SN Computer Science (2024) 5:1166
https://doi.org/10.1007/s42979-024-03504-x

ORIGINAL RESEARCH

3D Non-separable Moment Invariants and Their Use in Neural Networks

Tomáš Karella[1] · Tomáš Suk[1] · Václav Košík[1] · Leonid Bedratyuk[2] · Tomáš Kerepecký[1] · Jan Flusser[1]

Received: 21 February 2024 / Accepted: 5 November 2024

Abstract
Recognition of 3D objects is an important task in many bio-medical and industrial applications. The recognition algorithms should work regardless of a particular orientation of the object in the space. In this paper, we introduce new 3D rotation moment invariants, which are composed of non-separable Appell moments. We show that non-separable moments may outperform the separable ones in terms of recognition power and robustness thanks to a better distribution of their zero surfaces over the image space. We test the numerical properties and discrimination power of the proposed invariants on three real datasets—MRI images of human brain, 3D scans of statues, and confocal microscope images of worms. We show the robustness to resampling errors improved more than twice and the recognition rate increased by 2–10 % comparing to most common descriptors. In the last section, we show how these invariants can be used in state-of-the-art neural networks for image recognition. The proposed H-NeXtA architecture improved the recognition rate by 2–5 % over the current networks.

Keywords 3D rotation invariants · Non-separable moments · Appell polynomials · Convolutional neural networks

Introduction

Robust recognition of 3D objects is particularly important in bio-medical imaging, where modalities such as CT, MRI, and confocal microscopes yield full 3D volumetric data, as well as in numerous industrial applications. Two main approaches to this problem are via "handcrafted" and "learned" features. While in 2D the convolutional networks and deep learned features have almost completely replaced traditional handcrafted features, the situation in 3D recognition is not so clear-cut.

For volumetric data, there are several 2D-inspired architectures operating on voxels such as convolution networks [1–3], residual networks [4], U-Net [5, 6], generative models [7–11], and transformers [12–15]. However, one faces many practical problems when applying neural networks to 3D data. The data size and dimension imply the demand of large-scale annotated training sets. Such public datasets do not exist, unlike for instance ImageNet, that serves as a universal training set in 2D applications. We can find only few specialized benchmarks for narrow areas like Kitty (dataset for autonomous driving) [16] and fastMRI [17] containing knee and brain MRI snaps. These training data can be used in specific areas, but do not have a potential of pre-training general backbones suitable for transfer learning. The problem of geometric invariance of the network, widely investigated in 2D [18], has been studied in a few very recent papers [19, 20]. These shortcomings give way to traditional methods with low-demand training and effective descriptions resistant to deformations such as rotation, scale or translation. However, transformation robust 3D neural networks are already starting to appear for specific tasks, for

✉ Tomáš Karella
karella@utia.cas.cz

Tomáš Suk
suk@utia.cas.cz

Václav Košík
kosik@utia.cas.cz

Leonid Bedratyuk
leonid.uk@gmail.com

Tomáš Kerepecký
kerepecky@utia.cas.cz

Jan Flusser
flusser@utia.cas.cz

[1] Institute of Information Theory and Automation, Czech Academy of Sciences, Pod Vodárenskou věží 4, 182 08 Prague, Czech Republic

[2] Khmelnytsky National University, Instytuts'ka 11, Khmelnytsky 29016, Ukraine

Published online: 12 December 2024

example for the recently published equivariant networks for object detection [19, 20].

So, there is still a clear demand to develop efficient handcrafted invariant features that can be used standalone outside neural network framework, but can be also incorporated into state-of-the-art hybrid network architectures to improve recognition of deformed objects while avoiding massive augmentation.

Among many possible choices, *moment invariants* were proven to be very powerful descriptors of 3D bodies, because they provide invariance to the object pose and scale [21]. 3D moment invariants have been studied much less than their 2D counterparts, which means there are still many open questions concerning namely numerical stability and ability to represent objects by low-dimensional vectors. Both these issues are connected with the orthogonality of the moments (more precisely, with the orthogonality of the corresponding polynomial bases). Orthogonal (OG) moments provide generally better representation, stability and discrimination power than non-orthogonal ones. On the other hand, rotation invariants from OG moments are generally more difficult to construct than those from standard non-orthogonal moments [22, 23]. Two families of popular 3D rotation moment invariants composed of OG moments are those based on Zernike moments [24] and Gaussian-Hermite moments [25].

Both these systems (and actually all other ones that have been used in object recognition so far) are *separable*, which means their basis functions can be factorized as $\pi_{pqr}(x, y, z) = P_p(x)P_q(y)P_r(z)$. Zernike moments are separable in polar domain, Gaussian-Hermite moments are separable in cartesian domain. Separability is convenient from computational point of view, but results in certain limitations of the representation ability. The distribution of zeros of separable functions is constrained such that the zero surfaces fill a rectangular or polar grid (see Fig. 1). Hence, separable basis functions provide good representation in the grid directions, while the representation in "diagonal" directions may be worse. It may lead to the drop of discriminability, if characteristic object structures exhibit a diagonal-like orientation and / or if we employ only a few low-order basis functions. This has led recently to introducing *non-separable* bases, however so far in 2D only.

Bedratyuk et al. [26] introduced 2D non-separable Appell moment invariants. In this paper, we generalize their idea into 3D and we demonstrate their usage as standalone descriptors and we also show how they can be incorporated into novel network architectures.

The paper is organized as follows. The basic idea is outlined in "Basic Idea Behind 3D Invariants". "3D Appell Polynomials and Moments" introduces the Appell polynomials in 3D and their use for the design of rotation invariants. and "Experiments with Appell Invariants as Standalone Features" describes numerical experiments on real data of various kind. "Appell Moments in Convolutional Networks" deals with the possible usage of the Appell moments in

