# Accurate Computation of Geometric Moments Using Non-symmetry and Anti-packing Model for Color Images

Yunping Zheng<sup>1, 2\*</sup>, Yibin Chang<sup>1</sup>, Mudar Sarem<sup>3</sup>

<sup>1</sup> School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong 510006, China.

<sup>2</sup> Medical Image Processing Group, Department of Radiology, University of Pennsylvania, 423 Guardian Drive, Blockley Hall, 4th Floor, Philadelphia, PA 19104, United States.
 <sup>3</sup> School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, China.

\* Corresponding author. Tel.: (+86)020-39380295-3502; email: zhengyp@scut.edu.cn Manuscript submitted April 26, 2016; accepted July 28, 2016. doi: 10.17706/ijcce.2017.6.1.19-28

**Abstract:** Accurate computation of geometric moments is very important in computer vision, image processing and pattern recognition. In this paper, inspired by the idea of bit-plane decomposition and accurate computation of geometric moments on binary images, we put forward an accurate and fast algorithm for the computation of geometric moments using Non-symmetry and Anti-packing Model (NAM) for color images, which takes *O*(*N*) time where *N* is the number of all NAM blocks. By taking four color images 'Lena', 'Peppers', 'Frog', and 'Fish' as typical test objects, and by comparing our proposed NAM-based accurate algorithm with the popular Binary Tree (BT)-based accurate algorithm for computing the geometric moments, the theoretical and experimental results presented in this paper show that our NAM-based accurate algorithm can significantly improve the execution speed by 43.71%, 41.93%, 41.01%, and 38.63% over the BT-based accurate algorithm can significantly improve the average execution speed by 41.32% over the BT-based accurate algorithm. Therefore, in the case of computing lower order moments of color images, our proposed accurate algorithm is much faster than the BT-based accurate algorithm.

**Key words:** Accurate computation, geometric moments, image representation, non-symmetry and anti-packing model (NAM), binary tree (BT), color images.

# 1. Introduction

Image representation is a research spot in computer visualization, image processing and pattern recognition [1]-[4]. An efficient representation method can not only save the storage space of images, but also can reduce the time required for some image operation. In 1980s, Gargantini firstly put forward a linear quadtree representation method [5]. Generally speaking, the linear quadtree can reduce the storage space by 66% and even in some special cases up to 90%. Later, Chen *et al.* proposed a linear binary tree representation method for binary images [6], which showed that the space-efficiency of the BT is superior to that of the linear quadtree. Up to date, the binary tree (BT) is still one of most popular representation methods and it has been widely applied in many fields [7]-[12]. The geometric moments are used in pattern recognition to provide a scale, orientation and position invariant characterization of the shape of a given object [13], [14]. Spiliotis *et al.* [13] proposed a computation formula of geometric moments on binary

images using image block representation. However, due to zero-order approximation and numeric integration over each pixel, the geometric moments calculated are only approximate. Later, Flusser [14] by recalling the original definition of moments in the continuous domain, they put forward an accurate formula to compute the geometric moments. However, this method can not directly deal with gray or color images. In 2005, Chung *et al.* put forward an efficient image algorithm based on the S-tree Coding (STC) method for computing the lower order moments on gray images [15]. Although the representation methods of the hierarchical data structures, such as BT and STC, have many merits and applications, they put too much emphasis upon the symmetry of image partition. Therefore, they are not the optimal representation methods. Inspired by the concept of the packing problem, Chen et al. presented a novel non-symmetry and anti-packing model (NAM) for image representation in order to represent the image pattern more effectively [16]. By using the NAM representation and the extended shading approach, Zheng *et al.* proposed a fast algorithm for computing the lower order moments of gray images [17]. However, the computation results of the geometric moments in [15] and [17] are both approximate since they used the Gouraud shading method and the extended Gouraud shading method, respectively. In fact, accurate computation of the geometric moments on gray or color images are also very important in the field of image processing [18]-[20].

