IMAGE GENERATION BY OCTREE

Marcel HARAKAĽ
Department of Informatics and computers
Military Academy of Liptovský Mikuláš
031 19 Liptovský Mikuláš
Slovak Republic
E-mail: harakal@perun.valm.sk
and
Ján CHMÚRNY
Military Academy of Liptovský Mikuláš
031 19 Liptovský Mikuláš
Slovak Republic

Abstract

This paper deals with the generating of three-dimensional (3-D) images by octree and using the application of tesseral theory in image processing.

Keywords

octree representation, linear octree, tesseral addressing, tesseral arithmetic

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, 11] and octree [5] seem to be the most important. The quadtrees are used to represent and to process two-dimensional (2-D) image. Octrees are three-dimensional extension of quadtrees. Octrees have been used to represent 3-D solid objects [6, 9, 12].

The 3-D representations of solid objects create the base for computer graphics, development of robotics and other vision systems. The most widely used representation schemes are following [1, 13]: Boundary representation (B-Rep), Constructive Solid Geometry (CSG) and Octree representation.

A solid object as 3-D volume encoded by a set of surface elements is defined in the B-Rep method (see figure 1). Sections of planes and quadratic surfaces as spheres, cylinders and cones are typical surface elements. A graph structure is used to express the relations among the surface elements. The nodes representing surface elements are connected to other sharing common boundary edges and to nodes representing the boundary edges which are, in turn, connected to representing vertex points.

In the CSG method, volume elements rather than surface elements are used to bind 3-D volumes. Common volume elements are blocks, prism, spheres, cylinders, cones and tori. These elements are combined by set operations into the modeled objects. The CSG tree structure is used to relate the volume elements. In such a tree, leaf nodes represent the volume elements, branch nodes represent the set operations, and the connecting arcs represent 3-D geometric transformations.

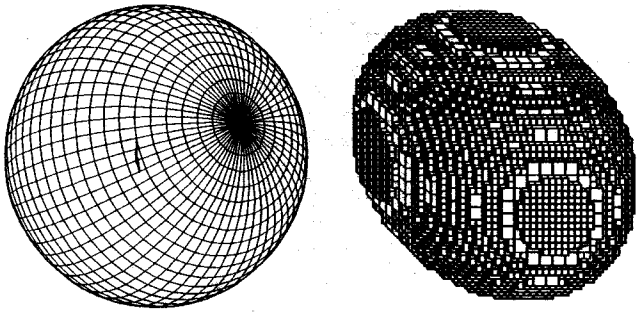


Fig. 1. Boundary and octree representations of the 3-D object (ball)

An octree representation of 3-D object is the tree structure describing selective recursive subdivision of object space. Object space can be represented by tree structure where the root node represents whole object space and 8 descendant nodes represent 8 primary subdivisions of object space into 8 equal volumes or cubes. Further levels in the tree structure are related to recursive subdivisions of space. A node will have descendents only if associated volume of object space is not homogeneous. Recursion continues until either all the nodes are homogeneous or until the resolution is achieved. A important form octree of 3-D solid object representation is the linear octree [4]. For encoding and image processing by linear octree we can use tesseral theory [2]. The tesseral theory is very interesting for computer processing of data in both 2 and 3 dimensions. The tesseral theory contains addressing theory and tesseral arithmetic.

2. Linear octree

Let's have a 3-D digital image O. We shall assume that image O is sampled at $\{(i+1/2, j+1/2, k+1/2) \in X \times Y \times Z;\ i, j, k$ integers in $<0, 2^n)\}$, where (X, Y, Z) is 3-D Cartesian co-ordinate system in which the image O is defined. The sample points create a lattice L. It is given a 3-D lattice point $(\underline{x}, \underline{y}, \underline{z}) \in L$. The unit volume $\{(x, y, z) \in X \times Y \times Z;\ \underline{x}-1/2 \le x < \underline{x}+1/2,\ \underline{y}-1/2 \le y < \underline{y}+1/2,\ \underline{z}-1/2 \le z < \underline{z}+1/2\}$ is called the voxel which is centred at $(\underline{x}, \underline{y}, \underline{z})$. The union of a set of voxels is 3-D digital object [14].

