Reduction of Numerical Errors in Zernike Invariants Computed via Complex-Valued Integral Images

Przemysław Klęsk(✉), Aneta Bera, and Dariusz Sychel

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, ul. Żołnierska 49, 71-210 Szczecin, Poland
{pklesk,abera,dsychel}@wi.zut.edu.pl

Abstract. Floating-point arithmetics may lead to numerical errors when numbers involved in an algorithm vary strongly in their orders of magnitude. In the paper we study numerical stability of Zernike invariants computed via complex-valued integral images according to a constant-time technique from [2], suitable for object detection procedures. We indicate numerically fragile places in these computations and identify their cause, namely—binomial expansions. To reduce numerical errors we propose *piecewise integral images* and derive a numerically safer formula for Zernike moments. Apart from algorithmic details, we provide two object detection experiments. They confirm that the proposed approach improves accuracy of detectors based on Zernike invariants.

Keywords: Zernike moments · Complex-valued integral images · Numerical errors reduction · Object detection

1 Introduction

The classical approach to object detection is based on sliding window scans. It is computationally expensive, involves a large number of image fragments (windows) to be analyzed, and in practice precludes the applicability of advanced methods for feature extraction. In particular, many *moment* functions [9], commonly applied in image recognition tasks, are often preculded from detection, as they involve inner products i.e. linear-time computations with respect to the number of pixels. Also, the deep learning approaches cannot be applied directly in detection, and require preliminary stages of prescreening or region-proposal.

There exist a few feature spaces (or descriptors) that have managed to bypass the mentioned difficulties owing to constant-time techniques discovered for them within the last two decades. Haar-like features (HFs), local binary patterns (LBPs) and HOG descriptor are state-of-the-art examples from this category [1,4,14] The crucial algorithmic trick that underlies these methods and allows

This work was financed by the National Science Centre, Poland. Research project no.: 2016/21/B/ST6/01495.

for constant-time—$O(1)$—feature extraction are *integral images*. They are auxiliary arrays storing cumulative pixel intensities or other pixel-related expressions. Having prepared them before the actual scan, one is able to compute fast the wanted sums via so-called 'growth' operations. Each growth involves two additions and one subtraction using four entries of an integral image.

In our research we try to broaden the existing repertoire of constant-time techniques for feature extraction. In particular, we have managed to construct such new techniques for Fourier moments (FMs) [7] and Zernike moments (ZMs) [2]. In both cases a *set* of integral images is needed. For FMs, the integral images cumulate products of image function and suitable trigonometric terms and have the following general forms: $\sum\sum_{j,k} f(j,k)\cos(-2\pi(jt/\text{const}_1 + ku/\text{const}_2))$, and $\sum\sum_{j,k} f(j,k)\sin(-2\pi(jt/\text{const}_1 + ku/\text{const}_2))$, where f denotes the image function and t, u are order-related parameters. With such integral images prepared, each FM requires only 21 elementary operations (including 2 growths) to be extracted during a detection procedure. In the case of ZMs, complex-valued integral images need to be prepared, having the form: $\sum\sum_{j,k} f(j,k)(k-\boldsymbol{i}j)^t(k+\boldsymbol{i}j)^u$, where $\boldsymbol{i}$ stands for the imaginary unit ($\boldsymbol{i}^2 = -1$). The formula to extract a single ZM of order (p,q) is more intricate and requires roughly $\frac{1}{24}p^3 - \frac{1}{8}pq^2 + \frac{1}{12}q^3$ growths, but still the calculation time is not proportional to the number of pixels.

It should be remarked that in [2] we have flagged up, but not tackled, the problem of *numerical errors* that may occur when computations of ZMs are backed with integral images. ZMs are complex numbers, hence the natural data type for them is the `complex` type with real and imaginary parts stored in the double precision of the IEEE-754 standard for floating-point numbers (a precision of approximately 16 decimal digits). The main culprit behind possible numerical errors are *binomial expansions*. As we shall show the algorithm must explicitly expand two binomial expressions to benefit from integral images, which leads to numbers of different magnitudes being involved in the computations. When multiple additions on such numbers are carried out, digits of smaller-magnitude numbers can be lost.

In this paper we address the topic of numerical stability. The key new contribution are **piecewise integral images**. Based on them we derive a **numerically safer formula for** the computation of a single **Zernike moment**. The resulting technique introduces some computational overhead, but remains to be a constant-time technique.

2 Preliminaries

Recent literature confirms that ZMs are still being applied in many image recognition tasks e.g: human age estimation [8], electrical symbols recognition [16], traffic signs recognition [15], tumor diagnostics from magnetic resonance [13]. Yet, it is quite difficult to find examples of detection tasks applying ZMs directly. Below we describe the constant-time approach to extract ZMs within detection, together with the proposition of numerically safe computations.

2.1 Zernike Moments, Polynomials and Notation

ZMs can be defined in both polar and Cartesian coordinates as:

$$M_{p,q} = \frac{p+1}{\pi}\int_0^{2\pi}\int_0^1 f(r,\theta) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{-iq\theta}\, r\, dr\, d\theta, \tag{1}$$

$$= \frac{p+1}{\pi} \iint\limits_{x^2+y^2\leqslant 1} f(x,y) \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s}(x+iy)^{\frac{1}{2}(p-q)-s}(x-iy)^{\frac{1}{2}(p+q)-s}\, dx\, dy, \tag{2}$$

where:

$$\beta_{p,q,s} = \frac{(-1)^s (p-s)!}{s!((p+q)/2-s)!((p-q)/2-s)!}, \tag{3}$$

i is the imaginary unit ($i^2 = -1$), and f is a mathematical or an image function defined over unit disk [2,17]. p and q indexes, representing moment order, must be simultaneously even or odd, and $p \geqslant |q|$.

