Rotationally invariant object detection on video using Zernike moments backed with integral images and frame skipping technique

Aneta Bera$^{1[0000-0002-0456-9451]}$, Dariusz Sychel$^{1[0000-0001-9835-869X]}$, and Przemysław Klęsk$^{1[0000-0002-5579-187X]}$

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

Abstract. This is a follow-up study on Zernike moments applicable in detection tasks owing to a construction of complex-valued integral images that we have proposed in [3]. The main goal of the proposition was to calculate the mentioned features fast (in constant-time). The proposed solution can be applied with success when dealing with single images, however it is still too slow to be used in real-time applications, for example in video processing. In this work we attempted to solve mentioned problem.
In this paper we propose a technique in order to reduce the detection time in real-time applications. The degree of reduction is controlled by two parameters: *fs* (related to the gap between frames that undergo a full scan) and *nb* (related to the size of neighborhood to be searched on non-fully scanned frames). We present a series of experiments to show how our solution performs in terms of both detection time and accuracy.

Keywords: Zernike moments · Complex-Valued Integral Images · Detection Time 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 [13], commonly applied in image recognition tasks, are often precluded from detection, as they involve inner products, i.e., linear-time computations with respect to the number of pixels. This problem becomes more evident when detecting objects on video. Also, the deep learning approaches cannot be applied directly in dense detection procedures (sliding window-based), 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

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

(LBPs) and histogram of oriented gradients (HOG) descriptor are state-of-the-art examples from this category [1,5,17]. The crucial algorithmic trick that underlies these methods and allows 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 the so-called 'growth' operations. Each growth involves two additions and one subtraction using four entries of an integral image.

In our previous work [3] we have introduced mentioned integral images in order to compute Zernike moments (ZMs). We prepared a set of special complex-valued integral images and an algorithm that allows to calculate Zernike moments fast, namely in constant time. Thanks to the proposed solution, Zernike moments become suitable for dense detection procedures, where the image is scanned by a sliding window at multiple scales, and where rotational invariance is required at the level of each window. In [8] we indicated numerically fragile places in our algorithm and identified their causes. Then, in order to reduce numerical errors, we propose piecewise integral images and derive a numerically safer formula for Zernike moments. Moreover, in [9] we enrich derived initial idea by proposing an extended space of Zernike invariants backed with integral images. This feature space includes not only the moduli of Zernike moments but also real and imaginary parts of suitable moment products. All mentioned solutions were supported by a series of experiments.

Recent literature confirms that ZMs are still popular and used in many applications. In [7] authors used ZMs with K-nearest Neighbors for leaf recognition, in [10] authors used selected ZMs to determine the rotation angle of the objects. Authors of [21] proposed to use ZMs and support vector machine for brain tumor diagnosis. ZMs are applied in many other image recognition tasks e.g: human age estimation [12] or traffic signs recognition [19]. The authors of [11] proposed algorithm using image normalization and Zernike moments which allows to recognize stars based on telescope images. This solution allows to assign stars to their position in catalog. In 2020 the authors of [18] presented a way to match terrain using Zernike moments and HOG descriptors based on data from Synthetic Aperture Radar (SAR) and REM Radar. Yet, it is quite difficult to find examples of detection tasks applying ZMs directly.

Zernike moments are also used for detection task of objects in motion. E.g [16] shows a way for detection of doubtful or uncommon actions in video sequence based on Zernike moments and Canny edge detector. In [2] authors used motion energy image (MEI) with Zernike moments in order to detect humans actions. In [22], the authors proposed to use two particular Zernike moments (selected by them) in order to detect moving objects. It is worth noting that also in this task Zernike moments were not applied directly on video frames but after some kind of preprocessing or feature selection.

In this paper we address the problem of using Zernike moments for object detection on video, which requires to calculate them faster than according to the original algorithm proposed in [3].

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

2 Zernike moments theory and calculation using integral images

2.1 Zernike moments

Zernike moments (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 [20,3]. p and q indexes, represent moment order, hence they must be simultaneously even or odd, moreover $p \geqslant |q|$.

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

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

ZMs are invariant to scale transformations, but only their absolute value is invariant to rotation. Proof of these properties were presented in [3].

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

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

2.2 Zernike moments in detection task

In practical tasks it is more convenient to work with rectangular, rather than circular, image fragments. Singh and Upneja [15] proposed a workaround to this problem: a square of size $w \times w$ pixels (w is even) becomes *inscribed* in the unit disc, as shown in Fig. 1, and zeros are "laid" over the square-disc complement. This reduces integration over the disc to integration over the square.