An octant of size 2^m for some integer $m \geq 0$ is semi-close cubic volume $\{(x, y, z) \in X \times Y \times Z;\ 2^m i \leq x < 2^m(i+1),\ 2^m j \leq y < 2^m(j+1),\ 2^m k \leq z < 2^m(k+1);\ i, j, k$ integers in $<0, 2^{n-m})\}$, where n is an integer such that the volume of interest is contained in an octant of size 2^n. An octant of size 1 is voxel. We say that an octant of size 2^m is a level m octant. The origin of an octant is the point in the octant with the smallest co-ordinates. A level m octant consists of 8 level $(m - 1)$ sub-octants. There are numbered 0, 1, 2, 3, 4, 5, 6 to 7. The value of an octant is 0 (VOID) or 1 (FULL).

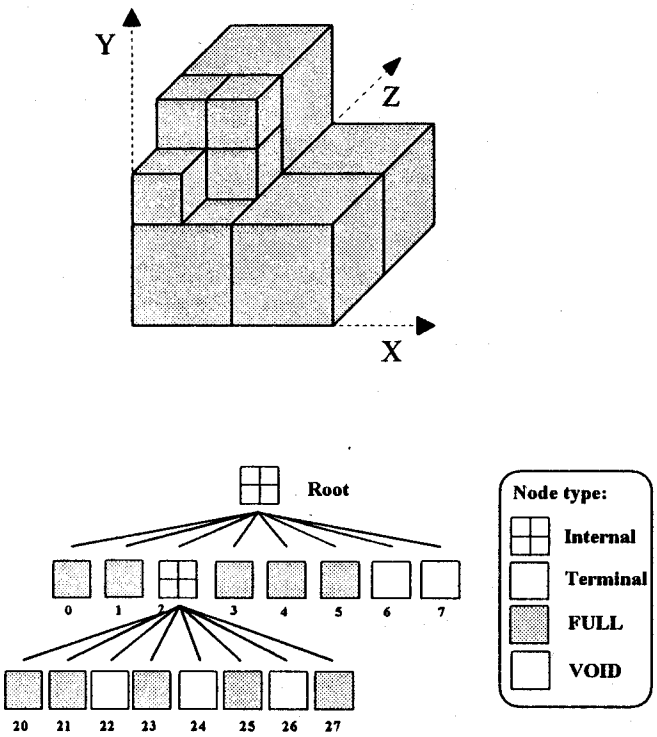


Fig. 2. Octree representation of 3-D object

We will consider the 3-D digital image which contains $2^n \times 2^n \times 2^n$ array of volume elements-voxels and its hierarchical decomposition into octants. This process is then repeated for each sub-octant until an octant is not homogeneous. The process is represented by a tree of degree 8 in which the root node represents the entire image, and the leaf nodes correspond to those cubes of array for which no further sub-division is necessary. The octree is tree in which each node is either terminal (leaf) or internal. The terminal nodes represent a coherent octants where every voxel has the same value (VOID, FULL). Internal nodes are not coherent octants and are sub-divided into further octants. The atomic node represents the smallest possible octant, corresponding to individual voxel in the picture. Atomic nodes are always leaves and all large nodes are called molecular nodes. An internal node (called parent) is sub-divided into 8 other nodes, these are called its children. The root node represents the whole picture. The depth of node in the octree is the number of its ancestors. The root node has depth 0. The atomic depth of picture specifies how big 3-D picture can be represented. A full octree representing the 3-D image has

$$N_{OT} = \sum_{i=0}^{n} 2^i \times 2^i \times 2^i = 1 + 8 + 8^2 + \ldots + 8^n = \frac{8^n + 1}{7} \quad (1)$$

nodes.