ZMs are in fact the *coefficients* of an *expansion* of function f, given in terms of Zernike polynomials $V_{p,q}$ as the orthogonal base:[1]

$$f(r,\theta) = \sum_{0\leqslant p\leqslant\infty} \sum_{\substack{-p\leqslant q\leqslant p\\ p-|q| \text{ even}}} M_{p,q} V_{p,q}(r,\theta), \tag{4}$$

where $V_{p,q}(r,\theta) = \sum_{s=0}^{(p-|q|)/2} \beta_{p,q,s} r^{p-2s} e^{iq\theta}$. As one can note $V_{p,q}$ combines a standard polynomial defined over radius r and a harmonic part defined over angle θ. In applications, finite partial sums of expansion (4) are used. Suppose ρ and ϱ denote the imposed maximum orders, polynomial and harmonic, respectively, and $\rho \geqslant \varrho$. Then, the partial sum that approximates f can be written down as:

$$f(r,\theta) \approx \sum_{0\leqslant p\leqslant\rho} \sum_{\substack{-\min\{p,\varrho\}\leqslant q\leqslant\min\{p,\varrho\}\\ p-|q| \text{ even}}} M_{p,q} V_{p,q}(r,\theta). \tag{5}$$

2.2 Invariants Under Rotation

ZMs are invariant to scale transformations, but, as such, are not invariant to rotation. Yet, they do allow to build suitable expressions with that property. Suppose f' denotes a version of function f rotated by an angle α, i.e. $f'(r,\theta) = f(r,\theta+\alpha)$. It is straightforward to check, deriving from (1), that the following identity holds

$$M'_{p,q} = e^{iq\alpha} M_{p,q}, \tag{6}$$

[1] ZMs expressed by (1) arise as inner products of the approximated function and Zernike polynomials: $M_{p,q} = \langle f, V_{p,q}\rangle / \|V_{p,q}\|^2$.

where $M'_{p,q}$ represents a moment for the rotated function f'. Hence in particular, the *moduli* of ZMs are one type of rotational invariants, since

$$|M'_{p,q}| = |e^{iq\alpha} M_{p,q}| = |e^{iq\alpha}||M_{p,q}| = |M_{p,q}|. \tag{7}$$

Apart from the moduli, one can also look at the following *products* of moments

$$M_{p,q}{}^{n} M_{v,s} \tag{8}$$

with the first factor raised to a natural power. After rotation one obtains

$$M'_{p,q}{}^{n}\ M'_{v,s}{=}e^{inq\alpha} M_{p,q}{}^{n}\ e^{is\alpha} M_{v,s}{=}e^{i(nq+s)\alpha} M_{p,q}{}^{n}\ M_{v,s}. \tag{9}$$

Hence, by forcing $nq + s = 0$ one can obtain many rotational invariants because the angle-dependent term $e^{i(nq+s)\alpha}$ disappears.

2.3 ZMs for an Image Fragment

In practical tasks it is more convenient to work with rectangular, rather than circular, image fragments. Singh and Upneja [12] proposed a workaround to this problem: a square of size $w \times w$ in pixels (w is even) becomes *inscribed* in the unit disc, and zeros are "laid" over the square-disc complement. This reduces integration over the disc to integration over the square. The inscription implies that widths of pixels become $\sqrt{2}/w$ and their areas $2/w^2$ (a detail important for integration). By iterating over pixel indexes: $0 \leqslant j, k \leqslant w-1$, one generates the following Cartesian coordinates within the unit disk:

$$x_k = \frac{2k-(w-1)}{w\sqrt{2}}, \qquad y_j = \frac{w-1-2j}{w\sqrt{2}}. \tag{10}$$

In detection, it is usually sufficient to replace integration involved in $M_{p,q}$ by a suitable summation, thereby obtaining a zeroth order approximation. In subsequent sections we use the formula below, which represents such an approximation (hat symbol) and is adjusted to have a convenient indexing for our purposes:

$$\widehat{M}_{2p+o,2q+o}{=}\frac{4p{+}2o{+}2}{\pi w^2}\sum\sum_{0\leqslant j,k\leqslant w-1} f(j,k)\sum_{q\leqslant s\leqslant p}\beta_{2p+o,2q+o,p-s}(x_k{+}iy_j)^{s-q}(x_k{-}iy_j)^{s+q+o} \tag{11}$$

— namely, we have introduced the substitutions $p := 2p + o$, $q := 2q + o$. They play two roles: they make it clear whether a moment is even or odd via the flag $o \in \{0, 1\}$; they allow to construct integral images, since exponents $\frac{1}{2}(p \mp s) - s$ present in (2) are now suitably reduced.