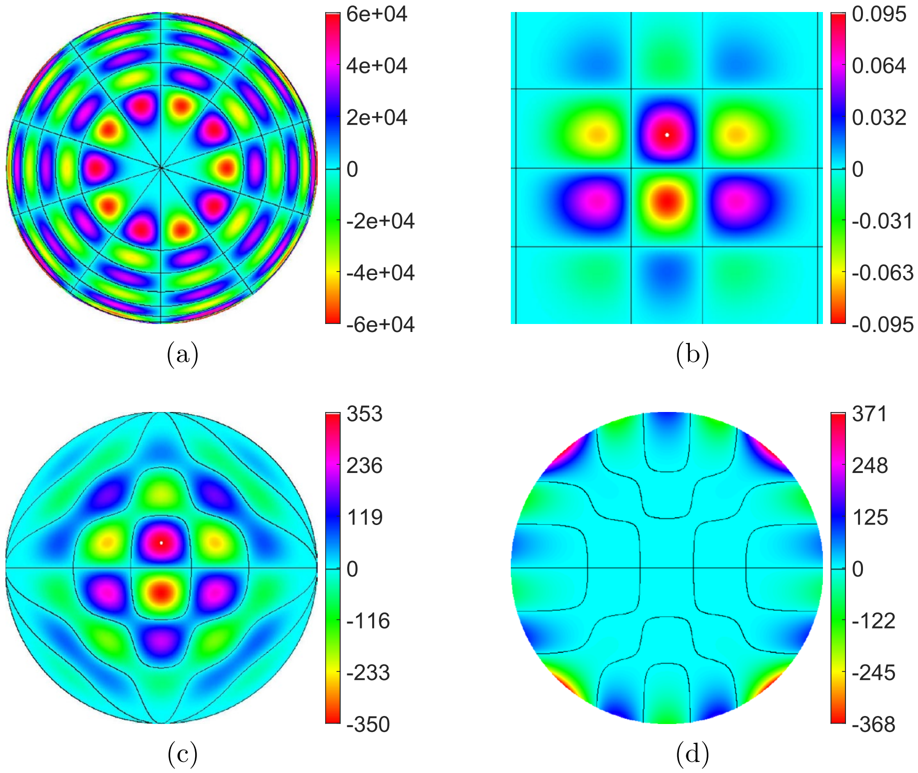


Fig. 1 Slices of 3D polynomials showing the zero distribution: **a** separable Zernike $\Re\left(Z_{15,9}^{5}\right)$, *xy* plane, **b** separable Gaussian–Hermite G_{456}, *xy* plane, **c** non-separable Appell U_{456}, *xy* plane, **d** non-separable Appell V_{456}, *xy* plane. The black curves are the zero sets

convolutional neural networks and “Conclusion” concludes the paper.

Basic Idea Behind 3D Invariants

To design 3D rotation invariants as functions of non-separable moments, we need to find basis polynomials that are *quasi-monomials*, are not separable, and there exists a stable and fast algorithm for their evaluation. Quasi-monomials are polynomials, that are transformed under coordinate rotation exactly in the same way as monomials $x^p y^q z^r$ [27]. This property is crucial for the invariant design. We can simply substitute the quasi-monomial moments into well-known invariants of geometric moments (i.e. moments w.r.t. the monomial basis) [21]. There is no need of designing invariant “from scratch”. However, quasi-monomials are rare. Among all separable polynomials, Hermite polynomials were proved to be the only quasi-monomials [28]. Among non-separable polynomials, there is no such necessary and sufficient condition. Bedratyuk et al. [26] proved that Appell polynomials [29] are quasi-monomials in 2D. As is shown below, this key property is preserved in 3D as well. In the next section, we present 3D Appell polynomials, Appell moments and original recurrent relations for their efficient computation.

3D Appell Polynomials and Moments

The term *Appell polynomials* (APs, named after Paul Émile Appell, a French mathematician) denotes two families of multivariate non-separable polynomials U and V. Appell polynomials are *bi-orthogonal*, which means any two polynomials, one being from U and the other one from V, are orthogonal (with a weight) on a unit sphere.

The definition of 3D Appell polynomials via expansion into standard powers is the following (for more details on the APs see [29]).

$$\begin{aligned} U_{m,n,o}(x,y,z) &= (m+n+o)! \sum_{i=0}^{[m/2]} \sum_{j=0}^{[n/2]} \sum_{k=0}^{[o/2]} \frac{(-1)^{i+j+k}(-m)_{2i}(-n)_{2j}(-o)_{2k}}{4^{i+j+k}\, i!\, j!\, k!\, (i+j+k)!} \\ &\quad \cdot x^{m-2i} y^{n-2j} z^{o-2k} (1-x^2-y^2-z^2)^{i+j+k} \\ V_{m,n,o}(x,y,z) &= 2^{m+n+o} \sum_{i=0}^{[m/2]} \sum_{j=0}^{[n/2]} \sum_{k=0}^{[o/2]} \binom{m}{i}\binom{n}{j}\binom{o}{k} \frac{(\frac{3}{2})_{m+n+o-i-j-k}(i-m)_i(j-n)_j(k-o)_k}{4^{i+j+k}} \\ &\quad \cdot x^{m-2i} y^{n-2j} z^{o-2k}, \end{aligned} \tag{1}$$

where $(a)_k = a(a+1)(a+2)\cdots(a+k-1)$ is Pochhammer symbol.

An equivalent definition by means of generating functions is given as

$$\begin{aligned} &\frac{1}{\left[(1-(ux+vy+wz))^2-(u^2+v^2+w^2)(x^2+y^2+z^2-1)\right]^{\frac{1}{2}}} \\ &\quad = \sum_{m,n,o=0}^{\infty} U_{m,n,o}(x,y,z)\,\frac{u^m}{m!}\,\frac{v^n}{n!}\,\frac{w^o}{o!}, \\ &\frac{1}{1-2(ux+vy+wz)+u^2+v^2+w^2} = \sum_{m,n,o=0}^{\infty} V_{m,n,o}(x,y,z)\,\frac{u^m}{m!}\,\frac{v^n}{n!}\,\frac{w^o}{o!}. \end{aligned} \tag{2}$$

Appell polynomials U and V from (1) or (2) are bi-orthogonal on the unit sphere $B = \{(x,y,z) \mid x^2+y^2+z^2 \le 1\}$. The relation of bi-orthogonality is

$$\begin{aligned} &\iiint_B U_{p,q,r}(x,y,z)V_{m,n,o}(x,y,z)W^{(s)}(x,y,z)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\ &\quad = d^2_{m,n,o}\,\delta_{mp}\,\delta_{nq}\,\delta_{or}, \end{aligned} \tag{3}$$

where δ_{ij} is the Kronecker delta function, the weight function is

$$W^{(s)}(x,y,z) = (1-x^2-y^2-z^2)^{s-1} \tag{4}$$

and the normalizing constant is

$$d_{m,n,o} = \frac{\pi^{\frac{3}{2}}\Gamma(s)(s+\frac{1}{2})(2s-1)_{m+n+o}}{\Gamma(s+\frac{3}{2})(m+n+o+s+\frac{1}{2})m!n!o!}. \tag{5}$$

Note that $W^{(s)}(x,y,z) = 1$ for $s = 1$. However, neither of the above definitions is convenient for numerical evaluation due to possible overflows. In Appendix B, we present recurrent formulas for stable and fast computation.

The generating functions play a crucial role in the following theorem.

Theorem 1 *Let us suppose that the polynomial family* $\{B_{m,n,o}(x,y,z);\, m,n,o \in \mathbb{N}_0\}$ *is defined by a generating function*

$$G(x,y,z,u,v,w) = \sum_{m,n,o=0}^{\infty} B_{m,n,o}(x,y,z)\, \frac{u^m}{m!}\, \frac{v^n}{n!}\, \frac{w^o}{o!}.$$

Then all $B_{m,n,o}$ *are quasi-monomials if and only if* G *is a function of* $ux + vy + wz,\ x^2 + y^2 + z^2$ *and* $u^2 + v^2 + w^2$ *only.*

For the proof of Theorem 1 see Appendix A. Now we can easily see that Appell polynomials are quasi-monomials, because their generating functions satisfy Theorem 1.

