Bit-paralleling geometrical algorithms

Vyacheslav M.Khachumov

Program (software) systems institute RAS
152140 Pereslavl-Zalessky, Russia
e-mail: vmh@vmh.botik.ru

Abstract.

Consideration is being given to problem of the geometrical algorithms parallel calculations, which based on Volder and Meggitt algorithms modification. Carefully analyzed are the parallel calculations and correction. Modeling results are given.

Key words:

geometrical processors, Volder and Meggitt algorithms, rotation, vectoring, parallel calculations, bit-paralleling form, group summation, correcting angels.

1. Introduction

Specialized graphical (geometrical) processors are widely used in computer systems. These graphical processes are aimed at the solution of such problems as the transformation of vector representation form into raster one, polygon painting, affined transforms, multiplication, etc. The designing of such devices causes the problem of choice of control algorithm possessing the effective hardware and software [1].

In this connection the problems are suggested to be solved in Volder [2] and Meggitt [3] algorithm systems, which can be successfully used for solving trajectory tasks, signal processing, calculations of elementary functions raster-vector scanning, interception of flat polygons and also for carrying out generalized geometric transformations and generation of flat curves. For these purpose it is well enough to have set of such operations: "multiplying-dividing", "vectoring", "rotation".

The technique of plotting of certain algorithms is fully enough represented in the above-listed works. Volder and Meggitt' s algorithms are known as a mode to possess iteration character. In general the parallelity of calculations may be only achieved at the level of separate formula constituting one iteration procedure[4]. However, in a certain case, it's possible to construct calculating processes with the clipper level of disparallelity, for instance, for the operations "vectoring" and "rotation".

2. Bit-paralleling forms

We shall therefore limit ourselves to consideration of some features resulting from the chosen system of base algorithms. Let us consider, for example, recurrent Volder relationships for the operation "rotation"[2]:

$$\varphi_{i+1} = \varphi_i - \varepsilon_i \cdot \mathrm{arctg} 2^{-i}, \quad \varepsilon_{i+1} = \mathrm{sign}\, \varphi_{i+1}, \tag{1}$$

$$X_{i+1} = X_i + \varepsilon_i 2^{-i} \cdot Y_i, \quad Y_{i+1} = Y_i - \varepsilon_i 2^{-i} \cdot X_i, \tag{2}$$

where: $\varphi_0 = \varphi$, $Y_n = Y$, $X_n = X$, $\varepsilon_0 = +1$, $y_0 = KY'$, $x_0 = KX'$.

[Y,X],[Y',X'] - are correspondingly the old and the new coordinates of the vector end rotated through the angle φ; n- is the number of iterations determined by the maximal rating of number representation (being fixed); ε_i, i=0,..,n-1 are Volder's operators, K- is the coefficient of linear distortion, ($K \approx 1.64$).

The values of all operators belong to set {-1,+1}, are calculated in advance and stored in ROM (table 1).

Table 1.

φ	ε_0	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ε_7	ε_8	ε_9
10	1	-1	-1	1	-1	1	1	-1	-1	1
20	1	-1	1	-1	-1	-1	1	-1	-1	-1
30	1	-1	1	-1	1	1	-1	1	-1	-1
40	1	-1	1	1	1	-1	-1	-1	-1	1
50	1	1	-1	-1	-1	1	1	1	1	-1
60	1	1	-1	1	-1	-1	1	-1	1	1
70	1	1	-1	1	1	1	-1	1	1	1
80	1	1	-1	1	-1	-1	1	1	-1	1
90	1	1	1	1	-1	1	-1	-1	1	1

Parallel realization of "rotation" operation is of some interest. We have the result of disparalleling of initial relationships for "rotation" (n=10) accordingly:

Table2.