The gray levels of an n-bit gray-scale image can be represented in their binary representation:  $a_{n-1}a_{n-2} \dots a_1a_0$ , where each  $a_j$ ,  $0 \le j \le n-1$ , is 0 or 1. Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the  $a_j$ s into n 1-bit bit planes. In this paper, inspired by the idea of bit-plane decomposition and accurate computation of geometric moments on binary images, we put forward an accurate and fast algorithm for the computation of geometric moments using Non-symmetry and Anti-packing Model for color images, which takes O(N) time where N is the number of all NAM blocks. By taking four color images 'Lena', 'Peppers', 'Frog', and 'Fish' as typical test objects, and by comparing our proposed NAM-based accurate algorithm with the popular BT-based accurate algorithm for computing the geometric moments, the theoretical and experimental results presented in this paper demonstrate the computational advantage of our proposed algorithm.

# 2. Proposed NAM-Based Algorithm for Computing Geometric Moments

In this section, we first simply reintroduce the NAM representation; however, a detailed description can refer to our previous work [16]. Then, we present a new algorithm for computing the geometric moments based on the NAM representation.

# 2.1. Idea of the NAM

The Non-symmetry and Anti-packing Model (NAM) is an anti-packing problem. The idea of the NAM can be described as follows: Given a packed pattern (a packed container) and *n* predefined subpatterns (*n* predefined objects) with different shapes, pick up these subpatterns (objects) from the packed pattern (the packed container) and then represent the packed pattern (the packed container) with a combination of these subpatterns (objects).

The concept of non-symmetry in this paper means that the anti-packing structure is not symmetrical. Non-symmetry, which is related to the symmetry of hierarchical structures, is used because non-symmetry has the capability of representing a packed pattern with the least number of subpatterns in the packing problem. Therefore, NAM representation has the capability of achieving the best pattern representation efficiency which cannot be achieved by the traditional hierarchical representation methods.

## 2.2. Description of the NAM for Color Images

Suppose an original color image pattern is represented by  $\Gamma$  and two reconstructed non-distortion and

distortion patterns of  $\Gamma$  are  $\Gamma'$  and  $\Gamma''$ . Then, the NAM is either a non-distortion transform model from  $\Gamma$  to  $\Gamma'$  or a distortion transform from  $\Gamma$  to  $\Gamma''$ . The transform procedure can be written as:

$$\Gamma' = T(\Gamma), \Gamma'' \approx T(\Gamma),$$

where *T*() is a transform or coding function.

The non-distortion coding procedure can be given by

$$\Gamma' = \bigcup_{j=1}^{n} p_j(v, A | A = \{a_1, a_2, \cdots, a_{m_i}\}) + \varepsilon(d),$$

where  $\Gamma'$  is the reconstructed pattern,  $P = \{p_1, p_2, \dots, p_n\}$  is a set of some predefined subpatterns, n is the number of subpattern types,  $p_j$  is the j<sup>th</sup> subpattern  $(1 \le j \le n)$ ; v is the value of  $p_j$ , A is a parameter set of the subpattern  $p_j$ ,  $a_i(1 \le i \le m_i)$  is a parameter set of shapes for  $p_j$ , m is the number of the subpattern  $p_j$ ,  $\varepsilon(d)$  is a residue pattern, and d is a threshold of  $\varepsilon(d)$ .

If the residue pattern,  $\varepsilon(d)$ , is removed from the non-distortion pattern, then the distortion pattern can be obtained as:

$$\Gamma'' = \bigcup_{j=1}^{n} p_j(v, A | A = \{a_1, a_2, \cdots, a_{m_i}\}).$$

Then the following expression is true.

$$\Gamma \propto \Gamma' = \Gamma'' + \varepsilon(d).$$

#### 2.3. Our NAM-Based Algorithm for Computing Geometric Moments of Color Images

In our NAM representation of color images, the predefined subpatterns are the rectangles,  $p = \{\text{rectangle} \mid \text{rectangle} = L \times W\}$  with two parameters, the length *L* and the width *W*. Since the background level is zero, only the pixels with level one are taken into account for the computation of the moments.

Algorithm: NAM representation for color images.

*Input*: A color image *C* of size  $2^n \times 2^n \times 3$  with a bit depth *m*.

**Output**: An encoding queue set Q for C, where  $Q = \{Q_r_0, Q_r_1, ..., Q_r_{3m-1}\}$ . The variable  $Q_r_i$  denotes the queues of the encoded rectangles of the *i*<sup>th</sup> binary bit-plane image, where  $0 \le i \le 3m-1$ .