2.4 Proposition from [2]

Suppose a digital image of size $n_x \times n_y$ is traversed by a $w \times w$ sliding window. For clarity we discuss only a single-scale scan. The situation is sketched in Fig. 1. Let

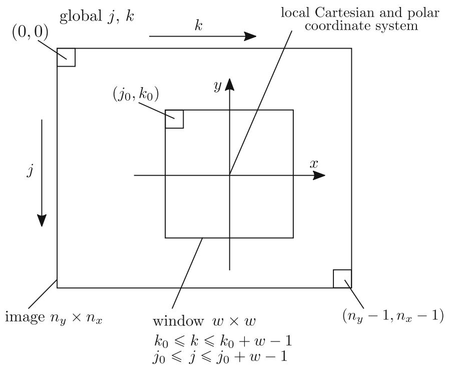
global j, k
$(0,0)$
k
local Cartesian and polar coordinate system
(j_0,k_0)
y
x
j
image $n_y \times n_x$
window $w \times w$
$k_0 \leqslant k \leqslant k_0+w-1$
$j_0 \leqslant j \leqslant j_0+w-1$
(n_y-1, n_x-1)

Fig. 1. Illustration of detection procedure using sliding window.

(j,k) denote global coordinates of a pixel in the image. For each window under analysis, its offset (top-left corner) will be denoted by (j_0,k_0). Thus, indexes of pixels that belong to the window are: $j_0 \leqslant j \leqslant j_0+w-1$, $k_0 \leqslant k \leqslant k_0+w-1$. Alse let (j_c,k_c) represent the *central index* of the window:

$$j_c = \frac{1}{2}(2j_0+w-1), \quad k_c = \frac{1}{2}(2k_0+w-1). \tag{12}$$

Given a global index (j,k) of a pixel, the local Cartesian coordinates corresponding to it (mapped to the unit disk) can be expressed as:

$$x_k = \frac{2(k-k_0)-(w-1)}{w\sqrt{2}} = \frac{\sqrt{2}}{w}(k-k_c), \quad y_j = \frac{(w-1)-2(j-j_0)}{w\sqrt{2}} = \frac{\sqrt{2}}{w}(j_c-j). \tag{13}$$

Let $\{ii_{t,u}\}$ denote a set of **complex-valued integral images**[2]:

$$ii_{t,u}(l,m) = \sum_{\substack{0\leqslant j\leqslant l\\0\leqslant k\leqslant m}}\sum f(j,k)(k-ij)^t(k+ij)^u, \quad {}^{0\leqslant l\leqslant n_y-1}_{0\leqslant m\leqslant n_x-1}; \tag{14}$$

where pairs of indexes (t,u), generating the set, belong to: $\{(t,u)\colon 0\leqslant t\leqslant \lfloor \rho/2\rfloor$, $0\leqslant u\leqslant \min(\rho-t, \lfloor(\rho+\varrho)/2\rfloor)\}$.

For any integral image we also define the **growth operator** over a rectangle spanning from (j_1,k_1) to (j_2,k_2):

$$\underset{\substack{j_1,j_2\\k_1,k_2}}{\Delta}(ii_{t,u}) = ii_{t,u}(j_2,k_2) - ii_{t,u}(j_1-1,k_2) - ii_{t,u}(j_2,k_1-1) + ii_{t,u}(j_1-1,k_1-1) \tag{15}$$

with two complex-valued substractions and one addition. The main result from [2] (see there for proof) is as follows.

[2] In [2] we have proved that integral images $ii_{t,u}$ and $ii_{u,t}$ are complex conjugates at all points, which allows for computational savings.

Proposition 1. *Suppose a set of integral images $\{ii_{t,u}\}$, defined as in (14), has been prepared prior to the detection procedure. Then, for any square window in the image, each of its Zernike moments (11) can be calculated in constant time — $O(1)$, regardless of the number of pixels in the window, as follows:*

$$\widehat{M}_{2p+o,2q+o} = \frac{4p+2o+2}{\pi w^2} \sum_{2q+o \leqslant 2s+o \leqslant 2p+o} \beta_{2p+o,2q+o,p-s} \left(\frac{\sqrt{2}}{w}\right)^{2s+o}$$
$$\cdot \sum_{t=0}^{s-q} \binom{s-q}{t} (-k_c+ij_c)^{s-q-t} \sum_{u=0}^{s+q+o} \binom{s+q+o}{u} (-k_c-ij_c)^{s+q+o-u} \underset{\substack{j_0,j_0+w-1\\k_0,k_0+w-1}}{\Delta}(ii_{t,u}). \quad (16)$$

3 Numerical Errors and Their Reduction

Floating-point additions or subtractions are dangerous operations [6,10] because when numbers of different magnitudes are involved, the right-most digits in the mantissa of the smaller-magnitude number can be lost when widely spaced exponents are aligned to perform an operation. When ZMs are computed according to Proposition 1, such situations can arise in *two* places.

The connection between the definition-style ZM formula (11) and the integral images-based formula (16) are expressions (13): $\frac{\sqrt{2}}{w}(k-k_c)$, $\frac{\sqrt{2}}{w}(j_c-j)$. They map global coordinates to local unit discs. When the mapping formulas are plugged into x_k and y_j in (11), the following subexpression arises under summations:

$$\cdots \left(\sqrt{2}/w\right)^{2s+o} (k-ij - k_c+ij_c)^{s-q} (k+ij - k_c-ij_c)^{s+q+o}. \quad (17)$$

Now, to benefit from integral images one has to explictly expand the two binomial expressions, distinguishing two groups of terms: $k \mp ij$—dependent on the global pixel index, and $-k_c \pm ij_c$—independent of it . By doing so, coordinates of the particular window can be isolated out and formula (16) is reached. Unfortunately, this also creates two numerically fragile places. The first one are integral images themselves, defined by (14). Global pixel indexes j, k present in power terms $(k-ij)^t(k+ij)^u$ vary within: $0 \leqslant j \leqslant n_y - 1$ and $0 \leqslant k \leqslant n_x - 1$. Hence, for an image of size e.g. 640×480, the summands vary in magnitude roughly from $10^{0(t+u)}$ to $10^{3(t+u)}$. To fix an example, suppose $t+u = 10$ (achievable e.g. when $\rho = \varrho = 10$) and assume a roughly constant values image function. Then, the integral image $ii_{t,u}$ has to cumulate values ranging from 10^0 up to 10^{30}. Obviously, the rounding-off errors amplify as the $ii_{t,u}$ sum progresses towards the bottom-right image corner. The second fragile place are expressions: $(-k_c + ij_c)^{s-q-t}$ and $(-k_c - ij_c)^{s+q+o-u}$, involving the central index, see (16). Their products can too become very large in magnitude as computations move towards the bottom-right image corner.