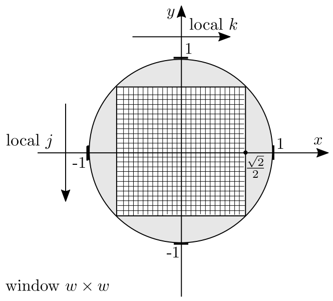

Fig. 1. The trick from [15] to calculate OFMMs: square image window inscribed in the unit circle, zero values laid in the complement of square.

2.3 Constant-time Calculation of Zernike Moments

In [3] we proposed a way to calculate Zernike moments using set of multiple integral images in order to use those features in reasonable time, during detection procedure based on sliding window technique.

Let's concentrate on a scenario of a computer detection procedure. Suppose a digital image of size $n_x \times n_y$ is traversed by a sliding window of size $w \times w$, where w is even (for clarity we discuss only a single-scale scan within the detection procedure). The situation is presented in Fig. 2. Let (j, k) denote global coordinates of a pixel in the image. For each window under analysis, its offset — the top left corner of the window — will be denoted by (j_0, k_0). Therefore, the indexes of pixels that belong to the window are: $j_0 \leqslant j \leqslant j_0 + w - 1$, $k_0 \leqslant k \leqslant k_0 + w - 1$. Additionally, it will be convenient to introduce a notation (j_c, k_c) for the *central index* of the window:

$$j_c = \frac{1}{2}(2j_0 + w - 1), \quad k_c = \frac{1}{2}(2k_0 + w - 1). \tag{6}$$

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

$$ii_{t,u}(l,m) = \sum_{\substack{0 \leqslant j \leqslant l \\ 0 \leqslant k \leqslant m}} \sum f(j,k)(k - ij)^t (k + ij)^u, \quad {\scriptstyle 0 \leqslant l \leqslant n_y - 1 \atop 0 \leqslant m \leqslant n_x - 1}; \tag{7}$$

[2] In [3] we have proved that integral images $ii_{t,u}$ and $ii_{u,t}$ are complex conjugates at all points, which allows for computational savings.

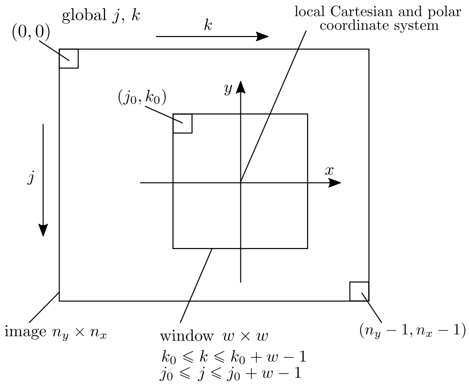
global j, k
$(0,0)$
k
local Cartesian and polar coordinate system
(j_0,k_0)
y
x
j
image $n_y \times n_x$
window $w \times w$
$k_0 \leqslant k \leqslant k_0+w-1$
$j_0 \leqslant j \leqslant j_0+w-1$
(n_y-1, n_x-1)

Fig. 2. Illustration of detection procedure using sliding window.

Proposition 1 *Suppose a set of integral images $\{ii_{t,u}\}$, defined as in (7), has been prepared prior to the detection procedure. Then, for any square window in the image, each of its Zernike moments can be calculated in constant time — $O(1)$, regardless of the number of pixels in the window, as follows:*

$$\widehat{M}_{2p+o,2q+o} = \frac{4p+2o+2}{\pi w^2} \sum_{2q+o\leqslant 2s+o\leqslant 2p+o} \beta_{2p+o,2q+o,p-s} \left(\frac{\sqrt{2}}{w}\right)^{2s+o}$$
$$\cdot \sum_{t=0}^{s-q} \binom{s-q}{t} (-k_c+ij_c)^{s-q-t} \sum_{u=0}^{s+q+o} \binom{s+q+o}{u} (-k_c-ij_c)^{s+q+o-u} \underset{\substack{j_0,j_0+w-1\\ k_0,k_0+w-1}}{\Delta}(ii_{t,u}). \quad (8)$$

The proof of this is presented in detail in [3].

2.4 Numerical errors and their reduction

Formula (8) contains two numerically fragile places. The first one are integral images themselves, defined by (7) where 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 $10^{3(t+u)}$. Obviously, the rounding-off errors amplify as the $ii_{t,u}$ sum progresses towards the bottom-right image corner.

The second fragile place are: $(-k_c+ij_c)^{s-q-t}$ and $(-k_c-ij_c)^{s+q+o-u}$, involving the central index, see (8). Their products can too become very large in magnitude as computations move towards the bottom-right image corner.

The solution to this numerical problem, presented in [8], is based on integral images that are defined *piecewise*. We partitioned 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 was chosen to exceed the maximum allowed width

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

for the sliding window. Each piece obtains its own "private" coordinate system. Informally speaking, the (j, k) indexes that are present in formula (7) become reset to $(0, 0)$ at top-left corners of successive pieces. Similarly, the values accumulated so far in each integral image $ii_{t,u}$ become zeroed at those points. For more details see [8].