**Step 1.** Decompose a color image *C* to 3m binary bit-plane images  $BP_i$  ( $0 \le i \le 3m-1$ ) by the method of the binary bit-plane decomposition and let *i*=0.

**Step 2.** Establish an unmarked start point *sp* from the first entrance of the binary image *BP*<sub>i</sub> according to the raster scanning order, and trace the corresponding rectangular subpatterns. Mark the biggest found rectangular subpattern in the *BP*<sub>i</sub>, and store the parameters of the rectangle into a queue  $Q_r_i$ , i.e,  $Q_r_i \leftarrow \{sp, l, w\}$ , where the variables '*sp*', '*l*', and '*w*' are the upper and left coordinates, the length, and the width of the rectangle, respectively.

Step 3. Repeat Step 2 until there is no unmarked rectangular subpattern in the BP<sub>i</sub>.

**Step 4.** Increase the variable *i* by one. If  $i \le 3m-1$ , then go back to Step 2.

**Step 5.** Output the encoding result *Q*, where  $Q = \{Q_r\}$ .

Suppose that a color image of size  $2^n \times 2^n \times 3$  with a bit depth *m* has been decomposed into the NAM

representation where the number of NAM blocks is *N*, and that the upper-left and the lower-right coordinates of the *i*th NAM block  $B_i$  are  $(x_{1i}, y_{1i})$  and  $(x_{2i}, y_{2i})$ , respectively. Therefore, we can redefine our color images as  $f(x_iy) = \{b_i: i=0, 1, ..., N-1\}$ .

Given a binary image with size  $M \times M$ , let f(x, y) denote the gray value of the pixel at the coordinate (x, y) for  $0 \le x$  and  $y \le M-1$ . Spiliotis *et al.* [13] proposed a computation formula of geometric moments on binary images using image block representation.

The (p+q)th order moment in [13] is defined as follows

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} x^p y^q f(x, y) = \sum_{i=0}^{N-1} \sum_{x=x_{ii}}^{x_{2i}} \sum_{y=y_{1i}}^{y_{2i}} x^p y^q = \sum_{i=0}^{N-1} (\sum_{x=x_{1i}}^{x_{2i}} x^p) (\sum_{y=y_{1i}}^{y_{2i}} y^q).$$
(1)

Let  $\xi(x_{1i}, x_{2i}, p) = \sum_{x=x_{1i}}^{x_{2i}} x^p, \xi(y_{1i}, y_{2i}, q) = \sum_{y=y_{1i}}^{y_{2i}} y^q$ , then the summation  $\xi(x_{1i}, x_{2i}, p)$  for a specific p, where

 $0 \le p \le 4$ , can be computed in O(1) time by using the following formula.

$$\xi(x_{1i}, x_{2i}, p) = \frac{(x_{2i}+1)^{p+1} - x_{1i}^{p+1} - (x_{2i} - x_{1i}+1)}{p+1} - \frac{\binom{p+1}{1}\xi(x_{1i}, x_{2i}, 1) + \binom{p+1}{2}\xi(x_{1i}, x_{2i}, 2)}{p+1} - \frac{\binom{p+1}{p-1}\xi(x_{1i}, x_{2i}, p-1)}{p+1}$$
(2)

where  $\forall p \in Z^+$ ,  $\binom{i}{j} = i!/(j!(i-j)!)$ .

Similarly, the summation  $\xi(y_{1i}, y_{2i}, q) = \sum_{y=y_{1i}}^{y_{2i}} y^q$  for a specific q, where  $0 \le q \le 4$ , can also be computed in

O(1) time. Thus, we can know from the Eq. (1) the moments  $(\sum_{x=x_{ii}}^{x_{2i}} x^p)(\sum_{y=y_{1i}}^{y_{2i}} y^q)$ , where  $0 \le p + q \le 3$ , can also be

computed in O(1) time.

Later, Flusser [14] by recalling the original definition of moments in the continuous domain

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy$$
(3)

where f(x, y) is the characteristic function of the block. Therefore, Eq. (1) is only an approximation of (3). An error  $|M_{Pq} - m_{Pq}|$  is introduced due to zero-order approximation and numeric integration of  $x^{p}y^{q}$  over each pixel. Then, Flusser put forward an accurate formula to compute the geometric moments.