The Appell moments M of a 3D image $f(x, y, z)$ are its projections onto the set of Appell polynomials

$$M_{pqr}^{(P)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P_{pqr}(x,y,z) f(x,y,z)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z, \tag{6}$$

where P stands either for U or for V. To obtain Appell invariants, these moments are substituted directly into geometric moment invariants [21, 30] (this is possible because APs were proven to be quasi-monomials), so we end up with formulas such as (the superscript (P) in $M_{pqr}^{(P)}$ is omitted)

$$\begin{aligned}
\Phi_1 &= M_{200} + M_{020} + M_{002},\\
\Phi_2 &= M_{200}^2 + 2M_{110}^2 + 2M_{101}^2 + M_{020}^2 + 2M_{011}^2 + M_{002}^2,\\
\Phi_3 &= M_{200}^3 + 3M_{200}M_{110}^2 + 3M_{200}M_{101}^2 + 3M_{110}^2M_{020} + 6M_{110}M_{101}M_{011}\\
&\quad + 3M_{101}^2M_{002} + M_{020}^3 + 3M_{020}M_{011}^2 + 3M_{011}^2M_{002} + M_{002}^3,\\
\Phi_4 &= M_{300}^2 + 3M_{210}^2 + 3M_{201}^2 + 3M_{120}^2 + 6M_{111}^2 + 3M_{102}^2 + M_{030}^2 + 3M_{021}^2\\
&\quad + 3M_{012}^2 + M_{003}^2.
\end{aligned} \tag{7}$$

Using the list of invariants from [30], we obtain a complete and independent set of 213 invariants up to the 9th moment order.

In this paper, we calculate the moments and the invariants globally from the entire object. Local application, for instance to small overlapping blocks as proposed in [31] is also possible and might be used for objects that are not completely visible. However, the efficient computational tricks based on matrix factorization [31] cannot be applied because Appell moments are not separable. Local application of Appell invariants is beyond the scope of this paper.

Experiments with Appell Invariants as Standalone Features

We present recognition experiments on three different data collections. In all these experiments, the 3D Appell invariants were used as traditional handcrafted features, which means no neural networks were employed and a simple nearest neighbor rule was used for classification.

Human Brain MRI

The aim of the first experiment is to numerically verify the rotation invariance. We used two MRI measurements of the brain of the same patient (Fig. 2) downloaded from [32]. Their original sizes are 192 × 224 × 224 and 193 × 229 × 193 voxels. We generated 8 random 3D rotations of each snap with bilinear interpolation and then computed 77 rotation

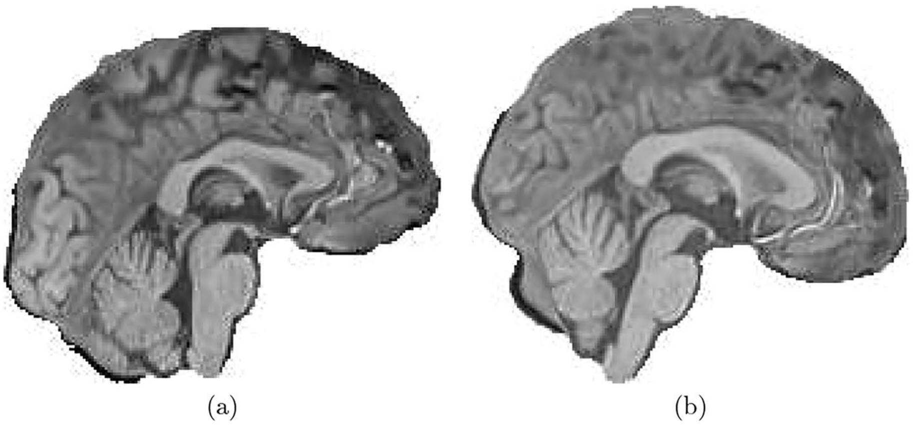


Fig. 2 Brain MRI images used in the experiment: **a** slice 96 (out of 192) of the first snap, **b** slice 97 (out of 193) of the second snap

Table 1 ERAs of the rotation invariants of the brains in %

Invariants	Appell U	Appell V	Complex	Geometric	G-H	Zernike
Brain 1	1.2067	0.9720	2.6408	2.6392	3.4373	1.4609
Brain 2	1.4592	1.1898	3.5169	3.5168	3.8445	1.8552
Average	1.3329	1.0809	3.0788	3.0780	3.6409	1.6580

The averages over all invariants are used

invariants up to the sixth order. We computed the Appell moment invariants both of U and V families by recurrence formulas (19)–(24) and compared them with the invariants from complex moments [23], geometric moments [22], Gaussian–Hermite moments [25] and Zernike moments [24].

As a measure of quality we used the error relative to average (ERA)

$$ERA = \frac{100\%}{n_i} \sum_{j=1}^{n_i} \frac{\frac{1}{n_r} \sum_{i=1}^{n_r} \left| I_j^i - \frac{1}{n_r} \sum_{i=1}^{n_r} I_j^i \right|}{\frac{1}{n_i n_r} \sum_{i=1}^{n_r} \sum_{j=1}^{n_i} \left| I_j^i \right|}, \qquad (8)$$

where n_i is the number of invariants ($n_i = 77$ for sixth order), $n_r = 8$ is the number of rotations, and I_j^i is jth invariant of ith rotation. ERA is similar to more common mean relative error (MRE), which is, however, unstable for invariants being close to zero. The average ERAs of all invariants are shown in Table 1. It is apparent that both Appell U and V invariants actually exhibit the rotation invariance, even with smaller error than traditional separable invariants. The ERAs of the individual invariants are in Fig. 3. The invariants sorted by order are on the horizontal axis.