2.5 Extended feature space of Zernike invariants

During our previous research [9] we proposed a technique to extend the feature space. The central role is played by expression for generating the invariants:

$$M_{p,q}{}^{n}\ M_{v,s}, \quad nq + s = 0.$$

In that context we considered which tuples (n, p, q, v, s) should be allowed into the final collection of feature indexes and what information they carry. We distinguished and presented several groups among them.

Remembering that $M_{p,q}{}^{n}\ M_{v,s}$ is a complex number, we used both its *real* and *imaginary* parts as separate features, if it was possible. For that purpose we extended the tuples to consist of six members: (n, p, q, v, s, i), where the last index $i \in \{0, 1\}$ indicates, whether we used real or imaginary part of $M_{p,q}{}^{n}\ M_{v,s}$.

Knowing the indexation scheme, we presented the actual extraction procedure to be invoked for each analyzed window in (see [9] for details).

Table 1 shows counts of features in both extended and non-extended spaces. Both spaces may constitute a useful input information for machine learning and rotationally-invariant detection.

Table 1. Number of features in extended (black) and non-extended (gray) feature spaces.

ρ \ ϱ	0	1	2	3	4	5	6
0	1 1						
1	1 1	2 2					
2	2 2	3 3	6 4				
3	2 2	6 4	11 5	16 6			
4	4 3	8 5	20 7	25 8	34 9		
5	4 3	13 6	29 8	45 10	56 11	63 12	
6	7 4	16 7	43 10	59 12	87 14	94 15	111 16

Proposed solution demonstrates how to generate a large number of constant-time Zernike invariants using computations supported by integral images (complex-valued). Thanks to that, we can provide many useful features which is a beneficial for machine learning.

3 Frame skipping technique

Zernike invariants, even with the computationally fast form (with use of integral images), are still not fast enough to be applied for object detection on video. In this section we present an approach that allows to reduce the detection time for that purpose.

The idea is based on performing a full scan on the image from the camera, but not on each frame. For better explanation, let's assume that every 5th frame is scanned fully in order to find (detect) some objects. Once this is done we need to memorize their positions in the image. With this information at disposal, in the next frame the sliding window shall only be placed in close neighborhood of detected objects, and once they are redetected their positions can be updated. For yet another frame, the sliding window shall take advantage of neighborhoods from two previous frames, etc. After 4 frames, the image would be re-scanned, which would enable, e.g., finding objects that just appeared on it.

We will now proceed to a more detailed description of the proposed solution.

For the purpose of presented solution we introduce two parameters fs and nb. The first of them fs determines how many frames the full search of the entire image takes place, e.g., if the value of this parameter is 4, there are 3 frames among the full scans for which the full scan will not be performed. When $fs = 1$, a full scan will occur every frame. The second parameter, neighborhood (radius), determines size of area around the found windows with objects that will be searched — in case of images from camera without a full search. For Example, if window with found object has size 100 × 100 of pixels, nb is set to 0.5, it means the searched area will be of size 200 × 200 — we are adding 50

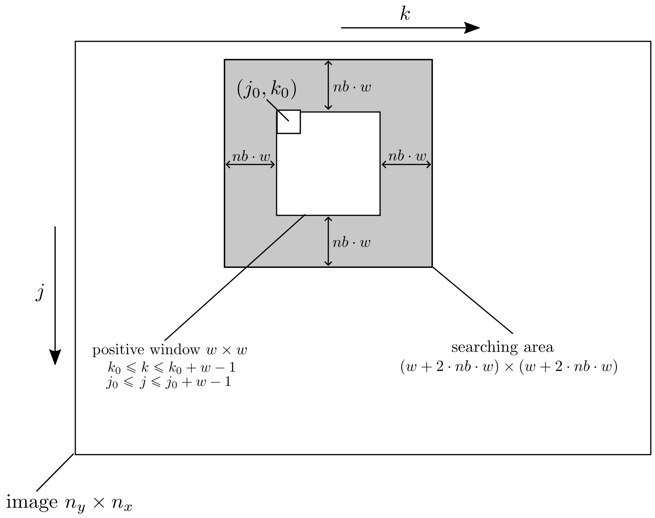
k
(j_0, k_0)
$nb \cdot w$
$nb \cdot w$
$nb \cdot w$
$nb \cdot w$
j
positive window $w \times w$
$k_0 \leqslant k \leqslant k_0 + w - 1$
$j_0 \leqslant j \leqslant j_0 + w - 1$
searching area
$(w + 2 \cdot nb \cdot w) \times (w + 2 \cdot nb \cdot w)$
image $n_y \times n_x$

Fig. 3. Illustration of nb parameter for detection procedure using sliding window and frame skipping technique.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

pixels (50% from 100) to each side of window marked as positive. Note that, the size of searched area depends on the window size marked as positive. The bigger the window, the bigger the area that will be searched.