In fact, according to their formula, we can easily deduce the following equation, which is used to compute the geometric moments of any NAM block *b*.

$$M_{pq}^{b} = \int_{x_{li}-0.5}^{x_{2i}+0.5} \int_{y_{li}-0.5}^{y_{2i}+0.5} x^{p} y^{q} f(x,y) dx dy = \frac{(x_{2i}+0.5)^{p+1} - (x_{1i}-0.5)^{p+1}}{p+1} \times \frac{(y_{2i}+0.5)^{q+1} - (y_{1i}-0.5)^{q+1}}{q+1}$$
(4)

where  $(x_{1i}, y_{1i})$ ,  $(x_{2i}, y_{2i})$  the coordinates of the upper left and down right corner of the NAM block.

Some quantities of Eq. (4) can be precalculated and stored in a lookup table. Therefore, for a given color image *C* of size  $2^n \times 2^n \times 3$  with a bit depth *m*, the NAM representation can compute the accurate lower order moments in O(N) time where *N* is the number of NAM blocks.

In the case of the BT representation is concerned, because the BT partition is symmetric, the partition method suffers from a great confine. However, since the NAM partition is asymmetrical, the partition method is unrestricted. The purpose of the NAM partition is to construct subpatterns as large as possible and yield the fewest subpatterns number for a packed pattern. Generally speaking, the total number of subpatterns of the NAM, say *N*, is less than the total number of nodes of the BT, say *K*, i.e., N < K. Therefore, it

can be deduced that the NAM-based representation can compute the geometric moments faster than the BT-based representation.

# 3. Experimental Results

An efficient image representation can not only save space but also facilitate the manipulation of the acquired images. In this section, four color images of size  $256 \times 256 \times 3$  with a bit depth *m*=8 (see Fig. 1) are tested as the benchmark to evaluate the performance of moment computation. We implement the BT-based and the NAM-based algorithms for computing the geometric moments of color images, respectively. For the sake of the fairness of comparison, both of the two algorithms use the strategies of the precalculation. All our experiments are performed on a Celeron microprocessor running at 2.4 GHz with 2 GB RAM. The operating system is MS-Windows XP running the Visual Studio 2015 environment. The programming language is C++.



Fig. 1. Four test images.

Tables 1, 2, 3 and 4 list the approximate and accurate geometric moments using the BT and the NAM representation for 'Lena', 'Peppers', 'Frog', and 'Fish', respectively, where the symbol *Gm* denotes geometric moments; the symbols  $\xi_{BT}$  and  $\xi_{NAM}$  represent the approximate values of the corresponding geometric moments using the BT and the NAM representation, respectively; the symbols  $\beta_{BT}$  and  $\beta_{NAM}$  represent the accurate values of the corresponding geometric moments using the BT and the NAM representation, respectively; the symbols  $\beta_{BT}$  and  $\beta_{NAM}$  representation, respectively; the symbol denote  $\varepsilon$  the error between  $\beta_{NAM}$  and  $\xi_{NAM}$  or between  $\beta_{BT}$  and  $\xi_{BT}$ .

From these four tables, it can easily be seen that the values of both  $\xi_{BT}$  and  $\xi_{NAM}$  are always equal. Also, the values of both  $\beta_{BT}$  and  $\beta_{NAM}$  are always equal. Further, the approximate values of the corresponding geometric moments for each image are always less than the accurate values of the corresponding geometric moments except for these four geometric moments, i.e.,  $m_{00}$ ,  $m_{01}$ ,  $m_{10}$ , and  $m_{11}$ , whose approximate values of the corresponding geometric moments are always equal to the accurate values of the corresponding geometric moments due to zero-order approximation and numeric integration of  $x^p y^q$  over each pixel. The parameter  $\varepsilon$  shows the difference between the approximate values and the accurate values.