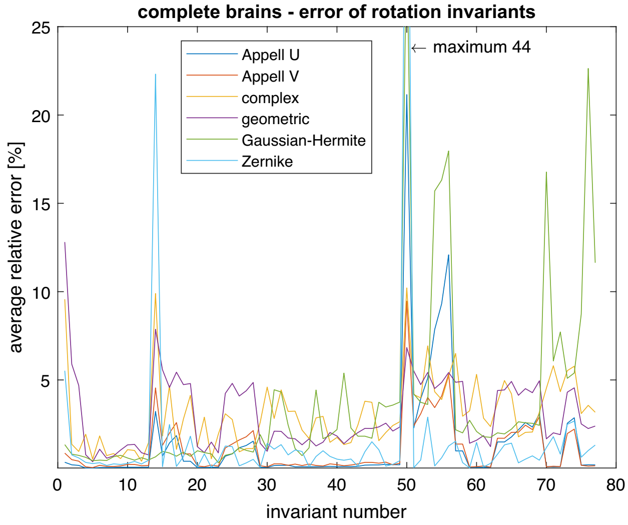


Fig. 3 ERAs of the rotation invariants—whole brains. Horizontal axis contains the labels of the invariants

The Statues

This experiment demonstrates the ability of the Appell invariants in a simple object recognition task. We scanned five visually similar small sculptures by a 3D scanner. It is based on projection of a special moving pattern on the scanned object and capturing it by a camera. The object lays on a rotating table enabling 8 scans from 8 different directions by 45°. The accessory software then creates the 3D model from the 8 scans as the triangulated surface, (see Fig. 4a–e for the models). We use neither texture on the surface nor any structure inside.

The original models were used as the training samples. Eight random rotations of each statue were classified by the same invariants that were used in the MRI experiment. We applied a simple nearest-neighbor classifier in the space of invariants. If there is no noise, all methods classified all statues correctly. To make the problem more challenging, we added random noise inside the circumscribed sphere around each test sample (see Fig. 4f for an example), that simulates scanner errors in recovering 3D surface. Noisy objects are more difficult to recognize and performance differences of individual methods become apparent, as is documented in Table 2.

We can see that the Appell U moments are the best performing ones, the only unsatisfactory result is for low order of the moments. Looking at the other results, it is interesting that good recognition rate does not necessarily correspond with low ERA value (compare complex and geometric invariants).

The Worms

In this experiment, we tested recognition via template matching. We used 3D data from confocal microscope that are publicly available [33]. The dataset was captured by Leica microscope with 63× oil objective [34] and consists of 28 volumes of worms *Caenorhabditis elegans* at the larval stage[1] and corresponding stacks of 555 ground-truth annotated cell nuclei, see Fig. 5.

[1] The dimension of the chosen volume is $1244 \times 140 \times 140$, the pixel size is $0.122 \times 0.116 \times 0.116$ µm

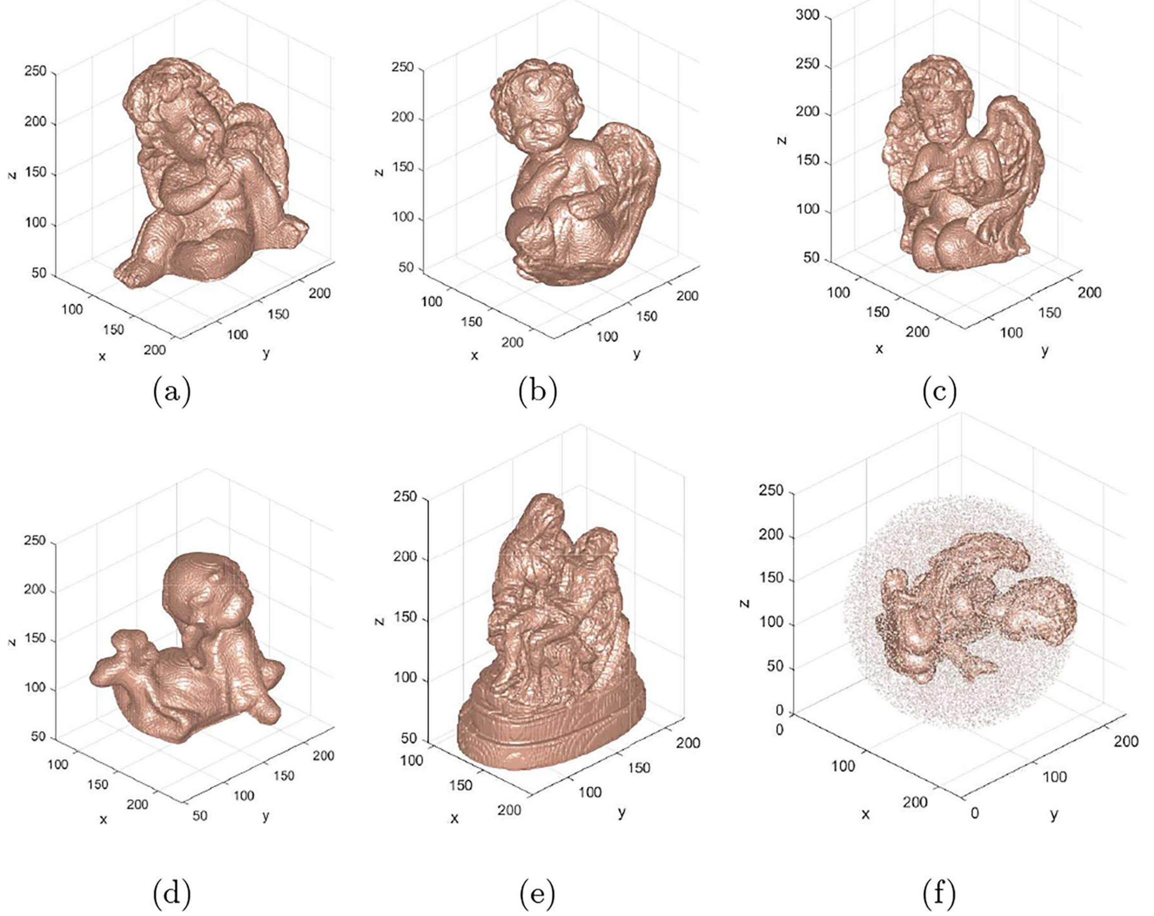


Fig. 4 Statues used in the experiment: **a** small angel, **b** medium-sized angel, **c** big angel, **d** lying angel, **e** pieta, and **f** rotated and noisy sample of the big angel to be recognized

Table 2 Success rates and relative errors of various rotation invariants of the statues in % for noisy objects

Max. order	Appell U	Appell V	Complex	Geometric	G-H	Zernike
2	60	62.2	100	60	93.3	95.6
3	100	91.1	97.8	100	100	100
4	100	100	97.8	100	100	95.6
5	100	88.9	97.8	97.8	80	95.6
6	100	93.3	97.8	100	86.7	95.6
ERA	0.246	0.303	2.675	0.324	2.744	2.506

The first column shows the maximum order of the moments used

Now we tried to detect the nuclei via template matching. Ten nuclei were chosen for training, i.e. we computed their invariants of all kinds up to the sixth order. Then we passed through the scan of the worm, computed invariants in the neighborhood of each voxel and compared them with the invariants of the training set. There is a hypothesis that the nuclei of different cells are very similar in their shape and appearance, but differ from one another by orientation in 3D space, so rotation invariance of the features is required. We optimized the radius of the spherical neighborhood for each type of moments individually to get the best performance (the optimal radius depends on the shape of the basis functions, so it cannot be the same in all cases).