Figure 3 presents a situation where one positive window was found in a full image scan. It is easy to see the advantages of using the presented solution. Instead of searching the entire $n_y \times n_x$ image with a sliding window (which comes in various sizes and scales), we will only search the area around the positive window, i.e., $w \cdot (1 + 2 \cdot nb) \times w \cdot (1 + 2 \cdot nb)$, which will save us a lot of time. This partial scan will be performed for $fs - 1$ frames.

Algorithm 1 Detection procedure with frame skipping technique

```
procedure DetectObject($\mathcal{I}$, $\mathcal{W}_d$, $fr_n$, $nb$, $fs$, clf)
        ▷ $\mathcal{W}_d$ contain coordinates of windows classified as positive on previous step.
    Create list $\mathcal{W}_r$ for storing coordinates of windows classified as positive.
    if $fr_n$ mod $fs = 0$ then
        for $w \in$ GenDetCoords($\mathcal{I}_w$, $\mathcal{I}_h$, $detection\_parameters$) do
            Use $cls$ to classify fragment of $\mathcal{I}$ at coordinates $w$.
            if $cls$ return 1 then
                Append $w$ to $\mathcal{W}_r$.
    else
        if Areas contained in $\mathcal{W}_d$ intersect with each other then
            Merge areas to avoid redundancy.
        for $w \in \mathcal{W}_d$ do
            Expand area represented by $w$ by adding margin equal to $nb$
            for $coords \in$ GenDetCoords($w_{expand}$, $detection\_parameters$) do
                Use $cls$ to classify fragment of $\mathcal{I}$ at coordinates $coords$.
                if $cls$ return 1 then
                    Append $coords$ to $\mathcal{W}_r$.
    return $\mathcal{W}_r$
```

Algorithm 1 presents a detection procedure using frame skipping technique. clf represents a classifier, fr_n is frame number, nb neighborhood, fs skip, I video frame. Wd is a coordinate vector in the form of $[x, y, w, h]$, which describes the area that needs to be processed by standard detection procedure. In code below you can see two *GenDetCoords* procedure calls. The first one is a standard detection procedure, where whole the video frame is processed. The second one generates coordinates only in indicated areas — specified by positive frames and neighborhood.

4 Experiments

4.1 Learning algorithms and general settings

In experiments we apply *RealBoost+bins* as the main learning algorithm producing ensembles of weak classifiers. Each weak classifier is based on a single selected feature. Bins are equally wide and set up regularly, once 1% of outliers has been removed. The responses of classifiers are real-valued and calculated using the logit transform. For more details we address the reader, e.g., to [6,14].

It is worth remarking that we use Jaccard index (ratio of intersection and union areas) in two places in experiments: (1) to postprocess detected windows and (2) to check positive indications against the ground truth in tests. Typically, a detector produces a cluster of many positive windows around each target. At the postprocessing stage, we group such clusters into single indications. This means that at each step within the postprocessing loop, two windows with the highest Jaccard index become averaged. Later, when comparing positive indications against the ground truth, we expect each indication to have an index of at least 0.5 with respect to some target position in order to be counted as a true positive. Otherwise, it becomes a false alarm.

In experiment we have arranged a data set containing capital letters from the modern English alphabet. Pictures containing the characters from computer fonts were retrieved from the set prepared by T.E. de Campos et al. [4]. We have limited the subset representing the letter ‘A’ to several fonts with similar characteristics and treated it as our base for creating positive examples. Subsets with other letters were combined to prepare the negative examples.

Fig. 4 depicts the source graphical material used in the experiment. Images, for both training and testing, were generated by randomly placing objects over random backgrounds.

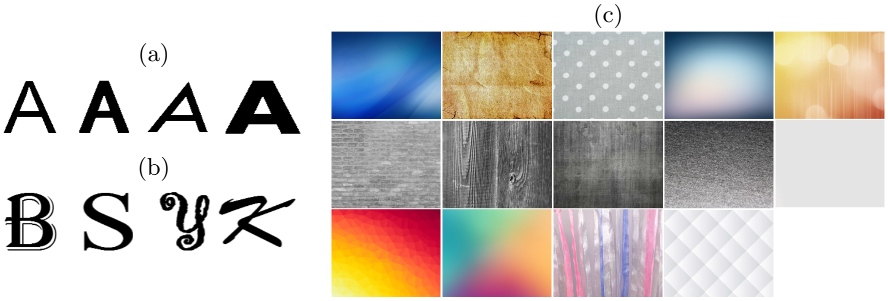

Fig. 4. Sample images and all backgrounds used to generate the data. Positives: letters A (a), negatives: other letters (b), backgrounds (c).