	11		U		
Gm	$\xi_{BT}$	ξNAM	βвт	Влам	ε
$m_{00}$	25210831	25210831	25210831	25210831	0
$m_{01}$	3127945069	3127945069	3127945069	3127945069	0
$m_{02}$	524666810497	524666810497	524668911400	524668911400	2100903
$m_{03}$	99658682214541	99658682214541	99659464200808	99659464200808	781986267
$m_{10}$	3330676309	3330676309	3330676309	3330676309	0
$m_{11}$	419972389817	419972389817	419972389817	419972389817	0
<i>m</i> <sub>12</sub>	70753766888973	70753766888973	70754044445332	70754044445332	277556359
$m_{20}$	575471331463	575471331463	575473432366	575473432366	2100903
$m_{21}$	73434594060799	73434594060799	73434854722888	73434854722888	260662089
$m_{30}$	110655658557439	110655658557439	110656491226516	110656491226516	832669077

T-11. 1 American to and American	Commenter Manager to Hade		····· (1 ····)
Table 1. Approximate and Accurate	Geometric Moments Usir	ig bi and NAM Re	presentation for Lena

Table 2. Approximate and Accurate Geometric Moments Using BT and NAM Representation for 'Peppers'

Gm	$\xi_{BT}$	$\xi_{NAM}$	$eta_{\scriptscriptstyle BT}$	$\beta_{NAM}$	ε
$m_{00}$	21789686	21789686	21789686	21789686	0
$m_{01}$	2678015499	2678015499	2678015499	2678015499	0
$m_{02}$	450030089863	450030089863	450031905670	450031905670	1815807
$m_{03}$	85996217826225	85996217826225	85996887330100	85996887330100	669503875
$m_{10}$	2803248817	2803248817	2803248817	2803248817	0
$m_{11}$	336218218453	336218218453	336218218453	336218218453	0
<i>m</i> <sub>12</sub>	55345684287279	55345684287279	55345917891347	55345917891347	233604068
$m_{20}$	478895170545	478895170545	478896986352	478896986352	1815807
<i>m</i> <sub>21</sub>	56793505664915	56793505664915	56793728832873	56793728832873	223167958
<i>m</i> <sub>30</sub>	92143088329339	92143088329339	92143789141543	92143789141543	700812204

Table 3. Approximate and Accurate Geometric Moments Using BT and NAM Representation for 'Frog'

Gm	<i>ξBT ξNAM</i>		$eta_{\scriptscriptstyle BT}$	$eta_{\scriptscriptstyle NAM}$	ε	
$m_{00}$	12238742 12238742		12238742	12238742	0	
$m_{01}$	1599312932	1599312932	1599312932	1599312932	0	
$m_{02}$	273228876830	273228876830	273229896725	273229896725	1019895	
$m_{03}$	51967337177774	51967337177774	51967737006007	51967737006007	399828233	
$m_{10}$	1633924130	1633924130	1633924130	1633924130	0	
$m_{11}$	212109416760	212109416760	212109416760	212109416760	0	
$m_{12}$	36049915581240	36049915581240	36050051741584	36050051741584	136160344	
$m_{20}$	282526224846	282526224846	282527244741	282527244741	1019895	
<i>m</i> <sub>21</sub>	36415825290030	36415825290030	36415958566108	36415958566108	133276078	
<i>m</i> <sub>30</sub>	54578463383492	54578463383492	54578871864525	54578871864525	408481033	

Table 4. Approximate and Accurate Geometric Moments Using BT and NAM Representation for 'Fish'

Gm	$\xi_{BT}$	$\xi_{NAM}$	$eta_{\scriptscriptstyle BT}$	$\beta_{NAM}$	ε
$m_{00}$	16712860	16712860	16712860	16712860	0
$m_{01}$	2112465980	2112465980	2112465980	2112465980	0
$m_{02}$	338080731968	338080731968	338082124706	338082124706	1392738
$m_{03}$	60669916241972	60669916241972	60670444358467	60670444358467	528116495
$m_{10}$	2127420606	2127420606	2127420606	2127420606	0
$m_{11}$	267986393348	267986393348	267986393348	267986393348	0
<i>m</i> <sub>12</sub>	43250531589712	43250531589712	43250708874762	43250708874762	177285050
$m_{20}$	339321091538	339321091538	339322484276	339322484276	1392738
$m_{21}$	43108758435632	43108758435632	43108934474464	43108934474464	176038832
<i>m</i> <sub>30</sub>	61822784849328	61822784849328	61823316704480	61823316704480	531855152