In error reports we shall observe *relative errors*, namely:

$$\mathrm{err}(\phi, \phi^*) = |\phi - \phi^*| \big/ |\phi^*|, \quad (18)$$

where ϕ denotes a feature (we skip indexes for readability) computed via integral images while ϕ^* its value computed by the definition. To give the reader an initial outlook, we remark that in our C++ implementation noticeable errors start to be observed already for $\rho{=}\varrho{=}8$ settings, but they do not yet affect significantly the detection accuracy. For $\rho{=}\varrho{=}10$, the frequency of relevant errors is already clear they do deteriorate accuracy. For example, for the sliding window of size 48×48, about 29.7% of all features have relative errors equal at least 25%. The numerically safe approach we are about to present reduces this fraction to 0.7%.

3.1 Piecewise Integral Images

The technique we propose for reduction of numerical errors is based on integral images that are defined *piecewise*. We partition every integral image into a number of adjacent pieces, say of size $W \times W$ (border pieces may be smaller due to remainders), where W is chosen to exceed the maximum allowed width for the sliding window. Each piece obtains its own "private" coordinate system. Informally speaking, the (j, k) indexes that are present in formula (14) become reset to $(0, 0)$ at top-left corners of successive pieces. Similarly, the values cumulated so far in each integral image $ii_{t,u}$ become zeroed at those points. Algorithm 1 demonstrates this construction. During detection procedure, global coordinates (j, k) can still be used in main loops to traverse the image, but once the window position gets fixed, say at (j_0, k_0), then we shall recalculate that position to new coordinates (j_0', k_0') valid for the current piece in the following manner:

$$N = \lfloor j_0/W \rfloor, \qquad M = \lfloor k_0/W \rfloor. \tag{19}$$

$$j_0' = j_0 - N \cdot W, \qquad k_0' = k_0 - M \cdot W. \tag{20}$$

Note that due to the introduced partitioning, the sliding window may cross partitioning boundaries, and reside in either: one, two or four pieces of integral images. That last situation is illustrated in Fig. 2. Therefore, in the general case, the outcomes of growth operations $\varDelta$ for the whole window will have to be combined from four parts denoted in the figure by P_1, P_2, P_3, P_4. An important role in that context will be played by the central index (j_c, k_c). In the original approach its position was calculated only once, using global coordinates and formula (12). The new technique requires that we "see" the central index differently from the point of view of each part P_i. We give the suitable formulas below, treating the first part P_1 as reference.

$$j_{c,P_1} = (2j_0' + w - 1)/2, \qquad k_{c,P_1} = (2k_0' + w - 1)/2. \tag{21}$$

$$j_{c,P_2} = j_{c,P_1}, \qquad k_{c,P_2} = k_{c,P_1} - W. \tag{22}$$

$$j_{c,P_3} = j_{c,P_1} - W, \qquad k_{c,P_3} = k_{c,P_1}. \tag{23}$$

$$j_{c,P_4} = j_{c,P_1} - W, \qquad k_{c,P_4} = k_{c,P_1} - W. \tag{24}$$

Note that the above formulation can, in particular, yield negative coordinates. More precisely, depending on the window position and the P_i part, the coordinates of the central index can range within: $-W + w/2 < j_{c,P_i}, k_{c,P_i} < 2W - w/2$.

Algorithm 1. Piecewise integral image

```
procedure PiecewiseIntegralImage($f$, $t$, $u$, $W$)
    create array $ii_{t,u}$ of size $n_x \times n_y$ and auxiliary array $ii$ of size $n_y$
    $k := 0$, $j := 0$
    for $x := 0, \dots, n_x - 1$ do
        if $x \bmod W = 0$ then
            $k := 0$
        for $y := 0, \dots, n_y - 1$ do
            if $y \bmod W = 0$ then
                $j := 0$
            $s := f(x, y)(k - ij)^t (k + ij)^u$
            if $j > 0$ then
                $ii[y] := ii[y-1] + s$
            else
                $ii[y] := s$
            if $k > 0$ then
                $ii_{t,u}[x, y] := ii_{t,u}[x-1, y] + ii[y]$
            else
                $ii_{t,u}[x, y] := ii[y]$
            $j := j + 1$
        $k := k + 1$
    return $ii_{t,u}$
```

