: Multiplication by 1 - Both register pairs are unchanged.
- Rule 3: Multiplication by 2 - The register is 2' complemented to T_y^{\oplus} and T_x and T_y registers are then exchanged.
- Rule 4: Multiplication by 3 - Multiplication is dependent on the observation that $3 = (2+1)$ and that $(T_y*2)+(T_y*1)$. This is accomplished by normal binary addition of the original T_x and T_y register contents to their contents after execution of rule 3.

These rules determine the procedure for multiplying any tesseral number by a single tesseral digit. The selected rule depends on the value of the multiplier digit. To multiply multidigit tesserals a procedure similar to decimal multiplication is employed, with successive multiplication and product evaluation for each of the multiplier digits.

Tesseral division by the single digit 0, 1 and 2 is similar to multiplication rules:

- Rule 1: Division by 0 - Produces an error.
 - Rule 2: Division by 1 - The registers pair T_x and T_y remain unchanged.
 - Rule 3: Division by 2 - The T_x register is 2' complemented to T_x^{\oplus} and then T_x and T_y are exchanged.
- Division by 3 is implemented using the tabular methods.

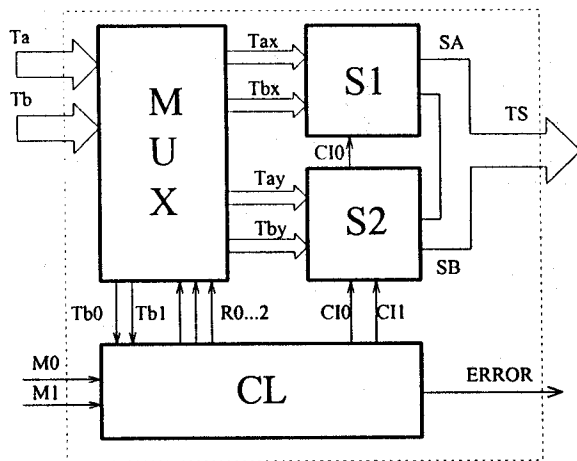


Fig. 6. The tesseral arithmetic unit block diagram

The design of TP is realized by high-level expressions of ABEL-HDL language [8]. The basic program structures of design module TP in ABEL-HDL translates into an industry standard JEDEC fuse maps file for programming PLD. Before programming PLD we use simulation of TP model. This approach takes advantage,

creates high-quality logical design in an abstract form in less time.

The designed TAU and CL_REG of TP are implemented with Xilinx Complex Programmable Logic Devices (X-CPLD). The speed of the TP is 54 MHz.

4. Image Processing by Tesseral Processor

The new designed tesseral processor was applied for processing of color, gray and thermovision images. To verify the TP we consider the dimension of pictures 16x16 and 256x256.

Before image processing a new method of encoding of the linear quadtree by standard locational codes into color levels [4] was applied in the first step. For improving of the gray and thermovision image encoding we applied adaptive quadtree decomposition determined on the basis of the mean square error obtained between the original and its reconstructed image [8].

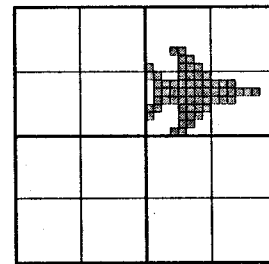


Fig. 7. The original image T_0 of F-15 aeroplane

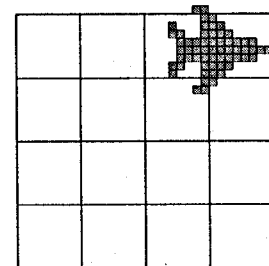


Fig. 8. The tesseral addition of F-15 aeroplane ($T_0 + 0321$)

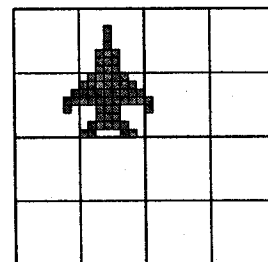


Fig. 9. The tesseral multiplication of F-15 aeroplane ($T_0 * 2$)

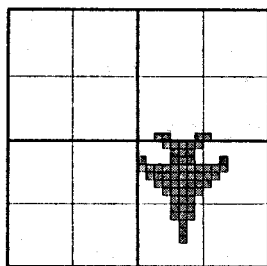


Fig. 10. The tesseral division of F-15 aeroplane ($T_0/2$)