The voxel is considered to be the center of the nucleus if the two following conditions are satisfied:

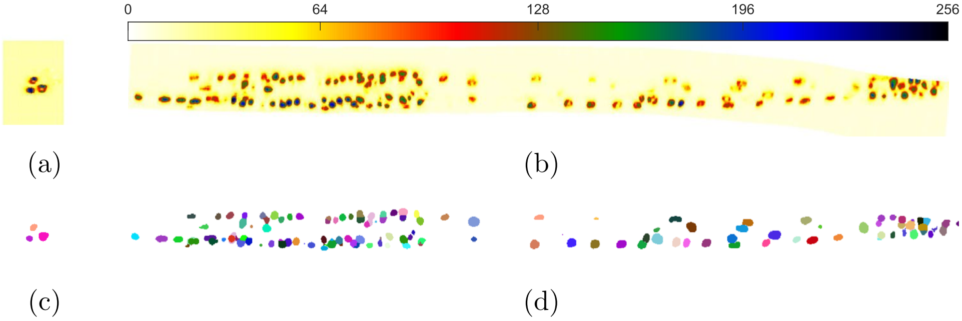


Fig. 5 The worm used in the experiment: **a** cross-section, **b** longitudinal section, **c** ground-truth nucleus masks in the cross-section, **d** ground-truth nucleus masks in the longitudinal section

- The feature distance must be below a user-defined threshold and must form the local minimum in the $3 \times 3 \times 3$ neighborhood of the voxel in question.
- The detected nucleus cannot overlap the nuclei detected before.

The quality of the detection was evaluated by means of the ground-truth masks. If the spatial distance between the detected nucleus and the nearest mask is less than 10 voxels, the detection is considered correct.

The results are summarized in Table 3. Again, Appell U invariants detected almost all nuclei and won the contest, followed by Complex, Geometric, and Zernike invariants.

Due to the high computation demand of a pattern matching problem, the source code was implemented in PyTorch framework allowing us to run the algorithm in parallel on Nvidia A100 GPU. Thanks to this, the task run by several orders faster than in case of traditional implementation, but still it took about two hours due to a large number of template positions to be tested. A speed up via pyramidal search and / or sparse space sampling would definitely be possible, but the runtime was not the issue we were primarily interested in. Therefore, the invariant calculation in each voxel took about two hours using Nvidia A100 GPU. The source codes are available at `https://github.com/karellat/nuclei`.

Appell Moments in Convolutional Networks

Convolutional neural networks (CNN) have attracted a noticeable attention of image processing community only since 2012, when a CNN named AlexNet won the ImageNet Large Scale Visual Recognition Challenge [35]. Soon the recognition rate have surpassed the human performance [36] thanks to a substantial increase of the computer performance at that time.

After a dynamic development in 2012-20, when many successful applications were reported, some limitations of the CNNs became apparent. One of the most significant ones is the non-invariance of CNNs even to very simple transformations such as shift and rotation of the images. Since these transformations are very often involved into intra-class variations, the non-invariance substantially decreases the recognition power of the network. A model that was optimized without these transformations in the training set is unable to process the changed features correctly and can be producing even random results.

Traditional CNNs handle this problem by the *augmentation* of the training set [37], which is in fact a brute-force approach, where we first artificially generate many transformations of training samples and the CNN is trained on this augmented set. This is an extremely time and memory consuming process that still does not guarantee the same network performance as on data without deformations [38]. Also, as shown by Zeiler and Fergus [39], the models trained with augmentation contain redundant filters that are rotated, scaled, and translated copies of each other.

The geometric non-invariance of classic CNNs has been a widely studied problem in the last few years. We refer to a survey paper [40], where the reader can find over 200 references to various approaches how to make CNN invariant to geometric transformations. Many of them incorporate various handcrafted features into the network [41–49]. Appell invariants can be used in these methods as well.

In this section, we show how Appell invariants can be incorporated into so-called *group equivariant CNN* (G-CNN).

Table 3 The numbers of correctly detected worm cell nuclei out of 545 instances

Invariants	Appell U	Appell V	Complex	Geometric	G-H	Zernike
# Detected nuclei	528	359	473	437	338	414
Radius [voxels]	13	11	11	13	15	17

Group Equivariant Networks

Cohen et al. introduced G-CNN [50], a general idea applied to 90 rotations and mirror reflections, where the main principle was rotating and mirroring convolutional filters. The work then inspired many other authors, similar ideas are used in [51, 52]. A very promising and widely studied branch of equivariant networks came with applying steerable filters, introduced in [53]. The first steerable CNNs were introduced by Cohen and Welling [54] and Worrall et al. [55] and they were followed by [56, 57], where the authors used complex circular harmonics for constructing steerable convolutional filters. All these works get rid of the equivariant term after the last equivariant convolutional layer to be invariant at the output. This can be achieved by a variety of poolings like standard Global Average Pooling, Max Pooling or even Zernike moments [58] and Polar Harmonic Transforms [59]. Most recently, Karella et al. [60] further improved H-Net [55]. Their H-NeXt is a modular invariant architecture with an equivariant backbone followed by invariant pooling. For the pooling part, they tested Global Average Pooling, Zernike moments, and also Multi-Head Self-Attention Pooling. This can be generalized to any other roto-translation invariant pooling, such as Appell moments.

H-NeXtA Network

Here we demonstrate that invariant pooling by Zernike moments can be in the H-NeXt architecture replaced by Appell moments. We call the new network H-NeXtA (see Figs. 6, 7 for visual explanation).

Due to lack of 3D data, we tested the performance of H-NeXtA (with Appell invariants simplified to 2D) on rotated MNIST dataset [61], which is a common image benchmark. It consists of 62,000 handwritten digits all randomly rotated. Only the 10,000 images were used for training, while 2000 were used for validation and 50,000 were used for testing. (Validation dataset is a subset reserved during model training to estimate model performance while tuning hyperparameters. The test dataset is used solely to provide an unbiased evaluation of the final model's performance.)

As you can see in Fig. 8, H-NeXtA outperforms not only the classical CNN, which is inherently not rotation invariant, but also the same CNN trained on extensively augmented set (each training image was randomly rotated at each epoch). Since the confidence intervals are disjoint, the differences of the performance are statistically significant.