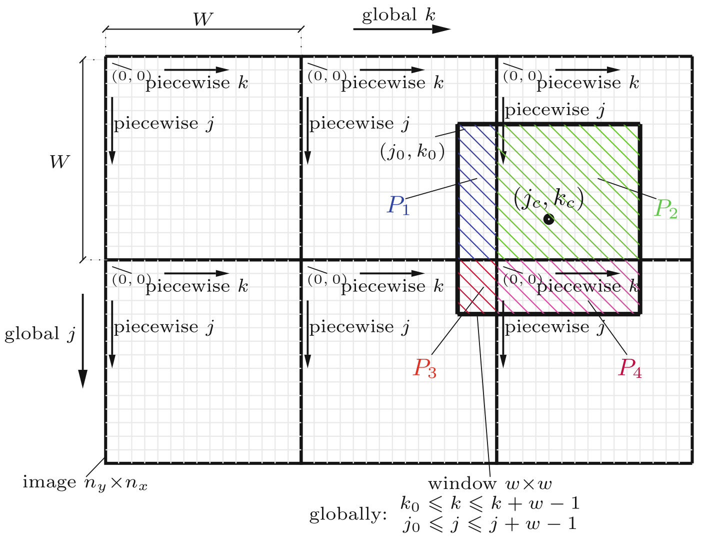

Fig. 2. Illustration of piecewise integral images. Growth operations must be in general combined from four summands corresponding to parts P_1, P_2, P_3, P_4 using redefined coordinates of the central index (j_c, k_c).

3.2 Numerically Safe Formula for ZMs

As the starting point for the derivation we use a variant of formula (11), taking into account the general window position with global pixel indexes ranging as follows: $j_0 \leqslant j \leqslant j_0{+}w{-}1$, $k_0 \leqslant k \leqslant k_0{+}w{-}1$. We substitute $\frac{4p+2o+2}{\pi w^2}$ into γ.

$$\begin{aligned}
\widehat{M}_{2p+o,2q+o} &= \gamma \sum_{\substack{j_0\leqslant j\leqslant j_0+w-1\\ k_0\leqslant k\leqslant k_0+w-1}}\sum f(j,k)\sum_{q\leqslant s\leqslant p}\beta_{2p+o,2q+o,p-s}\Bigg(\underbrace{\frac{\sqrt{2}}{w}(k{-}k_c)}_{x_k}+i\,\underbrace{\frac{\sqrt{2}}{w}(j_c{-}j)}_{y_j}\Bigg)^{s-q}\\
&\cdot\Bigg(\underbrace{\frac{\sqrt{2}}{w}(k{-}k_c)}_{x_k}-i\,\underbrace{\frac{\sqrt{2}}{w}(j_c{-}j)}_{y_j}\Bigg)^{s+q+o} = \gamma \sum_{\substack{j_0\leqslant j\leqslant j_0+w-1\\ k_0\leqslant k\leqslant k_0+w-1}}\sum f(j,k)\sum_{q\leqslant s\leqslant p}\beta_{2p+o,2q+o,p-s}\left(\frac{\sqrt{2}}{w}\right)^{2s+o}\\
&\cdot\Big(k{-}k_c+i(j_c{-}j)\Big)^{s-q}\Big(k{-}k_c-i(j_c{-}j)\Big)^{s+q+o} = \gamma \sum_{P\in\{P_1,\ldots,P_4\}}\ \sum_{(j_P,k_P)\in P}\sum f_P(j_P,k_P)\\
&\cdot\sum_{q\leqslant s\leqslant p}\beta_{2p+o,2q+o,p-s}\left(\frac{\sqrt{2}}{w}\right)^{2s+o}\big(k_P{-}k_{c,P}{+}i(j_{c,P}{-}j_P)\big)^{s-q}\big(k_P{-}k_{c,P}{-}i(j_{c,P}{-}j_P)\big)^{s+q+o}\\
&=\gamma\sum_{q\leqslant s\leqslant p}\beta_{2p+o,2q+o,p-s}\left(\frac{\sqrt{2}}{w}\right)^{2s+o}\sum_{t=0}^{s-q}\binom{s-q}{t}\sum_{u=0}^{s+q+o}\binom{s+q+o}{u}\sum_{P\in\{P_1,\ldots,P_4\}}\big(\text{-}k_{c,P}{+}ij_{c,P}\big)^{s-q-t}\\
&\cdot\big(\text{-}k_{c,P}{-}ij_{c,P}\big)^{s+q+o-u}\underbrace{\sum_{(j_P,k_P)\in P}\sum f_P(j_P,k_P)\,(k_P{-}ij_P)^t\,(k_P{+}ij_P)^u}_{\underset{P}{\Delta}(ii_{t,u})}. \qquad (25)
\end{aligned}$$

The second pass in the derivation of (25) splits the original summation into four smaller summations over window parts lying within different pieces of integral images, see Fig. 2. We write this down for the most general case when the window crosses partitioning boundaries along both axes. We remind that three simpler cases are also possible: $\{P_1\}$ (no crossings), $\{P_1, P_2\}$ (only vertical boundary crossed), $\{P_1, P_3\}$ (only horizontal boundary crossed). The parts are directly implied by: the offset (j_0, k_0) of detection window, its width w and pieces size W. For strictness, we could have written a functional dependence of form $P(j_0, k_0, w, W)$, which we skip for readability. Also in this pass we switch from global coordinates (j, k) to piecewise coordinates (j_P, k_P) valid for the given part P. The connection between those coordinates can be expressed as follows

$$j=\begin{cases} N{\cdot}W{+}j_P, & P\in\{P_1,P_2\};\\ (N{+}1){\cdot}W{+}j_P, & P\in\{P_3,P_4\};\end{cases}\quad k=\begin{cases} M{\cdot}W{+}k_P, & P\in\{P_1,P_3\};\\ (M{+}1){\cdot}W{+}k_P, & P\in\{P_2,P_4\}.\end{cases} \qquad (26)$$

In the third pass we apply two binomial expansions. In both of them we distinguish two groups of terms: $-k_{c,P} \pm ij_{c,P}$ (independent of the current pixel index) and $k_P \mp ij_P$ (dependent on it). Lastly, we change the order of summations and apply the constant-time Δ operation.