In training images, letters were allowed to rotate randomly within a limited range of ±45°. In the testing material letters were allowed to rotate randomly within the full range of 360°.

For detection procedure we created a video sequence from an image. We set a certain frame size, and move it around the given image and with the offset set according to pattern presented in Fig. 5.

Tab. 2 presents the experimental setup.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

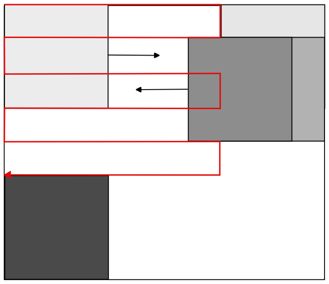

Fig. 5. Illustration of how video from image was generated.

Table 2. Experimental setup.

train data		
quantity / parameter	value	additional information
no. of positive examples	20 384	windows with letter ‘A’
no. of negative examples	323 564	windows with letters other than ‘A’ plus random samples of backgrounds
train set size	343 948	positive and negative examples in total
test data		
no. of images	100	
no. of positive examples	213	windows with letter ‘A’
no. of negative examples	1 858 587	implied by detection procedure
test set size	1 858 800	positive and negative examples in total
no. of negative examples	100 419	sampled on random from negatives
detection procedure (scanning with a sliding window)		
video frame resolution	640 × 480	imposed resolution
no. of detection scales	5	
window growing coefficient	1.2	
smallest and largest window size	100 × 100, 208 × 208	
window jumping coefficient	0.05	
$frame\_skip$ (fs)	$\{1, 5, 10\}$	
$radius$ (nb)	$\{15, 30, 50, 100\}$	

When reporting results of experiments, we use the following names: M, M-NER, E and E-NER — they define the type of features that was used. M stands for the moduli of Zernike moments, E extended product invariants. The ’-NER’ suffix stands for numerical errors reduction. r parameter suggests that ring-based variation to increase number of features (by a factor of $2R-1$) was used. Experiments were conducted parallel on Intel Xeon E5-2699 v4 CPU, 22/44 core/thread, 55MB cache.

4.2 Detection performance

Tables 3, 4, 5, 6 present results of detection performance. FPS stands for ’frames per second’, *total* is the total time of detection on whole video and *avg* is the average time of detection for single video frame.

As can be seen, both nb and fs have influence on time performance. The lower the value of nb is, the more significantly the time of the detection procedure decreases. The higher the value of fs, the lower the time of detection. It is worth paying a particular attention to FPS parameter. For the traditional detection

Table 3. Detection performance for moduli of Zernike moments and $p = 8$, $q = 8$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$			$nb = 0.3$			$nb = 0.4$		
M	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	1.5	32352	652	1.5	34081	686	1.5	33462	676	1.6	31942	643
5	7.4	8756	135	6.4	9142	156	5.8	9444	171	5.3	9953	188
10	13.7	6183	73	12	6155	83	10.2	5970	98	8.6	6498	116
M-NER	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	0.4	128685	2658	0.3	149028	3085	0.3	151250	3127	0.4	129931	2683
5	1.6	31627	611	1.6	31079	623	1.6	31771	638	1.4	35805	726
10	3.4	16683	293	3.3	15841	303	2.7	188893	374	2.5	20481	406

Table 4. Detection performance for extended product invariants of Zernike moments and $p = 8$, $q = 8$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$			$nb = 0.3$			$nb = 0.4$		
E	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	0.5	97185	2000	0.5	100212	2067	0.5	103255	2127	0.5	97824	2012
5	2.2	23936	457	2.2	23521	463	2	24984	498	1.8	27708	560
10	4.9	12249	204	4.1	13457	245	3.5	14595	286	3.2	16042	316
E-NER	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	0.3	157316	3252	0.3	166613	3446	0.3	178216	3690	0.3	157205	3251
5	1.5	33360	652	1.4	36835	740	1.3	37729	768	1.2	41399	841
10	2.7	20094	366	2.5	20865	403	2.5	20437	408	2.1	23746	477

procedure with sliding window and Zernike moments, its value is never bigger than 2. When using the proposed solution with frame skipping technique, the FPS was even 13.7 for moduli of Zernike moments.

Table 3 presents the results obtained for moduli of Zernike moments (with and without numerical error reduction) and $p = 8$, $q = 8$, $r = 8$. For features of type M we can observe the greatest gain in terms of time performance. The algorithm was able to process 13.7, 12, 10.2 and 8.6 frames per second, respectively for nb being set to 0.1, 0.2, 0.3 or 0.4 and fs equals to 10. For fs set to 5, we can also see a noticeable decrease in window processing time.
The time needed to generate features in $M - NER$ version is greater, and thus the profit from introducing modification is smaller, but still visible.