Table 4 also compares the complexity of the networks. H-NeXtA has six times fewer parameters than the CNN, showing that the invariant networks use parameters more efficiently than the classical CNN. As an example of CNN

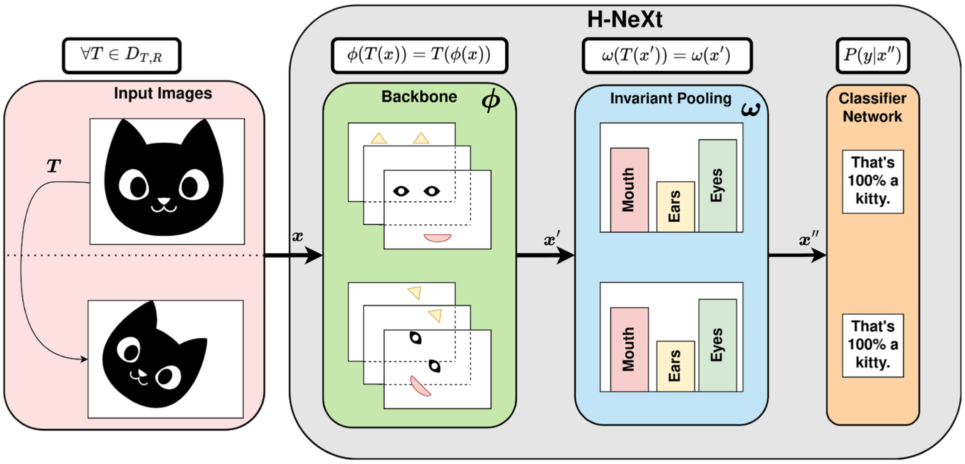


Fig. 6 H-NeXt / H-NeXtA network invariant with respect to the roto-translation ($\boldsymbol{T} \in D_{T,R}$) consisting of three blocks: equivariant backbone $\boldsymbol{\phi}$, invariant pooling $\boldsymbol{\omega}$ (using Zernike or Appell moments), and classifier network. The equivariant backbone $\boldsymbol{\phi}$ is commutative with the roto-translation. The output of invariant pooling $\boldsymbol{\omega}$ is the same regardless of the roto-translation of an input

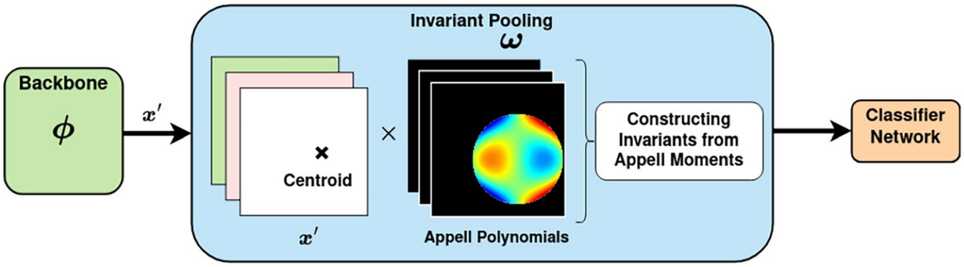
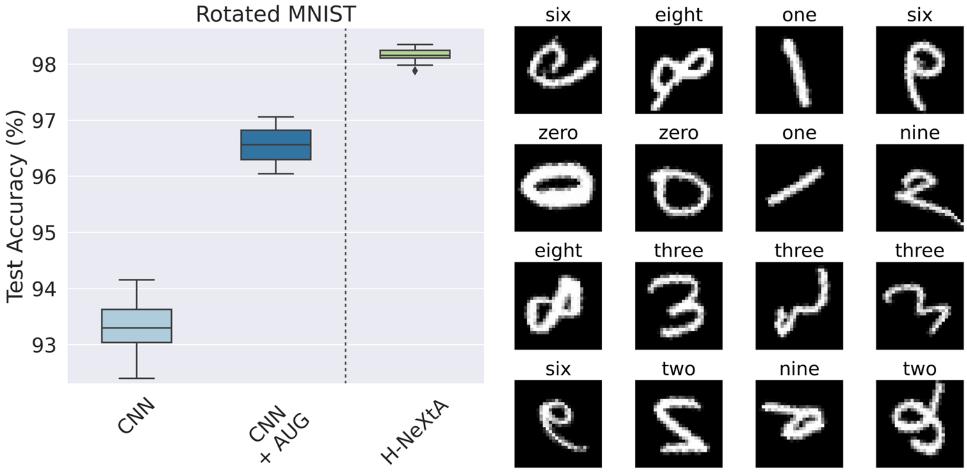


Fig. 7 Invariant pooling $\boldsymbol{\omega}$ by constructing Appell invariants from backbone feature maps. Channels are translated to have a center of gravity in the middle. Aligned channels are multiplied by Appell polynomials and finally Appell invariants are generated

Fig. 8 **a** Recognition rate of H-NeXtA on rotated MNIST dataset [61]. **b** Examples from the rotated MNIST dataset [61]. The goal of this benchmark is to classify rotated handwriting digits. The training set has only 10,000 images, and all digits, including the test, training, and validation sets, are rotated. This is in contrast to traditional MNIST [62], where all digits are upright and the training set has 50,000 examples



(a)

(b)

architecture, we choose an extended version of LeNet [62]. All classical CNNs perform similarly, including small ResNets [63], AlexNet [37] and VGG [64]. For the hyperparameters and other details of the experiment see Appendix C.

Conclusion

We introduced new 3D rotation moment invariants, which are composed of non-separable Appell moments. To the best of our knowledge, this is the first application of 3D non-separable polynomials in object recognition. The design of the invariants was possible because the Appell polynomials are quasi-monomials. At this moment, we are not aware of any other non-separable quasi-monomials. Furthermore, we proposed recursive formulae for fast and stable computation.

To show the performance of the new Appell invariants in practice, we presented three experiments of different kind – invariance verification on MRI scans, object recognition of real 3D objects, and template matching in a volumetric microscopic images. In all of them, Appell invariants outperformed the competitors. This is mainly due to more even distribution of zeros of the Appell polynomials over the image space, which leads to a better representation ability of the Appell moments, especially if only low-order features are used.

At the end, we demonstrated the possibility of incorporating Appell invariants into state-of-the-art network architectures that use both learned and handcrafted features in order to reduce training data and increase the performance on rotated images.

Appendix 1: Proof of Theorem 1

- ($\Rightarrow$): Let $\{B_{m,n,o}(x,y,z)\}$ be a quasi-monomial family. First, consider a coordinate rotation around the z-axis by θ.

$$\begin{aligned} &B_{m,n,o}(x',y',z)\\ &\quad= B_{m,n,o}(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta, z)\\ &\quad= \sum_{i=0}^{m}\sum_{j=0}^{n}(-1)^i\binom{m}{i}\binom{n}{j}\\ &\quad\times(\cos\theta)^{m-i+j}(\sin\theta)^{n-j+i}B_{m+n-i-j,i+j,o}(x,y,z). \end{aligned} \tag{9}$$