The key idea behind formula (25), regarding its numerical stability, lies in the fact that it uses decreased values of indexes j_P, k_P, $j_{c,P}$, $k_{c,P}$, comparing to the original approach, while maintaining the same differences $k_P - k_{c,P}$ and $j_{c,P} - j_P$. Recall the initial expressions $(k - k_c)\sqrt{2}/w$ and $(j_c - j)\sqrt{2}/w$ and note that one can introduce arbitrary shifts, e.g. $k := k \pm \alpha$ and $k_c := k_c \pm \alpha$ (analogically for the pair: j, j_c) as long as differences remain intact. Later, when indexes in such a pair become separated from each other due to binomial expansions, their numerical behaviour is safer because the magnitude orders are decreased.

Features with high errors can be regarded as damaged, or even useless for machine learning. In Table 1 and Fig. 3 we report percentages of such features for a given image, comparying the original and the numerically safer approach. The figure shows how the percentage of features with relative errors greater than 0.25, increases as the sliding window moves towards the bottom-right image corner.

Table 1. Percentages of damaged features (with relative errors at least 0.1, 0.25, 0.5) for a 640×480 image and different feature spaces and window sizes. Percentage values averaged over all possible window positions. Smaller percentages marked by gray color.

Type of integral images and window size	$\rho = \varrho = 6$ Relative errors:			$\rho = \varrho = 8$ Relative errors:			$\rho = \varrho = 10$ Relative errors:			$\rho = \varrho = 12$ Relative errors:		
	0.1	0.25	0.5	0.1	0.25	0.5	0.1	0.25	0.5	0.1	0.25	0.5
Original, 48×48	0.020	0.004	0.001	10.95	8.280	6.521	32.75	29.71	27.49	49.14	46.62	44.74
Piecewise, 48×48	0	0	0	0.0009	0	0	1.358	0.673	0.344	12.34	9.821	8.090
Original, 56×56	0.002	0	0	6.915	4.882	3.590	27.68	24.67	22.45	44.79	42.18	40.22
Piecewise, 56×56	0	0	0	0	0	0	0.370	0.128	0.049	8.039	5.929	4.539
Original, 68×68	0	0	0	3.187	1.887	1.120	21.08	18.07	15.86	38.79	35.95	33.80
Piecewise, 68×68	0	0	0	0	0	0	0.027	0.007	0.002	3.704	2.307	1.513
Original, 82×82	0	0	0	0.958	0.389	0.162	14.52	11.71	9.792	32.21	29.19	26.97
Piecewise, 82×82	0	0	0	0	0	0	0.001	0	0	1.179	0.560	0.258

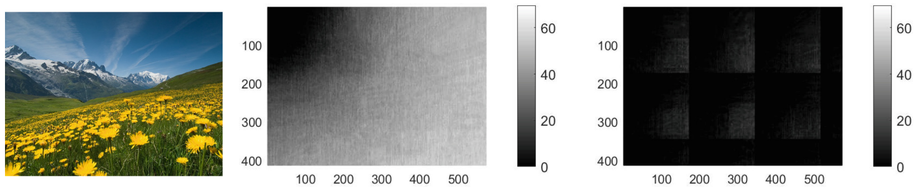

Fig. 3. Percentage of features with relative error at least 0.25 for each possible position of 68×68 sliding window: (left) input image of size 640×480, (middle) original approach, (right) piecewise approach with $W = 172$. Feature space settings: $\rho = \varrho = 12$.

4 Experiments

In subsequent experiments we apply *RealBoost* (RB) learning algorithm producing ensembles of weak classifiers: shallow *decision trees* (RB+DT) or *bins*

(RB+B), for details see [2,5,11]. Ensemble sizes are denoted by T. Two types of feature spaces are used: based solely on *moduli* of Zernike moments (ZMs-M), and based on *extended* Zernike product invariants (ZMs-E); see formulas (7), (8), respectively. The '-NER' suffix stands for: *numerical errors reduction*. Feature counts are specified in parenthesis. Hence, e.g. ZMs-E-NER (14250) represents extended Zernike invariants involving over 14 thousand features, computed according to formula (25) based on piecewise integral images.

Experiment: "Synthetic A letters" For this experiment we have prepared a synthetic data set based on font material gathered by T.E. de Campos et al. [2,3]. A subset representing 'A' letters was treated as positives (targets). In train images, only objects with limited rotations were allowed ($\pm 45°$ with respect to their upright positions). In contrast, in test images, rotations within the full range of 360° were allowed. Table 2 lists details of the experimental setup.

Table 2. "Synthetic A letters": experimental setup.

Train data		Test data		Detection procedure	
Quantity/Parameter	Value	Quantity/Parameter	Value	Quantity/Parameter	Value
No. of positives	20 384	No. of positives	417	Image resolution	600 × 480
No. of negatives	323 564	No. of negatives	3 745 966	No. of detection scales	5
Total set size	343 948	Total set size	3 746 383	Window growing coef.	1.2
		No. of images	200	Smallest window	100 × 100
				Largest window size	208 × 208
				Window jumping coef.	0.05

We report results in the following forms: examples of detection outcomes (Fig. 4), ROC curves over logarithmic FAR axis (Fig. 5a), test accuracy measures for selected best detectors (Table 4). Accuracy results were obtained by a batch detection procedure on 200 test images including 417 targets (within the total of 3 746 383 windows).