$X'=X_{10}$

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
				basic		**part:**			
B	ε_1A	ε_2A	ε_3A	ε_4A	ε_5A	ε_6A	ε_7A	ε_8A	ε_9A
			$-\varepsilon_1\varepsilon_2$B	$-\varepsilon_1\varepsilon_3$B	$-\varepsilon_1\varepsilon_4$B	$-\varepsilon_1\varepsilon_5$B	$-\varepsilon_1\varepsilon_6$B	$-\varepsilon_1\varepsilon_7$B	$-\varepsilon_1\varepsilon_8$B
					$-\varepsilon_2\varepsilon_3$B	$-\varepsilon_2\varepsilon_4$B	$-\varepsilon_2\varepsilon_5$B	$-\varepsilon_2\varepsilon_6$B	$-\varepsilon_2\varepsilon_7$B
							$-\varepsilon_3\varepsilon_4$B	$-\varepsilon_3\varepsilon_5$B	$-\varepsilon_3\varepsilon_6$B
									$-\varepsilon_4\varepsilon_5$B
						$-\varepsilon_1\varepsilon_2\varepsilon_3$A	$-\varepsilon_1\varepsilon_2\varepsilon_4$A	$-\varepsilon_1\varepsilon_2\varepsilon_5$A	$-\varepsilon_1\varepsilon_2\varepsilon_6$A
								$-\varepsilon_1\varepsilon_3\varepsilon_4$A	$-\varepsilon_1\varepsilon_3\varepsilon_5$A
									$-\varepsilon_2\varepsilon_3\varepsilon_4$A
				compen	**Sation**	**part:**			
			B	ε_1A	ε_2A	ε_3A	ε_4A	ε_5A	ε_6A
						$-\varepsilon_1\varepsilon_2$B	$-\varepsilon_1\varepsilon_3$B	$-\varepsilon_1\varepsilon_4$B	$-\varepsilon_1\varepsilon_5$B
								$-\varepsilon_2\varepsilon_3$B	$-\varepsilon_2\varepsilon_4$B
									$-\varepsilon_1\varepsilon_2\varepsilon_3$A
				B	ε_1A	ε_2A	ε_3A	ε_4A	ε_5A
							$-\varepsilon_1\varepsilon_2$B	$-\varepsilon_1\varepsilon_3$B	$-\varepsilon_1\varepsilon_4$B
									$-\varepsilon_2\varepsilon_3$B
						B	ε_1A	ε_2A	ε_3A
									$-\varepsilon_1\varepsilon_2$B
							B	ε_1A	ε_2A
									B

Table3.

$Y'=Y_{10}$

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
				basic		**part:**			
A	$-\varepsilon_1$B	$-\varepsilon_2$B	$-\varepsilon_3$B	$-\varepsilon_4$B	$-\varepsilon_5$B	$-\varepsilon_6$B	$-\varepsilon_7$B	$-\varepsilon_8$B	$-\varepsilon_9$B
			$-\varepsilon_1\varepsilon_2$A	$-\varepsilon_1\varepsilon_3$A	$-\varepsilon_1\varepsilon_4$A	$-\varepsilon_1\varepsilon_5$A	$-\varepsilon_1\varepsilon_6$A	$-\varepsilon_1\varepsilon_7$A	$-\varepsilon_1\varepsilon_8$A
					$-\varepsilon_2\varepsilon_3$A	$-\varepsilon_2\varepsilon_4$A	$-\varepsilon_2\varepsilon_5$A	$-\varepsilon_2\varepsilon_6$A	$-\varepsilon_2\varepsilon_7$A
							$-\varepsilon_3\varepsilon_4$A	$-\varepsilon_3\varepsilon_5$A	$-\varepsilon_3\varepsilon_6$A
									$-\varepsilon_4\varepsilon_5$A
						$\varepsilon_1\varepsilon_2\varepsilon_3$B	$\varepsilon_1\varepsilon_2\varepsilon_4$B	$\varepsilon_1\varepsilon_2\varepsilon_5$B	$\varepsilon_1\varepsilon_2\varepsilon_6$B
								$\varepsilon_1\varepsilon_3\varepsilon_4$B	$\varepsilon_1\varepsilon_3\varepsilon_5$B
									$\varepsilon_2\varepsilon_3\varepsilon_4$B
				compen	**sation**	**part:**			
			A	$-\varepsilon_1$B	$-\varepsilon_2$B	$-\varepsilon_3$B	$-\varepsilon_4$A	$-\varepsilon_5$B	$-\varepsilon_6$B
						$-\varepsilon_1\varepsilon_2$A	$-\varepsilon_1\varepsilon_3$A	$-\varepsilon_1\varepsilon_4$A	$-\varepsilon_1\varepsilon_5$A
								$-\varepsilon_2\varepsilon_3$A	$-\varepsilon_2\varepsilon_4$A
									$\varepsilon_1\varepsilon_2\varepsilon_3$B
				A	$-\varepsilon_1$B	$-\varepsilon_2$B	$-\varepsilon_3$B	$-\varepsilon_4$B	$-\varepsilon_5$B
							$-\varepsilon_1\varepsilon_2$A	$-\varepsilon_1\varepsilon_3$A	$-\varepsilon_1\varepsilon_4$A
									$-\varepsilon_2\varepsilon_3$A
						A	$-\varepsilon_1$B	$-\varepsilon_2$B	$-\varepsilon_3$B
									$-\varepsilon_1\varepsilon_2$A
							A	$-\varepsilon_1$B	$-\varepsilon_2$B
									A

where $A=X+\varepsilon_0 Y$, $B=Y-\varepsilon_0 X$.