Complexities of many image operation algorithms are proportion to the node or block number of the images. The less the node or block number of the images is, the faster the image operation algorithm is. Table 5 presents the performance parameters of both the BT and the NAM representation methods of color images where the symbols  $N_{BT}$  and  $N_{NAM}$  represent the node number of the BT and the NAM representation, respectively; the symbol  $\sigma = (N_{BT} - N_{NAM})/N_{BT}$ ; the symbols  $C_{BT}$  and  $C_{NAM}$  represent the compression ratios of the BT and the NAM representation, respectively; the symbol  $\sigma = (N_{BT} - N_{NAM})/N_{BT}$ ; the symbol  $\gamma = (C_{NAM} - C_{BT})/C_{BT}$ ; the symbols  $\varphi_{BT}$  and  $\varphi_{NAM}$  represent the approximate time of the corresponding geometric moments using the BT and the NAM represent the accurate time of the corresponding geometric moments using the NAM representation, respectively; the symbol  $\omega = (\varphi_{NAM} - \varphi_{BT})/\varphi_{BT}$ ; the symbols  $\varphi_{BT}$  and the NAM representation, respectively; the symbol  $\omega = (\varphi_{NAM} - \varphi_{BT})/\varphi_{BT}$ ; the symbols  $\varphi_{BT}$  and the NAM represent the accurate time of the corresponding geometric moments using the BT and the NAM representation, respectively; the symbol  $\omega = (\varphi_{BT} - \varphi_{NAM})/\varphi_{BT}$ . The time unit for the moment computation is millisecond.

It can be seen from Table 1 that the block numbers of Figs. 1(a)–(d) using the BT (NAM) are 377631 (205680), 365616 (204419), 262903 (151412), and 357236 (212866) respectively. In the case of the given four standard test images, from Table 1 we can easily notice that the block numbers of the NAM are always less than those of the BT. In fact, further comparison of the number of blocks allows us to work out that the

NAM can significantly reduce the block number by 45.53%, 44.09%, 42.41%, and 40.41% over BT in images 'Lena', 'Peppers', 'Frog', and 'Fish', respectively. In other words, the average block numbers of BT and NAM for the four images are 340847 and 193594, respectively. Therefore, the NAM can reduce the block numbers by 43.11% than the BT in average. Therefore, our proposed representation method is more effective than the BT representation method with respect to the node number of the images.

Table 5 presents the compression ratios of the BT and the NAM representation methods. The compression ratios of 'Lena', 'Peppers', 'Frog', and 'Fish' using the BT (NAM) are 0.1485 (0.1772), 0.1555 (0.1871), 0.2028 (0.2402), and 0.1679 (0.2071) respectively. In the case of the compression ratios of both the BT and the NAM representation methods, we can see from Table 5 that our NAM representation method can save the storage room by 19.33%, 20.32%, 18.44%, and 23.35% over the BT representation method in images 'Lena', 'Peppers', 'Frog', and 'Fish', respectively. From Table 5, we can also see that the average compress ratios of BT and NAM for the four images are 0.1687 and 0.2029, respectively. That's is to say, in the case of the four test images are concerned, our proposed NAM over the BT has about 20.36% memory-saving improvement ratio in average. Therefore, our proposed representation method is also more effective than the BT representation method with respect to the storage room.

Table 5. Comparison of Number of Blocks or Pixels between BT and NAM Representation Methods

Image	$N_{BT}$	<b>N</b> NAM	$\sigma$	$C_{BT}$	CNAM	γ	$arphi_{BT}$	arphiNAM	$\sigma$	$\phi_{\scriptscriptstyle BT}$	$\phi$ NAM	$\theta$
Lena	377631	205680	45.53%	0.1485	0.1772	19.33%	923.44	503.91	45.43%	58.29	32.81	43.71%
Peppers	365616	204419	44.09%	0.1555	0.1871	20.32%	936.25	526.25	43.79%	58.91	34.21	41.93%
Frog	262903	151412	42.41%	0.2028	0.2402	18.44%	650.31	370.00	43.10%	42.65	25.16	41.01%
Fish	357236	212866	40.41%	0.1679	0.2071	23.35%	890.46	544.22	38.88%	57.50	35.29	38.63%
Avg	340847	193594	43.11%	0.1687	0.2029	20.36%	850.12	486.10	42.80%	54.34	31.87	41.32%

As far as the approximate computation method is concerned, from Table 5, we can know that  $\varphi_{NAM}$  is always less than  $\varphi_{BT}$  for each image. In addition, from the values of  $\varpi$ , it can be seen that our NAM-based approximate algorithm can significantly improve the execution speed by 45.43%, 43.79%, 43.10%, and 38.88% over the BT-based approximate algorithm in images 'Lena', 'Peppers', 'Frog', and 'Fish', respectively. Also, it can be seen that the average execution time of the BT-based and the NAM-based approximate algorithms for the four images are 850.12 ms, and 486.10 ms, respectively. In other words, our NAM-based approximate algorithm can significantly improve the average execution speed by 42.80% over the BT-based approximate algorithm.