A well-know form of the 3-D solid object representation is the linear octree [4]. The linear octree is a hierarchical data structure consisting of a pointerless list which contains only terminal nodes in octree. Internal nodes are omitted as of no interest. Linear octree is designed primarily for binary pictures. All FULL leaves are stored in octree, VOID nodes are omitted. The terminal nodes in the linear octree are labelled by locational codes [5] which identify their position in the tree. Because each terminal node is individually identified, it is possible to remove all nodes of value VOID (the supplement of 3-D object). Nodes include the identity of each as nodal information which has certain advantages. The code may be disordered without the loss of meaning.

Locational codes are important labels for octree nodes. The codes give the position of the octree node in numerical form. A particular numerical label (number) is assigned to each terminal node depending on the direction of the route trough the octree from the root to node (Fig. 2). This gives an octal number. The number of digits in the octal number corresponds to the depth of the node in the tree. The variable length of codes cannot be stored directly into the memory system. Therefore we must also separate the code using an additional digit "X" as a postfix in place of irrelevant digits.

3. Tesseral addressing and tesseral arithmetic

For image processing by linear octree the tesseral theory [3, 7] we can be used. The tesseral theory is based on an approach to image processing as of groups of tesseral elements (tiles). The tesseral elements are identically shaped elements. For 3-D space we use cubes that correspond to voxel in the octree. 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 is a group of molecular and atomic tiles.

In the division of 3-D space into identically shaped elements and combination of tesseral elements into large elements 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 for 3-D space is an octal (base 8) number giving the position of the atomic tile in the tiling hierarchy (see figure 3).

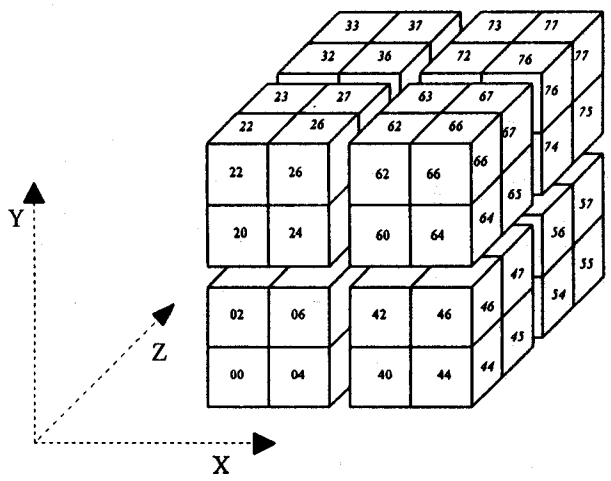


Fig. 3. Labelling of elements in linear octree

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. Example:

$$45_8 = 100\ 101_2 \Rightarrow (3, 0, 1) = (11_2, 00_2, 01_2)$$

$$45 \left\{ \begin{matrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{matrix} \right. \begin{matrix} x = 11_2 = 3 \\ y = 00_2 = 0 \\ z = 01_2 = 1 \end{matrix}$$

The practical result 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 octree nodes.

The importance of tesseral arithmetic consists in the possibility of spatial operations (translation, scaling, rotation) mapping into arithmetic operations. For example, addition produces a translation of the tesseral image and multiplication result into scaling and rotation. The tesseral arithmetic is differ from linear arithmetic.

The definitions of tesseral arithmetic operations for 2-D image [2] we can apply to 3-D space. 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). The multiplication and division of tesseral numbers simultaneously realise scaling and rotation of tesseral object. The multiplication and division are very complicated and we must apply it in tesseral hypercube. The tesseral hypercube corresponds to 3-D Cartesian co-ordinates system for negative octants.

For performing tesseral addition and subtraction of any elementary digits we can use the tesseral addition and subtraction table. The multi-digit addition is performed by addition table lookup and by carry as well as in ordinary arithmetic. The subtraction of tesseral number is complicated and performed by the method used for P-Adic fields [3].