The results show that detectors based on features obtained by the numerically safe formula achieved higher classification quality. This can be well observed in ROCs, where solid curves related to the NER variant dominate dashed curves, and is particularly evident for higher orders ρ and ϱ (blue and red lines in Fig. 5a). These observations are also confirmed by Table 4

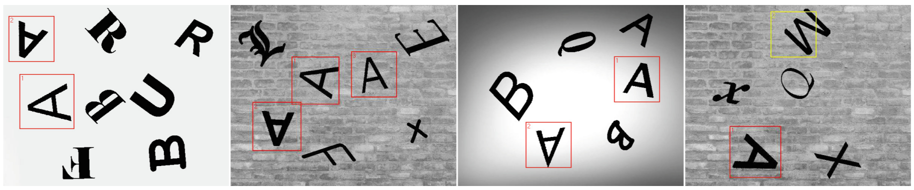

Fig. 4. "Synthetic A letters": detection examples. Two last images show error examples—a misdetection and a false alarm (yellow). (Color figure online)

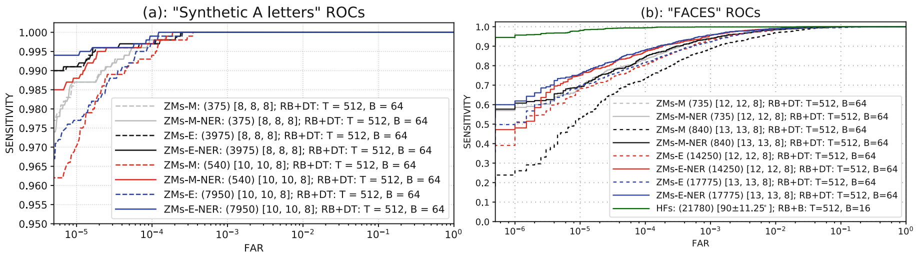

Fig. 5. (a): "Synthetic A letters": ROC curves for test data, (b) "Faces": ROC curves for test data for prescreeners and one example of angle-dependent classifier.

Experiment: "Faces" The learning material for this experiment consisted of 3 000 images with faces (in upright position) looked up using the *Google Images* and produced a train set with 7 258 face examples (positives). The testing phase was a two-stage process, consisting of preliminary and final tests. Preliminary tests were meant to generate ROC curves and make an initial comparison of detectors. For this purpose we used another set of faces (also in upright position) containing 1 000 positives and 2 000 000 negatives. The final batch tests were meant to verify the rotational invariance. For this purpose we have prepared the third set, looking up faces in unnatural rotated positions (example search queries: "people lying down in circle", "people on floor", "face leaning to shoulder", etc.). We stopped after finding 100 such images containing 263 faces in total (Table 3).

Table 3. "Faces": experimental setup.

Train data (upright faces)		Final test data (rotated faces)		Detection procedure	
Quantity/Parameter	Value	Quantity/Parameter	Value	Quantity/Parameter	Value
No. of positives	7 258	No. of positives	263	Image height	480
No. of negatives	100 000	No. of negatives	14 600 464	Window growing coef	1.2
Total set size	107 258	Total set size	14 600 564	Smallest window	48 × 48
		No. of images	100	Largest window	172 × 172
				Window jumping coefficient	0.05

Zernike Invariants as Prescreeners. Preliminary tests revealed that classifiers trained on Zernike features were *not* accurate enough to work as standalone face detectors invariant to rotation. ROC curves (Fig. 5b) indicate that sensitivities at satisfactory levels of ≈90% are coupled with false alarm rates $\approx 2{\cdot}10^{-4}$, which, though improved with respect to [2][3], is still too high to accept. Therefore, we decided to apply obtained detectors as *prescreeners* invariant to rotation. Candidate windows selected via prescreening[4] were then analyzed by angle-dependent classifiers. We prepared 16 such classifiers, each responsible

[3] In [2] the analogical false alarm rates were at the level $\approx 5 \cdot 10^{-3}$.
[4] We chose decision thresholds for prescreeners to correspond to 99.5% sensitivity.

for an angular section of 22.5°, trained using 10 125 HFs and RB+B algorithm. Figure 5b shows an example of ROC for an angle-dependent classifier (90° ± 11.25°). The prescreening eliminated about 99.5% of all windows for ZMs-M and 97.5% for ZMs-E.

Table 4. Batch detection results for: "Synthetic A letters" and "Faces".

"Synthetic A letters"					"Faces"				
Feature space (no. of feats.)	Sensitivity	FAR per image	FAR per window [$\cdot 10^{-6}$]	Accuracy per window	Feature space (no. of feats.)	Sensitivity	FAR per image	FAR per window [$\cdot 10^{-7}$]	Accuracy per window
ZMs-M (375)	.979	8/200	2.13	.999995734	ZMs-M (735)	.890	14/100	9.6	.999997054
ZMs-M-NER (375)	.979	8/200	2.13	.999995734	ZMs-M-NER (735)	.905	16/100	11.0	.999997192
ZMs-M (540)	.966	17/200	4.53	.999992001	ZMs-M (840)	.897	15/100	10.3	.999997123
ZMs-M-NER (540)	.987	8/200	2.13	.999996534	ZMs-M-NER (840)	.905	8/100	5.5	.999997740
ZMs-E (3975)	.982	3/200	0.80	.999997334	ZMs-E (14250)	.867	6/100	4.1	.999997192
ZMs-E-NER (3975)	.979	2/200	0.53	.999997334	ZMs-E-NER (14250)	.878	8/100	5.5	.999997260
ZMs-E (7950)	.971	9/200	2.40	.999994667	ZMs-E (17775)	.897	6/100	4.1	.999997740
ZMs-E-NER (7950)	.992	0/200	0	.999999200	ZMs-E-NER (17775)	.875	10/100	6.8	.999997055