Similarly, as far as the accurate computation method is concerned, from Table 5, we can know that  $\phi_{NAM}$  is always less than  $\phi_{BT}$  for each image. In addition, from the values of  $\theta$ , it can be seen that our NAM-based accurate algorithm can significantly improve the execution speed by 43.71%, 41.93%, 41.01%, and 38.63% over the BT-based accurate algorithm in images 'Lena', 'Peppers', 'Frog', and 'Fish', respectively. Also, it can be seen that the average execution time of the BT-based and the NAM-based accurate algorithms for the four images are 54.34 ms, and 31.87 ms, respectively. In other words, our NAM-based accurate algorithm can significantly improve the average execution speed by 41.32% over the BT-based accurate algorithm.

Further, as far as both the approximate and the accurate computation methods are concerned, we can work out from Table 5 that the average execution time of the BT-based approximate algorithm is 15.6445 times of the BT-based accurate algorithm, which shows the BT-based accurate algorithm can significantly improve the execution speed by 93.61% than the BT-based approximate algorithm. Similarly, we can work out from Table 5 that the average execution time of the NAM-based approximate algorithm is 15.2526 times of the NAM-based accurate algorithm, which shows the NAM-based accurate algorithm can significantly improve the execution speed by 93.44% than the NAM-based approximate algorithm. Therefore,

precalculation can significant improve the execution speed of geometric moments whether for the BT-based accurate algorithm or the NAM-based accurate algorithm.

As stated above, the experimental results in this section show that our proposed NAM-based algorithm for accurate computation of geometric moments for color images is much faster than the previous BT-based algorithm for accurate computation of geometric moments for color images.

# 4. Conclusions

In this paper, inspired by the idea of bit-plane decomposition and accurate computation of geometric moments on binary images, we put forward an accurate and fast algorithm for the computation of geometric moments using Non-symmetry and Anti-packing Model for color images, which takes O(N) time where *N* is the number of all NAM blocks. By taking four color images 'Lena', 'Peppers', 'Frog', and 'Fish' as typical test objects, and by comparing our proposed NAM-based accurate algorithm with the popular BT-based accurate algorithm for computing the geometric moments, the theoretical and experimental results presented in this paper demonstrate the computational advantage of our proposed algorithm.

## Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 61300134, the Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20120172120036, the Natural Science Foundation of Guangdong Province of China under Grant No. S2011040005815, No. S2013010012515, and No. 2015A030313206, the Fundamental Research Funds for the Central Universities of China under Grant No. 2015ZM133, and Chinese National Scholarship Fund under Grant No. 201406155015.

## References

- [1] Wang, J., Jiang, H., Jia, Y., Hua, X. S., Zhang, C., & Quan, L. (2014). Regularized tree partitioning and its application to unsupervised image segmentation. *IEEE Transactions on Image Processing, 23(4)*, 1909-1922.
- [2] Zheng, Y., Yu, Z., You, J., & Sarem, M. (2012). A novel gray image representation using overlapping rectangular NAM and extended shading approach. *Journal of Visual Communication & Image Representation*, 23(7), 972-983.
- [3] Yang, X., Gao, X., Tao, D., Li, X., & Li, J. (2014). An efficient mrf embedded level set method for image segmentation. *IEEE Transactions on Image Processing*, *24(1)*, 9-21.
- [4] Zheng, Y., & Sarem, M. (2014). A novel binary image representation algorithm by using NAM and coordinate encoding procedure and its application to area calculation. *Frontiers of Computer Science*, 8(5), 763-772.
- [5] Gargantini, I. (1982). An effective way to represent quadtrees. *Communications of the ACM, 25(12),* 905-910.
- [6] Chen, C., & Zou, H. (1988). Linear binary tree. *Proceedings of the 9th International Conference on Pattern Recognition* (pp. 576 – 578). Rome.
- [7] David, T., Kempen, T. V., Huang, H., & Wilson, P. (2011). The geometry and dynamics of binary trees. *Mathematics & Computers in Simulation, 81(7),* 1464-1481.
- [8] Chen, T. W., Su, Y. C., Huang, K. Y., Tsai, Y. M., Chien, S. Y., & Chen, L. G. (2012). Visual vocabulary processor based on binary tree architecture for real-time object recognition in full-HD resolution. *IEEE Transactions on Very Large Scale Integration Systems*, 20(12), 2329-2332.
- [9] Veganzones, M. A., Tochon, G., Dallamura, M., Plaza, A. J., & Chanussot, J. (2014). Hyperspectral image