Table 1. Octree tesseral digit addition

+	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	01	10	03	12	05	14	07	16
2	02	03	20	21	06	07	24	25
3	03	12	21	30	07	16	25	34
4	04	05	06	07	40	41	42	43
5	05	14	07	16	41	16	43	52
6	06	07	24	25	42	43	60	61
7	07	16	25	34	43	52	61	70

Table 2. Octree tesseral digit subtraction

	0	1	2	3	4	5	6	7
0	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
1	1,1	0,0	1,3	0,2	1,5	0,4	1,7	0,6
2	2,2	2,3	0,0	0,1	2,6	2,7	0,4	0,5
3	3,3	2,2	1,1	0,0	3,7	2,6	1,5	0,4
4	2,2	4,5	4,6	4,7	0,0	0,1	0,2	0,3
5	5,5	4,4	5,7	4,6	1,1	0,0	1,3	0,2
6	6,6	6,7	4,4	4,5	2,2	2,3	0,0	0,1
7	7,7	6,6	5,5	4,4	3,3	2,2	1,1	0,0

Note: The row index of table octree subtraction is the T_a digit and the column index the T_b digit. The required digit x of result T_x and associated carry *cy* which have to be added to the remaining digits of T_a digit is in Table 2.

Suppose that the T_a and T_b are know tesseral numbers and T_x is unknown result. We will find tesseral addition and multiplication of two tesseral numbers T_a and T_b in the form $T_a + T_b = T_x$ and $T_a * T_b = T_x$ and the tesseral subtraction $T_b - T_a$ in the form $T_a + T_x = T_b$ and tesseral division T_b/T_a in the form $T_a * T_x = T_b$. We will find the digits of T_x result from right to left, starting with the least significant digit of the T_x. The application of P-Adic method for tesseral numbers we show at the subtraction of tesseral numbers $T_b - T_a = T_x$. We will find the results in the form

$$T_a + T_x = T_b. \qquad (2)$$

Suppose that the last digit of T_a is 2 and the last digit of T_b is 0. We require the last digit of T_x to be such one that when we add it to 2 we get the result 0. A glance at the addition table (Tab. 1) shows that we must make the last digit of T_x equal to 2 in order for this to be true. The last digits of T_a, T_x and T_b then will be canceled, and we can continue and find the penultimate digit of T_x' from the equation

$$T_a' + T_x' + 2 = T_b' \quad (3)$$

or $$(T_a' + 2) + T_x' = T_b' \quad (4)$$

or, if $T_a' + 2$ is written as T_a'' then

$$T_a'' + T_x' = T_b'. \quad (5)$$

It can be seen that equation (5) is of the same form as equation (2), and we can proceed to solve it in the same way. The process ceases when we obtain an equation

$$T_a''' + T_x''' = T_b''', \quad (6)$$

which is identical with some equation previously found $T_a''' + T_x''' = T_b'''$. (7)

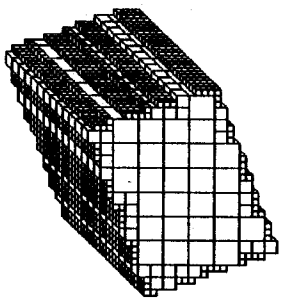
When sequence of digits found for T_x between equations (7) and (6) will repeat indefinitely, the value of all digits of T_x has been found. To automate the subtraction process we apply subtraction table. We explain the P-Adic method on subtraction of tesseral numbers 642 - 6530. The result we find in form $642 + T_x = 6530$. From the subtraction table lookup we can find that the least significant digit (2 - 0) is (2, 2). The least significant digit of T_x = ... 2 and the carry is 2. We obtain the equation in form $(64 + 2) + T_x' = 653$. The processes proceed in the same way

$$\begin{aligned}
66 + T_x' &= 653 \quad 6 - 3 \Rightarrow (4,5)\\
&\qquad T_x = \ldots 52\\
(6 + 4) + T_x'' &= 65 \quad 2 - 5 \Rightarrow (2,7)\\
42 + T_x' &= 65 \quad T_x = \ldots 752\\
(4 + 2) + T_x''' &= 6 \quad 6 - 6 \Rightarrow (0,0)\\
6 + T_x''' &= 6 \quad T_x = \ldots 0752\\
0 + T_x''' &= 0 \quad 0 - 0 \Rightarrow (0,0)\\
0 + T_x'''' &= 0 \quad T_x = \ldots 00752.
\end{aligned}$$