Table 4 reports detailed accuracy results, while Fig. 6 shows detection examples. It is fair to remark that this experiment turned out to be much harder than the former one. Our best face detectors based on ZMs and invariant to rotation have the sensitivity of about 90% together with about 7% of false alarms per image, which indicates that further fine-tuning or larger training sets should be considered. As regards the comparison between standard and NER variants, as previously the ROC curves show a clear advantage of NER prescreeners (solid curved dominate their dashed counterparts). Accuracy results

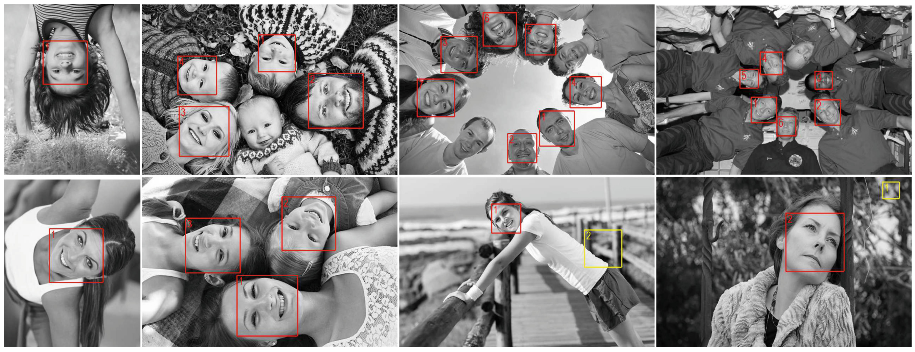

Fig. 6. "Faces": detection examples. Images prescreened with ZMs-E features then processed with angle-dependent classifiers (HFs).

presented in Table 4 also show such a general tendency but are less evident. One should remember that these results pertain to combinations: 'prescreener (ZMs) + postscreener (HFs)'. An improvement in the prescreener alone may, but does not have to, influence directly the quality of the whole system.

5 Conclusion

We have improved the numerical stability of floating-point computations for Zernike moments (ZMs) backed with piecewise integral images. We hope this proposition can pave way to more numerous detection applications involving ZMs. A possible future direction for this research could be to analyze the computational overhead introduced by the proposed approach. Such analysis can be carried out in terms of *expected value* of the number of needed growth operations.

References

1. Acasandrei, L., Barriga, A.: Embedded face detection application based on local binary patterns. In: 2014 IEEE International Conference on High Performance Computing and Communications (HPCC, CSS, ICESS), pp. 641–644 (2014)
2. Bera, A., Klęsk, P., Sychel, D.: Constant-time calculation of Zernike moments for detection with rotational invariance. IEEE Trans. Pattern Anal. Mach. Intell. **41**(3), 537–551 (2019)
3. de Campos, T.E., et al.: Character recognition in natural images. In: Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, pp. 273–280 (2009)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 886–893. IEEE Computer Society (2005)
5. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Stat. **28**(2), 337–407 (2000)
6. Goldberg, D.: What every computer scientist should know about floating-point arithmetic. ACM Comput. Surv. **23**(1), 5–48 (1991)
7. Klęsk, P.: Constant-time fourier moments for face detection—can accuracy of haar-like features be beaten? In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2017. LNCS (LNAI), vol. 10245, pp. 530–543. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59063-9_47
8. Malek, M.E., Azimifar, Z., Boostani, R.: Facial age estimation using Zernike moments and multi-layer perceptron. In: 22nd International Conference on Digital Signal Processing (DSP), pp. 1–5 (2017)
9. Mukundan, R., Ramakrishnan, K.: Moment Functions in Image Analysis - Theory and Applications. World Scientific, Singapore (1998)
10. Rajaraman, V.: IEEE standard for floating point numbers. Resonance **21**(1), 11–30 (2016). https://doi.org/10.1007/s12045-016-0292-x
11. Rasolzadeh, B., et al.: Response binning: improved weak classifiers for boosting. In: IEEE Intelligent Vehicles Symposium, pp. 344–349 (2006)
12. Singh, C., Upneja, R.: Accurate computation of orthogonal Fourier-Mellin moments. J. Math. Imaging Vision **44**(3), 411–431 (2012)

13. Sornam, M., Kavitha, M.S., Shalini, R.: Segmentation and classification of brain tumor using wavelet and Zernike based features on MRI. In: 2016 IEEE International Conference on Advances in Computer Applications (ICACA), pp. 166–169 (2016)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Conference on Computer Vision and Pattern Recognition (CVPR 2001), pp. 511–518. IEEE (2001)
15. Xing, M., et al.: Traffic sign detection and recognition using color standardization and Zernike moments. In: 2016 Chinese Control and Decision Conference (CCDC), pp. 5195–5198 (2016)
16. Yin, Y., Meng, Z., Li, S.: Feature extraction and image recognition for the electrical symbols based on Zernike moment. In: 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), pp. 1031–1035 (2017)
17. Zernike, F.: Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. Physica **1**(8), 668–704 (1934)