Table 5. Detection performance for moduli of Zernike moments and $p = 10$, $q = 10$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$			$nb = 0.3$			$nb = 0.4$		
M	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	1	49575	1013	1	51485	1052	0.9	51718	1054	1	48780	996
5	4.6	12632	218	4.1	12878	243	3.9	13482	257	3.6	14223	276
10	9.3	7790	107	7.8	7782	129	6.8	7992	146	5.9	8976	169
M-NER	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	0.3	187889	3892	0.2	198526	4110	0.2	196523	4070	0.3	179710	3725
5	1.3	33334	790	1.2	42119	860	1.1	43122	878	1	48386	984
10	1.9	27711	523	1.9	25793	520	2	24524	493	1.8	27313	551

Table 4 presents the results obtained for extended product invariants of Zernike moments (with and without numerical error reduction) and $p = 8$, $q = 8$,

$r = 8$. In this case, the base time was lower, and therefore the time after the modification also. However, you can see that for the parameter set to 10, the number of windows processed per second increased in every case.

Tables 5 and 6 present detection performance for moduli and extended product of Zernike moments, for $p = 10$, $q = 10$, $r = 8$. The results in the tables confirm the previously described observations.

Table 6. Detection performance for extended product invariants of Zernike moments and $p = 10$, $q = 10$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$			$nb = 0.3$			$nb = 0.4$		
E	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	0.3	150883	3119	0.3	147192	3045	0.3	152779	3131	0.3	1568289	3240
5	1.4	35619	701	1.4	35215	711	1.3	37181	755	1.2	39936	812
10	2.8	19649	353	2.6	19637	379	2.3	21447	426	2	25020	497
E-NER	FPS	total	avg	FPS	total	avg	FPS	total	avg	FPS	total	avg
1	0.2	249028	5161	0.2	266057	5517	0.2	265249	5500	0.2	256278	5313
5	1	51050	1024	0.9	54793	1121	0.9	56915	1164	0.8	63597	1305
10	1.6	28700	540	1.7	28998	576	1.5	32045	650	1.4	35738	720

4.3 Detection accuracy

Tables 7, 8, 9, 10 show accuracy of our detection experiments. Table 7 presents the results obtained for moduli of Zernike moments (with and without numerical error reduction) and $p = 8$, $q = 8$, $r = 8$. For the version of features without numerical error reduction (M) it can be seen that the sensitivity stabilizes for $nb = 0.3$. For lower values, especially for 0.1, there is a significant decrease in sensitivity compared to the version without using the proposed solution. For

Table 7. Detection results for moduli of Zernike moments and $p = 8$, $q = 8$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$		
M	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999993831	1	$6.169 \cdot 10^{-6}$	0.999993831	1	$6.169 \cdot 10^{-6}$
5	0.999948604	0.25	$1.028 \cdot 10^{-6}$	0.99999486	0.95833333	$1.028 \cdot 10^{-6}$
10	0.999937297	0.11458333	$1.028 \cdot 10^{-6}$	0.999977384	0.72916667	$1.028 \cdot 10^{-6}$
M-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999997944	1	$2.056 \cdot 10^{-6}$	0.999997944	1	$2.056 \cdot 10^{-6}$
5	0.99995066	0.27	$1.028 \cdot 10^{-6}$	0.999996916	0.97916667	$1.028 \cdot 10^{-6}$
10	0.99993825	0.125	$1.028 \cdot 10^{-6}$	0.999986635	0.875	$1.028 \cdot 10^{-6}$
fs	$nb = 0.3$			$nb = 0.4$		
M	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999993831	1	$6.169 \cdot 10^{-6}$	0.999993831	1	$6.169 \cdot 10^{-6}$
5	0.999996916	0.97916667	$1.028 \cdot 10^{-6}$	0.999996916	0.979	$1.028 \cdot 10^{-6}$
10	0.999986635	0.875	$1.028 \cdot 10^{-6}$	0.999986635	0.875	$1.028 \cdot 10^{-6}$
M-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999997944	1	$2.056 \cdot 10^{-6}$	0.999997944	1	$2.056 \cdot 10^{-6}$
5	0.999996916	0.97916667	$1.028 \cdot 10^{-6}$	0.999996916	0.97916667	$1.028 \cdot 10^{-6}$
10	0.999986635	0.875	$1.028 \cdot 10^{-6}$	0.999986635	0.875	$1.028 \cdot 10^{-6}$