Let us derive the equation by θ and assign $\theta = 0$ to the derivative. We get

$$\begin{aligned} &x\frac{\partial B_{m,n,o}(x,y,z)}{\partial y} - y\frac{\partial B_{m,n,o}(x,y,z)}{\partial x}\\ &\quad= nB_{m+1,n-1,o}(x,y,z) - mB_{m-1,n+1,o}(x,y,z) \end{aligned} \tag{10}$$

for all $m,n,o \in \mathbb{N}_0$ if we set $B_{-1,n,o} = B_{m,-1,o} = 0$ for any $m,n,o \in \mathbb{N}_0$. Thus, the generative function obeys

$$\begin{aligned}
x\frac{\partial G}{\partial y} - y\frac{\partial G}{\partial x} &= \sum_{m,n,o=0}^{\infty}\left(x\frac{\partial B_{m,n,o}(x,y,z)}{\partial y} - y\frac{\partial B_{m,n,o}(x,y,z)}{\partial x}\right)\frac{u^m}{m!}\frac{v^n}{n!}\frac{w^o}{o!}\\
&= \sum_{m,n,o=0}^{\infty}\left(nB_{m+1,n-1,o}(x,y,z) - mB_{m-1,n+1,o}(x,y,z)\right)\frac{u^m}{m!}\frac{v^n}{n!}\frac{w^o}{o!}\\
&= v\sum_{m,o=0,n=1}^{\infty} B_{m+1,n-1,o}(x,y,z)\frac{u^m}{m!}\frac{v^{n-1}}{(n-1)!}\frac{w^o}{o!}\\
&\quad - u\sum_{n,o=0,m=1}^{\infty} B_{m-1,n+1,o}(x,y,z)\frac{u^{m-1}}{(m-1)!}\frac{v^n}{n!}\frac{w^o}{o!}\\
&= v\sum_{m,n,o=0}^{\infty} B_{m+1,n,o}(x,y,z)\frac{u^m}{m!}\frac{v^n}{n!}\frac{w^o}{o!}\\
&\quad - u\sum_{m,n,o=0}^{\infty} B_{m,n+1,o}(x,y,z)\frac{u^m}{m!}\frac{v^n}{n!}\frac{w^o}{o!}\\
&= v\frac{\partial G}{\partial u} - u\frac{\partial G}{\partial v}. \qquad (11)
\end{aligned}$$

Doing the same trick with the rotation around the y-axis and the x-axis, we get the following system of three differential equations

$$\begin{aligned}
x\frac{\partial G}{\partial y} - y\frac{\partial G}{\partial x} &= v\frac{\partial G}{\partial u} - u\frac{\partial G}{\partial v}\\
x\frac{\partial G}{\partial z} - z\frac{\partial G}{\partial x} &= w\frac{\partial G}{\partial u} - u\frac{\partial G}{\partial w} \qquad (12)\\
z\frac{\partial G}{\partial y} - y\frac{\partial G}{\partial z} &= v\frac{\partial G}{\partial w} - w\frac{\partial G}{\partial v}.
\end{aligned}$$

The coefficient matrix of the system has the rank 3 which determines the number of independent solutions by $6 - 3 = 3$. Obviously, $x^2 + y^2 + z^2$, $u^2 + v^2 + w^2$ and $ux + vy + wz$ are solutions and they are independent. Hence, G is a function of $x^2 + y^2 + z^2$, $u^2 + v^2 + w^2$ and $ux + vy + wz$ only.

- ($\Leftarrow$): A rotation $\mathcal{R}$ in $\mathbb{R}^3$ can be decomposed as $\mathcal{R} = \mathcal{R}_x(\gamma)\mathcal{R}_y(\beta)\mathcal{R}_z(\alpha)$ where $\mathcal{R}_x, \mathcal{R}_y, \mathcal{R}_z$ are rotations around x-axis, y-axis and z-axis and the argument is the angle of the rotation. First, we prove that a rotation of $B_{m,n,o}$ around the z-axis behaves exactly like the rotation of the monomial $x^m y^n z^o$. The rotation around z-axis transforms the coordinates as

$$\begin{aligned}
x' &= x\cos\alpha - y\sin\alpha,\\
y' &= x\sin\alpha + y\cos\alpha, \qquad (13)\\
z' &= z.
\end{aligned}$$

Therefore, the monomial $x^m y^n z^o$ is transformed as

$$\begin{aligned}
&(x')^m(y')^n(z')^o =\\
&= \sum_{i=0}^{m}\sum_{j=0}^{n}(-1)^i\binom{m}{i}\binom{n}{j}(\cos\alpha)^{m-i+j}(\sin\alpha)^{n-j+i}x^{m+n-i-j}y^{i+j}z^o. \qquad (14)
\end{aligned}$$

If we set

$$\begin{aligned}
\overline{u} &:= u\cos\alpha + v\sin\alpha, \qquad (15)\\
\overline{v} &:= v\cos\alpha - u\sin\alpha,
\end{aligned}$$

we can easily verify the following equalities

$$\begin{aligned}
ux' + vy' + wz' &= \overline{u}x + \overline{v}y + wz,\\
x'^2 + y'^2 + z'^2 &= x^2 + y^2 + z^2, \qquad (16)\\
\overline{u}^2 + \overline{v}^2 + w^2 &= u^2 + v^2 + w^2.
\end{aligned}$$

Therefore, the generating function obeys $G(x', y', z', u, v, w) = G(x, y, z, \overline{u}, \overline{v}, w)$. It follows that

$$\begin{aligned}
&\sum_{m,n,o=0}^{\infty} B_{m,n,o}(x', y', z')\frac{u^m}{m!}\frac{v^n}{n!}\frac{w^o}{o!}\\
&= \sum_{m,n,o=0}^{\infty} B_{m,n,o}(x,y,z)\frac{(u\cos\alpha + v\sin\alpha)^m}{m!}\cdot \qquad (17)\\
&\qquad \cdot\frac{(v\cos\alpha - u\sin\alpha)^n}{n!}\cdot\frac{w^o}{o!}.
\end{aligned}$$

By matching the coefficients of the same powers of u, v and w, we get

$$\begin{aligned} B_{m,n,o}(x',y',z') = \sum_{i=0}^{m}\sum_{j=0}^{n}(-1)^i \binom{m}{i}\binom{n}{j} \\ (\cos\alpha)^{m-i+j}(\sin\alpha)^{n-j+i} \cdot \\ \cdot B_{m+n-i-j,i+j,o}(x,y,z), \end{aligned} \tag{18}$$

which corresponds to the rotation of monomials in (14). It is easy to see that the rotations around the y-axis and then the x-axis will again behave the same. This observation completes the proof. □