segmentation using a new spectral unmixing-based binary partition tree representation. *IEEE Transactions on Image Processing, 23(8),* 3574-3589.

- [10] Liu, Z., Shen, L., & Zhang, Z. (2011). Unsupervised image segmentation based on analysis of binary partition tree for salient object extraction. *Signal Processing*, *91(2)*, 290-299.
- [11] Alonso-Gonzalez, A., Lopez-Martinez, C., & Salembier, P. (2012). Filtering and segmentation of polarimetric sar data based on binary partition trees. *IEEE Transactions on Geoscience & Remote Sensing*, *50*(2), 593-605.
- [12] Huang, K. K., & Dai, D. Q. (2012). A new on-board image codec based on binary tree with adaptive scanning order in scan-based mode. *IEEE Transactions on Geoscience & Remote Sensing*, 50(10), 3737-3750.
- [13] Spiliotis, I. M., & Mertzios, B. G. (1998). Real-time computation of two-dimensional moments on binary images using image block representation. *IEEE Transactions on Image Processing*, *7(11)*, 1609-1615.
- [14] Flusser, J. (2000). Refined moment calculation using image block representation. *IEEE Transactions on Image Processing*, *9*(*11*), 1977-1978.
- [15] Chung, K. L., & Chen, P. C. (2005). An efficient algorithm for computing moments on a block representation of a grey-scale image. *Pattern Recognition*, *38(12)*, 2578-2586.
- [16] Chen, C., Zheng, Y., & Sarem, M. (2009). A novel non-symmetry and anti-packing model for image representation. *Chinese Journal of Electronics*, *18*(1), 89-94.
- [17] Zheng, Y., & Sarem, M. (2011). A fast algorithm for computing moments of gray images based on nam and extended shading approach. *Frontiers of Computer Science in China*, *5*(1), 57-65.
- [18] Shebaita, A., Das, D., Petranovic, D., & Ismail, Y. (2011). A novel moment based framework for accurate and efficient static timing analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *30(8)*, 1258-1262.
- [19] Kotoulas, L., & Andreadis, I. (2007). Accurate calculation of image moments. *IEEE Transactions on Image Processing*, *16(8)*, 2028-2037.
- [20] Karakasis, E. G., Papakostas, G. A., Koulouriotis, D. E., & Tourassis, V. D. (2013). A unified methodology for computing accurate quaternion color moments and moment invariants. *IEEE Transactions on Image Processing*, 23(2), 596-611.



**Yunping Zheng** was born in Hubei, China, in 1979. He received his B.S. degree in computer science from Airforce Radar Academy, Wuhan, Hubei, China, in 2001, M.S. degree and Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan, Hubei, China in 2005 and 2008, respectively.

He is currently an associate professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He has published over 70 papers in referred journals and conferences.

Dr. Zheng is a member of ACM. His interests include computer graphics, fractal image compression, image processing, and pattern recognition.



**Yibin Chang** was born in Shandong Province, China, in 1989. He received the B.E. degree from Shandong University of Science and Technology, Qingdao, in 2010, in computer science and technology.

He is pursuing the master degree with the Department of Computer Technology in South China University of Technology.

His research interests include image processing and computer vision.



**Mudar Sarem** received his BS in electronic engineering from Tishreen University, Lattakia, Syria, in 1989, and his MS and PhD degrees in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1997 and 2002, respectively.

He is currently an Associate Professor with the School of Software Engineering, Huazhong University of Science and Technology, Wuhan, China. He has published over 50 papers in referred conferences and journals.

Prof. Sarem's research interests include computer graphics, multimedia databases, and image processing.