Table 8. Detection results for extended product invariants of Zernike moments and $p = 8$, $q = 8$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$		
E	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999990744	1	$9.256\cdot10^{-6}$	0.999990744	1	$9.256\cdot10^{-6}$
5	0.999956827	0.39583333	$2.057\cdot10^{-6}$	0.999994859	0.97916667	$3.085\cdot10^{-6}$
10	0.999941408	0.20833333	$2.057\cdot10^{-6}$	0.999998458	0.875	$3.085\cdot10^{-6}$
E-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999993831	1	$6.169\cdot10^{-6}$	0.999993831	1	$6.169\cdot10^{-6}$
5	0.999958832	0.41666667	$1.028\cdot10^{-6}$	0.999996916	0.97916667	$1.028\cdot10^{-6}$
10	0.999942437	0.20833333	$1.028\cdot10^{-6}$	0.999986636	0.875	$1.028\cdot10^{-6}$
fs	$nb = 0.3$			$nb = 0.4$		
E	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999990744	1	$9.256\cdot10^{-6}$	0.999990744	1	$9.256\cdot10^{-6}$
5	0.999995887	0.97916667	$2.056\cdot10^{-6}$	0.999995887	0.97916667	$2.056\cdot10^{-6}$
10	0.999985604	0.875	$2.057\cdot10^{-6}$	0.999985604	0.875	$2.057\cdot10^{-6}$
E-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999993831	1	$6.169\cdot10^{-6}$	0.999993831	1	$6.169\cdot10^{-6}$
5	0.999996916	0.97916667	$1.028\cdot10^{-6}$	0.999996916	0.97916667	$1.028\cdot10^{-6}$
10	0.999986636	0.875	$1.028\cdot10^{-6}$	0.999986636	0.875	$1.028\cdot10^{-6}$

version with numerical error reduction (M-NER) sensitivity stabilizes for $nb = 0.2$, and further increasing the value of this parameter is no longer beneficial.

Table 8 presents the results obtained for extended product invariants of Zernike moments (with and without numerical error reduction) and $p = 8$, $q = 8$, $r = 8$. For both versions of features, with and without numerical error reduction it can be seen that the sensitivity stabilizes for $nb = 0.2$.

Table 9. Detection results for moduli of Zernike moments and $p = 10$, $q = 10$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$		
M	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999995887	0.96180556	0	0.999995887	0.96180556	0
5	0.999947575	0.23263889	0	0.999995888	0.96180556	0
10	0.999936269	0.09722222	0	0.999985607	0.85763889	0
M-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999995888	1	$4.113\cdot10^{-6}$	0.999995888	1	$4.113\cdot10^{-6}$
5	0.999947576	0.23958333	$1.028\cdot10^{-6}$	0.999996916	0.97916667	$1.028\cdot10^{-6}$
10	0.999937297	0.11458333	$1.028\cdot10^{-6}$	0.999986635	0.875	$1.028\cdot10^{-6}$
fs	$nb = 0.3$			$nb = 0.4$		
M	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999995887	0.96180556	0	0.999995887	0.96180556	0
5	0.999995888	0.96180556	0	0.999995888	0.96180556	0
10	0.999985607	0.85763889	0	0.999985607	0.85763889	0
M-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999995888	1	$4.113\cdot10^{-6}$	0.999995888	1	$4.113\cdot10^{-6}$
5	0.999996916	0.97916667	$1.02\cdot10^{-6}8$	0.999996916	0.97916667	$1.028\cdot10^{-6}$
10	0.999986635	0.875	$1.028\cdot10^{-6}$	0.999986635	0.875	$1.028\cdot10^{-6}$

Table 9 presents the results obtained for moduli of Zernike moments (with and without numerical error reduction) and $p = 10$, $q = 10$, $r = 8$. For both versions of features, with and without numerical error reduction, it can be seen that the sensitivity stabilizes for $nb = 0.2$.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_5

Table 10 presents the results obtained for extended product invariants of Zernike moments (with and without numerical error reduction) and $p = 10$, $q = 10$, $r = 8$. As in previous cases, for both versions of features, with and without numerical error reduction, it can be seen that the sensitivity stabilizes for $nb = 0.2$.

Table 10. Detection results for extended product invariants of Zernike moments and $p = 10$, $q = 10$, $r = 8$.

fs	$nb = 0.1$			$nb = 0.2$		
E	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999995887	0.96180556	0	0.999995887	0.96180556	0
5	0.999956827	0.38888889	0	0.999995887	0.96180556	0
10	0.999941408	0.19097222	0	0.999985608	0.87563889	0
E-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.99999383	1	$6.17{\cdot}10^{-6}$	0.99999383	1	$6.17{\cdot}10^{-6}$
5	0.999956827	0.38541667	$1.028{\cdot}10^{-6}$	0.999996916	0.97916667	$1.028{\cdot}10^{-6}$
10	0.999942437	0.20933333	$1.028{\cdot}10^{-6}$	0.999986636	0.875	$1.028{\cdot}10^{-6}$
fs	$nb = 0.3$			$nb = 0.4$		
E	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.999995887	0.96180556	0	0.999995887	0.96180556	0
5	0.999995887	0.96180556	0	0.999995887	0.96180556	0
10	0.999985605	0.87563889	0	0.999985605	0.87563889	0
E-NER	accuracy	sensitivity	FAR	accuracy	sensitivity	FAR
1	0.99999383	1	$6.17{\cdot}10^{-6}$	0.99999383	1	$6.17{\cdot}10^{-6}$
5	0.999996916	0.97916667	$1.028{\cdot}10^{-6}$	0.999996916	0.97916667	$1.028{\cdot}10^{-6}$
10	0.999986636	0.875	$1.028{\cdot}10^{-6}$	0.999986636	0.875	$1.028{\cdot}10^{-6}$