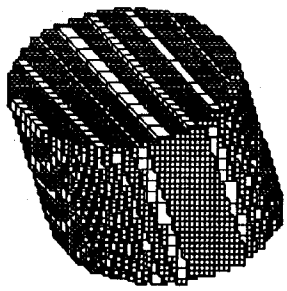
The forms of equations now repeat identicaly and the result is 752. We can also extend the P-Adic method to tesseral division.

4. Modelling of real objects

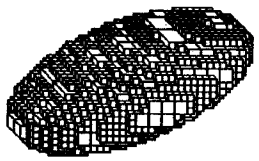
To represent the experiment of 3-D images by 5-7 degree octree the program in language C^{++} has been suggested. This program generates basic objects (prism, spheroid, cylinder, cone) and provides their geometric transformations (Fig. 4). The program enables to combine the basic objects by set operations (union, intersection, difference and negation). Each object is specified by the parameters (e.g. length d_x, radius r, high v, etc.) which have been used for its generation. The angle α determines rotation of the object round the axis x, the angle β round the axis y and γ round the axis z.



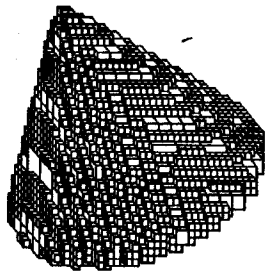
a, Cone: dimensions
$d_a = 330$, $d_b = 400$,
$d_c = 444$,
rotations $\alpha=0$, $\beta=20$, $\gamma=0$.



b, Cylinder: dimensions
r = 330,
v = 400,
rotations $\alpha=0$, $\beta=20$, $\gamma=0$.

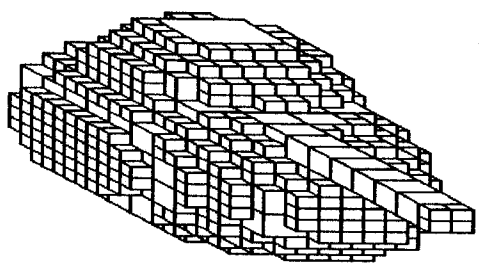


c, Spheroid: dimensions
$d_a = 300$, $d_b = 150$,
$d_c = 80$,
rotations $\alpha=0$, $\beta=25$, $\gamma=0$.



d, Cone: dimensions
r = 320,
v = 400,
rotations $\alpha=10$, $\beta=25$, $\gamma=0$.

Fig. 4. Basic objects for 3-D images generating by octree of degree 6



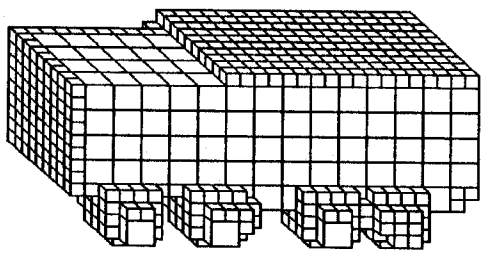


Fig. 5. 3-D model of T-55 tank and T-815 automobile by octree of 5 degree

At the figure 5 we can see the 3-D real objects models which are represented by degree 5 octree combined from basic objects by set operations. The model of T-55 tank is created by the union of 3 basic objects and the model of T-815 automobile by 10 basic objects. For approaching of the model to real objects we can use the octree of high degree (see figure 6) but it is demanding for computer memory and time.