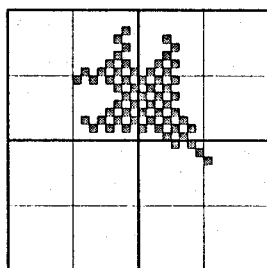


Fig. 11. The tesseral multiplication of F-15 aeroplane ($T_0 \cdot 2221$)

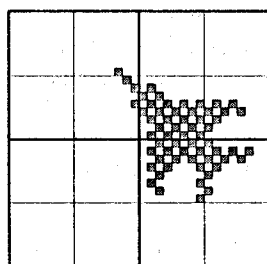


Fig. 12. The tesseral multiplication of F-15 aeroplane ($T_0 \cdot 1112$)

Next, in the second step we applied the tesseral processor on encoded images. The previous coded images T_0 as sequences of tesseral operands are transferred from central computer and loaded into tesseral processor. The tesseral processor realizes elementary tesseral operations on operands and deliver the results to the central computer. For example, the total time for elementary tesseral operation to gray image LENA 256x256 dimension was 2,1751ms. Figure 7-12 shows how tesseral arithmetic can be utilized for image processing.

5. Conclusion

The hierarchical data structures as linear quadtree are very efficient forms used for representation and processing of pictures. Image processing by linear quadtree includes recursion and composition processes which are demanding for computer time and memory. To simplify these processes we focused on new architectures, design and hardware implementation of tesseral processor with programmable logic devices. The designed TP gives new possibilities of digital real-time image processing.

References

- [1] BELL, S. B. M.-DIAZ, B. M.: Spatial Data Processing Using Tesseral Methods (Collected Papers from Tesseral Workshop 1 and 2: Natural Environment Research Council, Polaris House, Swindon (1986).
- [2] GARGANTINI, I.: An Effective Way to Represent Quadtrees, Communications of the ACM, 25, No. 12 (1982), 905-910.
- [3] COENEN, F. P. - BEATTIE, B. - DIAZ, B.M. - BENCH-CAPON, T.J.M. - SHAVE, M.J.R: Temporal Reasoning Using Tesseral Addressing: Towards an Intelligent Environmental Impact Assessment System, Journal of Knowledge-Based Systems, 9, No. 5 (1996), 287-300.
- [4] HARAKAL, M. - CHMÚRNÝ, J.: Locational Codes for Encoding of Color Pictures, Journal of Electrical Engineering, 44, No. 9 (1993), 267-270.
- [5] HARAKAL, M. - CHMÚRNÝ, J.: Tesseral Arithmetic in Hardware Implementation, Journal of Electrical Engineering 46, No. 2. (1995), 41-45.
- [6] HARAKAL, M.- CHMÚRNÝ, J.: The Tesseral Processor for Image Processing, Proc. of Int. Conf. TELECOMMUNICATION '96, Bratislava, June 1996, 162-167.
- [7] HARAKAL, M. - CHMÚRNÝ, J.: Image Generation by Octree, Radioengineering, 5, No. 3 (1996), 1-5.
- [8] HARAKAL, M.: The Tesseral Processor for Image Processing Based on Hierarchical Data Structures, Ph.D. Dissertation, Department of Informatics and Computers, Military Academy of Liptovský Mikuláš, April 1997. (in Slovak).
- [9] HARAKAL M. - CHMÚRNÝ, J.: The Tesseral Processor for Image Processing, Proc. of Int. Conf. DIGITAL SIGNAL PROCESSING '97, Košice-Herfany, September 1997, 57-59.
- [10] SAMET, H.: The Quadtree and Related Hierarchical Data Structures, ACM Computing Survey 16, No. 2 (1984), 188-260.

About authors ...

Marcel Harakal' was born in Bratislava, Slovak Republic, on November 13, 1958. He received Ing. (MS) degree in electrical engineering from Faculty of Electrical Engineering Slovak Technical University in 1983. Now he is taking part in CSc (PhD) in the field of digital image processing at Department of Informatics and Computers at the Military Academy of Liptovský Mikuláš. His research interests include digital image processing, digital electronics and microprocessor applications.

Ján Chmúrny was born in Martin, Slovakia, on June 7, 1923. He received his undergraduate Diploma degree in electrical engineering from the Slovak Technical University of Bratislava in 1947. The CSc (PhD) degree in Radioelectronics from the Moscow Technical University in 1954, and the DrSc degree in Radioelectronics from the Technical University of Brno (Czech republic). He is currently an Professor of Electrical Engineering in Military Academy of Liptovský Mikuláš. His research interests are in the areas of image processing, coding and mobile communication systems.