5 Conclusion

We have proposed an algorithm that allows for faster rotationally-invariant object detection based on Zernike moments in comparison to standard detection procedure involving full scans by the sliding window. The presented technique takes advantage of frame skipping and is applicable only for video sequences. Although the detection time can be significantly reduced, it is fair to remark that the sensitivity measure can drop when number of skipped frames is set to a too large value. Of course, everything depends on the speed of movement, whether it's the camera or the object in the scene. Selecting a larger nb parameter provides more flexibility, but of course it takes more computation time.

References

1. Acasandrei, L., Barriga, A.: Embedded face detection application based on local binary patterns. In: 2014 IEEE Intl Conf on High Performance Computing and Communications (HPCC,CSS,ICESS). pp. 641–644 (2014)
2. Aly, S., sayed, A.: An effective human action recognition system based on zernike moment features. In: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE). pp. 52–57 (2019)
3. Bera, A., Klęsk, P., Sychel, D.: Constant-Time Calculation of Zernike Moments for Detection with Rotational Invariance. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(3), 537–551 (2019)

4. 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)
5. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Conference on Computer Vision and Pattern Recognition (CVPR'05) – Volume 1. pp. 886–893. IEEE Computer Society (2005)
6. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. The Annals of Statistics **28**(2), 337–407 (2000)
7. Jia, Z., Liao, S.: Leaf recognition using k-nearest neighbors algorithm with zernike moments. In: 2023 8th International Conference on Image, Vision and Computing (ICIVC). pp. 665–669 (2023)
8. Klęsk, P., Bera, A., Sychel, D.: Reduction of numerical errors in zernike invariants computed via complex-valued integral images. In: Computational Science – ICCS 2020. pp. 327–341. Springer International Publishing (2020)
9. Klęsk, P., Bera, A., Sychel, D.: Extended zernike invariants backed with complex-valued integral images for detection tasks. Procedia Computer Science **192**, 357–368 (2021)
10. Lai, W., Lei, G., Meng, Q., Shi, D., Cui, W., Wang, Y., Han, K.: Single-pixel detecting of rotating object using zernike illumination. Optics and Lasers in Engineering **172**, 107867 (2024)
11. Liang, X., Ma, W., Zhou, J., Kong, S.: Star identification algorithm based on image normalization and zernike moments. IEEE Access **8**, 29228–29237 (2020)
12. Malek, M.E., Azimifar, Z., Boostani, R.: Facial age estimation using Zernike moments and multi-layer perceptron. In: 22nd Int. Conference on Digital Signal Processing (DSP). pp. 1–5 (2017)
13. Mukundan, R., Ramakrishnan, K.: Moment Functions in Image Analysis — Theory and Applications. World Scientific (1998)
14. Rasolzadeh, B., et al.: Response Binning: Improved Weak Classifiers for Boosting. In: IEEE Intelligent Vehicles Symposium. pp. 344–349 (2006)
15. Singh, C., Upneja, R.: Accurate Computation of Orthogonal Fourier-Mellin Moments. Journal of Mathematical Imaging and Vision **44**(3), 411–431 (2012)
16. Vengurlekar, S.G., Jadhav, D., Shinde, S.: Object detection and tracking using zernike moment. In: 2019 International Conference on Communication and Electronics Systems (ICCES). pp. 12–17 (2019)
17. 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)
18. Wang, K., Wang, H., Wang, J.: Terrain matching by fusing hog with zernike moments. IEEE Transactions on Aerospace and Electronic Systems **56**(2), 1290–1300 (2020)
19. 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)
20. Zernike, F.: Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode. Physica **1**(8), 668–704 (1934)
21. Zheng, N., Zhang, G., Zhang, Y., Sheykhahmad, F.R.: Brain tumor diagnosis based on zernike moments and support vector machine optimized by chaotic arithmetic optimization algorithm. Biomedical Signal Processing and Control **82** (2023)
22. Zhou, Z., Liu, P., Chen, G., Liu, Y.: Moving object detection based on zernike moments. In: 2016 5th International Conference on Computer Science and Network Technology (ICCSNT). pp. 696–699 (2016)

DOI: 10.1007/978-3-031-63751-3_5