There are bit-paralleling forms for binary representation, where $A=a_1a_2...a_8$, $B=b_1b_2...b_8$, $Y'=y_1y_2...y_8$, $X'=x_1x_2...x_8$ are made from Tables 2,3 and can be written as:

$$x_1=b_1, \qquad x_2=b_2+\varepsilon_1 a_1, \qquad (3)$$

$x_3=b_3+\varepsilon_1a_2+\varepsilon_2a_1$,
$x_4=b_4+\varepsilon_1a_3+\varepsilon_2a_2+(\varepsilon_3a_1-\varepsilon_1\varepsilon_2b_1)$,
$x_5=b_5+\varepsilon_1a_4+\varepsilon_2a_3+(\varepsilon_3a_2-\varepsilon_1\varepsilon_2b_2)+(\varepsilon_4a_1-\varepsilon_1\varepsilon_3b_1)$,
$x_6=b_6+\varepsilon_1a_5+\varepsilon_2a_4+(\varepsilon_3a_3-\varepsilon_1\varepsilon_2b_3)+(\varepsilon_4a_2-\varepsilon_1\varepsilon_3b_2)+((\varepsilon_5a_1-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)b_1)$,
$x_7=b_7+\varepsilon_1a_6+\varepsilon_2a_5+(\varepsilon_3a_4-\varepsilon_1\varepsilon_2b_4)+(\varepsilon_4a_3-\varepsilon_1\varepsilon_3b_3)+(\varepsilon_5a_2-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)b_2)+$
$+((\varepsilon_6-\varepsilon_1\varepsilon_2\varepsilon_3)a_1-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)b_1)$,
$x_8=b_8+\varepsilon_1a_7+\varepsilon_2a_6+(\varepsilon_3a_5-\varepsilon_1\varepsilon_2b_5)+(\varepsilon_4a_4-\varepsilon_1\varepsilon_3b_4)+(\varepsilon_5a_3-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)b_3)+$
$+((\varepsilon_6-\varepsilon_1\varepsilon_2\varepsilon_3)a_2-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)b_2)+((\varepsilon_7-\varepsilon_1\varepsilon_2\varepsilon_4)a_1-(\varepsilon_1\varepsilon_6+\varepsilon_2\varepsilon_5+\varepsilon_3\varepsilon_4)b_1)$.

$y_1=a_1$,
(4)
$y_2=a_2-\varepsilon_1b_1$,
$y_3=a_3-\varepsilon_1b_2-\varepsilon_2b_1$,
$y_4=a_4-\varepsilon_1b_3-\varepsilon_2b_2-(\varepsilon_3b_1+\varepsilon_1\varepsilon_2a_1)$,
$y_5=a_5-\varepsilon_1b_4-\varepsilon_2b_3-(\varepsilon_3b_2+\varepsilon_1\varepsilon_2a_2)-(\varepsilon_4b_1+\varepsilon_1\varepsilon_3a_1)$,
$y_6=a_6-\varepsilon_1b_5-\varepsilon_2b_4-(\varepsilon_3b_3+\varepsilon_1\varepsilon_2a_3)-(\varepsilon_4b_2+\varepsilon_1\varepsilon_3a_2)-(\varepsilon_5b_1+(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_1)$,
$y_7=a_7-\varepsilon_1b_6-\varepsilon_2b_5-(\varepsilon_3b_4+\varepsilon_1\varepsilon_2a_4)-(\varepsilon_4b_3+\varepsilon_1\varepsilon_3a_3)-(\varepsilon_5b_2+(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_2)+$
$+((\varepsilon_1\varepsilon_2\varepsilon_3-\varepsilon_6)b_1-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_1)$,
$y_8=a_8-\varepsilon_1b_7-\varepsilon_2b_6-(\varepsilon_3b_5+\varepsilon_1\varepsilon_2a_5)-(\varepsilon_4b_4+\varepsilon_1\varepsilon_3a_4)-(\varepsilon_5b_3+(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_3)+$
$+((\varepsilon_1\varepsilon_2\varepsilon_3-\varepsilon_6)b_2-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_2)+((\varepsilon_1\varepsilon_2\varepsilon_4-\varepsilon_7)b_1-(\varepsilon_1\varepsilon_6+\varepsilon_2\varepsilon_5+\varepsilon_3\varepsilon_4)a_1)$.