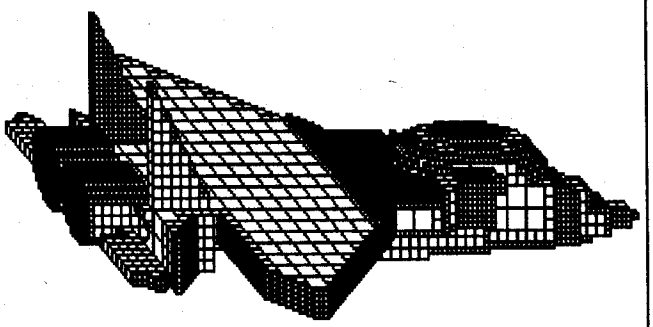


Fig. 6. 3-D model of F-15E aeroplane by octree of 7 degree

5. Conclusion

In this paper we have focused on the new technique of generating of 3-D images by linear octree. For improving of the image processing we use the tesseral addressing and tesseral arithmetic. Image processing by octree includes recursion and composition processes [5, 11] which are demanding for computer time and memory. Demands for the processing are increasing proportionally with the degree of used octree (see equation (1)). To simplify these processes we must look for new methods and for specialised architecture in hardware [7].

References

[1] ANAND, S. - KNOTT, K.: An Algorithm for Converting the Boundary Representation of a CAD Model to Its Octree Representation, Computers and Industrial Engineering, **21,** No. 1-4 (1991), 343 - 347.

[2] 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).

[3] BELL, S. B. M. - MASON, D. C.: Tesseral Quaternions for the Octree, The Computer Journal, **33,** No. 5 (1990), 386-397.

[4] GARGANTINI, I.: Linear Octrees for Fast Processing of Three-Dimensional Objects, Computer Graphics and Image Processing, **20** (1982), 365 - 374.

[5] HARAKAĽ, M.: Spracovanie obrazu pomocou hierarchických stromových štruktúr, Rigorózna práca, Liptovský Mikuláš, marec 1995.

[6] CHMÚRNY, J. - BAČÍK, M.: Reprezentácia trojrozmerných obrazov oktantovým stromom, Elektrotechnický časopis **37**, č.2 (1986), 159-162.

[7] CHMÚRNY, J.- HARAKAĽ, M.: Tesseral Arithmetic in Hardware Implementation, Journal of Electrical Engineering, **46,** No. 2. (1995), 41-45.

[8] JACKINS, CH. L. - TANIMOTO, L.: Oct-Trees and Their Use in Representing Three-Dimensional Objects, Computer Graphics and Image Processing, **14,** (1980), 249 - 270.

[9] MEAGHER, D.: Geometric Modelling Using Octree Encoding, Computer Graphics and Image Processing, **19,** (1982), 129 - 147.

[10] POTMESIL, M.: Generating Octree Models of 3D Objects from Their Silhouettes in a Sequence of Images, Computer Vision, Graphics and Image Processing, **40,** (1987), 1 - 29.

[11] SAMET, H.: The Quadtree and Related Hierarchical Data Structures, ACM Computing Survey, **16,** No. 2 (1984), 188-260.

[12] TAMMINEN, M. - KARONEN, O. - MÄNTYLÄ, M.: Ray-Casting and Block Model Conversion Using a Spatial Index, Computer Aided Design, **16,** No. 4, (1984), 203 - 208.

[13] WILHELMS, J. - GELDER, A.: Octrees for Faster Isosurface Generation, ACM Transactions on Graphics, **11,** No. 3 (1992), 201-227.

[14] YAMAGUCHI, K. - KUNII, T. L.: A Layered String Data Structure for an Octree Model, Technical Report 83-15, Department of Information Science Faculty of Science, University of Tokyo (1983).

About autors, ...

Marcel Harakal' was born in Bratislava, Slovak Republic, on November 13, 1958. He received Ing. (MS) degree in electrical engeneering 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.