The value of each element ε_i (i=0,..,7) for "vectoring" can be generating, for example from equations (4), when all $y_i=0$ (vector is on OX axis). On condition that $a_1=0$, $b_1=1$:

$\varepsilon_0=0$,
(5)
$\varepsilon_1=a_2$,
$\varepsilon_2=a_3-\varepsilon_1b_2$,
$\varepsilon_3=a_4-\varepsilon_1b_3-\varepsilon_2b_2-\varepsilon_1\varepsilon_2a_1$,
$\varepsilon_4=a_5-\varepsilon_1b_4-\varepsilon_2b_3-\varepsilon_3b_2-\varepsilon_1\varepsilon_2a_2-\varepsilon_1\varepsilon_3a_1$,
$\varepsilon_5=a_6-\varepsilon_1b_5-\varepsilon_2b_4-\varepsilon_3b_3-\varepsilon_1\varepsilon_2a_3-\varepsilon_4b_2-\varepsilon_1\varepsilon_3a_2-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_1$,
$\varepsilon_6=a_7-\varepsilon_1b_6-\varepsilon_2b_5-\varepsilon_3b_4-\varepsilon_1\varepsilon_2a_4-\varepsilon_4b_3-\varepsilon_1\varepsilon_3a_3-\varepsilon_5b_2-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_2+\varepsilon_1\varepsilon_2\varepsilon_3b_1-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_1$,
$\varepsilon_7=a_8-\varepsilon_1b_7-\varepsilon_2b_6-\varepsilon_3b_5-\varepsilon_1\varepsilon_2a_5-\varepsilon_4b_4-\varepsilon_1\varepsilon_3a_4-\varepsilon_5b_3-(\varepsilon_1\varepsilon_4+\varepsilon_2\varepsilon_3)a_3+(\varepsilon_1\varepsilon_2\varepsilon_3-\varepsilon_6)b_2-(\varepsilon_1\varepsilon_5+\varepsilon_2\varepsilon_4)a_2+\varepsilon_1\varepsilon_2\varepsilon_4b_1-(\varepsilon_1\varepsilon_6+\varepsilon_2\varepsilon_5+\varepsilon_3\varepsilon_4)a_1$.

Thus, the set of operators $\{\varepsilon_0,..,\varepsilon_7\}$ can be computed in parallel. Parallel calculation of the expressions (3) obtained is perform by means of algorithms and devices of group summation.

3. Compensation methods

The other feature is connected with obtaining effective compensation methods of linear distortions which are the attributes of Volder and Meggitt operations of "rotation" and "vector". For the operation of "rotation" the following method of correction of results is suggested. The idea is to use iteration procedure for each transformed coordinate separately and to introduce the correcting angles. It is assumed here that the initial angle of vector position φ' associated with the position of the point [Y,X] is additionally known. Using Volder procedure "rotation", for example, we turn vector to the angles $\varphi_1=\varphi+d\varphi_1$ and to $\varphi_2=\varphi+d\varphi_2$, where the correcting angles $d\varphi_1$, $d\varphi_2$ are found according to the:

$d\varphi_1=\arccos(\cos(\varphi+\varphi')/K)-(\varphi+\varphi')$,
$d\varphi_2=\arcsin(\sin(\varphi+\varphi')/K)+(\varphi+\varphi')$.

The demand for initial determination of angle of vector position undoubtedly decreases the practical meaning of distortion compensation, however in some cases the method appears quite helpful, for instance, in organizing smooth rotation of a graphical object. Other methods are using multiply table 3,4 elements on coefficient 1/K (compensation part), or duplication of some iterations until K=2 (Table 4).

Table 4.

$\mathbf{P_i}$	$p_2=2$ $p_4=0$ $p_6=0$	$p_2=4$ $p_4=0$ $p_6=0$	$p_2=6$ $p_4=0$ $p_6=0$	$p_2=6$ $p_4=2$ $p_6=0$	$p_2=6$ $p_4=4$ $p_6=0$	$p_2=6$ $p_4=6$ $p_6=0$	$p_2=6$ $p_4=6$ $p_6=2$	$p_2=6$ $p_4=6$ $p_6=4$	$p_2=6$ $p_4=6$ $p_6=6$
K	1.7497	1.8590	1.9752	1.9829	1.9907	1.9985	1.9990	1.9994	1.9999

P_i -is the number of the Volder i-iteration duplication.

In conclusion we can note that above-mentioned methods of realization of bit-paralleling geometrical algorithms on the basis of Volder and Meggitt procedures can find application in geometrical processors for computer graphics systems.

References:

1. Khachumov V.M. Geometrical processor with reduced instruction set computing. //Proc. of the 5-th International Conference of Computer Graphics and Visualization (GRAPHICON'95).-St.Peterburg: GRAFO, 1995, vol.2, p.43.-44.

2.Volder J.E. The CORDIC trigonometric computing technique. //Computer design Development.-1976.- N.-Y.- P.301-307.

3.Meggitt J.E. Digit by digit methods for polynomials.- IBM J. Res.a. Develop., 1963, vol.7, N 3, p.237-245.

4.Naseem A. Fisher D.P. The modified cordic algorithm // Proceeding IEEE Intern. Conference on computer design; VLSI in computers, 1985.-P.144-152.