Appendix 2: Recurrence Formulas

In this Appendix, we present recurrent relations for fast and stable computation of 3D Appell polynomials. The polynomials $U_{m,n,o} = U_{m,n,o}(x,y,z)$ satisfy the recurrences

$$\begin{aligned} U_{m+1,n,o} &= x(2m+n+o+1)U_{m,n,o} + moxzU_{m,n,o-1} + mnxyU_{m,n-1,o} \\ &+ 2mnoxyzU_{m,n-1,o-1} + m((y^2+z^2-1)m + (y^2+2z^2-1)o \\ &+ (2y^2+z^2-1)n)U_{m-1,n,o} + moz((y^2-1)(m+o-1) \\ &+ (3y^2-1)n)U_{m-1,n,o-1} + mny((3z^2-1)o + (z^2-1)(m+n+1)) \\ &\times U_{m-1,n-1,o} - 2mnoyz(m+n+o-2)U_{m-1,n-1,o-1} \end{aligned} \tag{19}$$

$$\begin{aligned} U_{m,n+1,o} &= y(m+2n+o+1)U_{m,n,o} + noyzU_{m,n,o-1} + mnxyU_{m-1,n,o} \\ &+ 2mnoxyzU_{m-1,n,o-1} + n((x^2+z^2-1)n + (x^2+2z^2-1)o \\ &+ (2x^2+z^2-1)m)U_{m,n-1,o} + noz((x^2-1)(n+o-1) \\ &+ (3x^2-1)m)U_{m,n-1,o-1} + mnx((3z^2-1)o + (z^2-1)(m+n-1)) \\ &\times U_{m-1,n-1,o} - 2mnoxz(m+n+o-2)U_{m-1,n-1,o-1} \end{aligned} \tag{20}$$

$$\begin{aligned} U_{m,n,o+1} &= z(m+n+2o+1)U_{m,n,o} + moxzU_{m-1,n,o} + noyzU_{m,n-1,o} \\ &+ 2mnoxyzU_{m-1,n-1,o} + o((x^2+y^2-1)o + (2x^2+y^2-1)m \\ &+ (x^2+2y^2-1)n)U_{m,n,o-1} + mox((y^2-1)(m+o-1) \\ &+ (3y^2-1)n)U_{m-1,n,o-1} + noy((x^2-1)(n+o-1) \\ &+ (3x^2-1)m)U_{m,n-1,o-1} - 2mnoxy(m+n+o-2)U_{m-1,n-1,o-1} \end{aligned} \tag{21}$$

and the polynomials $V_{m,n,o} = V_{m,n,o}(x,y,z)$ satisfies the recurrence relations

$$\begin{aligned} (2(m+n+o+1)+s)xV_{m,n,o} &= V_{m+1,n,o} - n(n-1)V_{m+1,n-2,o} \\ &- o(o-1)V_{m+1,n,o-2} + m(m+2n+2o+1+s)V_{m-1,n,o} \end{aligned} \tag{22}$$

$$\begin{aligned} (2(m+n+o+1)+s)yV_{m,n,o} &= V_{m,n+1,o} - m(m-1)V_{m-2,n+1,o} \\ &- o(o-1)V_{m,n+1,o-2} + n(2m+n+2o+1+s)V_{m,n-1,o} \end{aligned} \tag{23}$$

$$\begin{aligned} (2(m+n+o+1)+s)zV_{m,n,o} &= V_{m,n,o+1} - m(m-1)V_{m-2,n,o+1} \\ &- n(n-1)V_{m,n-2,o+1} + o(2m+2n+o+1+s)V_{m,n,o-1} \end{aligned} \tag{24}$$

with the initial conditions

$$\begin{aligned} U_{0,0,0} &= 1 & V_{0,0,0} &= 1 \\ U_{1,0,0} &= (2s-1)x & V_{1,0,0} &= (2s+1)x \\ U_{0,1,0} &= (2s-1)y & V_{0,1,0} &= (2s+1)y \\ U_{0,0,1} &= (2s-1)z & V_{0,0,1} &= (2s+1)z \\ U_{2,0,0} &= (2s-1)((2s+1)x^2+y^2+z^2-1) & V_{2,0,0} &= (2s+1)((2s+3)x^2-1) \\ U_{0,2,0} &= (2s-1)(x^2+(2s+1)y^2+z^2-1) & V_{0,2,0} &= (2s+1)((2s+3)y^2-1) \\ U_{0,0,2} &= (2s-1)(x^2+y^2+(2s+1)z^2-1) & V_{0,0,2} &= (2s+1)((2s+3)z^2-1) \\ U_{1,1,0} &= 2s(2s-1)xy & V_{1,1,0} &= (2s+1)(2s+3)xy \\ U_{1,0,1} &= 2s(2s-1)xz & V_{1,0,1} &= (2s+1)(2s+3)xz \\ U_{0,1,1} &= 2s(2s-1)yz & V_{0,1,1} &= (2s+1)(2s+3)yz. \end{aligned} \tag{25}$$

Since the magnitudes of the both families are unbalanced, we can balance them by

$$\tilde{U}_{m,n,o} = U_{m,n,o}\frac{1}{((m+n+o)!)^{\frac{9}{32}}} \times \sqrt{\frac{2(m+n+o)+3}{4\pi\left(\Gamma\left(\frac{m+n+o}{3}+1\right)\right)^{\frac{3}{2}}(\Gamma(m+n+o+1))^{\frac{1}{2}}}} \tag{26}$$

$$\tilde{V}_{m,n,o} = V_{m,n,o}\frac{1}{((m+n+o)!)^{\frac{23}{32}}} \times \sqrt{\frac{2(m+n+o)+3}{4\pi\left(\Gamma\left(\frac{m+n+o}{3}+1\right)\right)^{\frac{3}{2}}(\Gamma(m+n+o+1))^{\frac{1}{2}}}}. \tag{27}$$

Note that (26) and (27) are functions of the sum $m+n+o$ only, so the rotation invariance is not violated.

If we use the recurrence relations, we must compute all the polynomials from the zeroth degree up to the desired maximum degree. The entire algorithm is described in form of a pseudo-code (see Algorithm 1). Provided that the processed image can be inscribed into a sphere of radius r, then we need $2r+1$ coordinate values in each direction. The precomputed array P is then used for computation of Appell moments by (6).

Algorithm 1 Computation of Appell polynomials by recurrence formulas.

```
Input: Type $t$ is $U$ or $V$, Maximum order $Ord$,
       Vectors with coordinates $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$.
       Typically, $\mathbf{x}, \mathbf{y}, \mathbf{z} \leftarrow [-1, -1+1/r, -1+2/r, \ldots, 1]$.
Output: Four-dimensional array $\mathbf{P}$ with the computed polynomials.
$P \leftarrow$ array $(Ord+1) \times (Ord+1) \times (Ord+1) \times (2r+1)$ filled by zeros
Fill beginning of $P$ by the initial conditions from (B17). Use left or right column
of (B17) according to $t$ is $U$ or $V$.          ▷ Substitute $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ as $x$, $y$, and $z$.
for $or = 2 : Ord - 1$ do
    for $m = 0 : or$ do
        for $n = 0 : or - m$ do
            $o \leftarrow or - m - n$
            if $t$=$U$ then
                Fill $P(m+1, n, o, 0 : 2r)$ according to (B11).
                Fill $P(m, n+1, o, 0 : 2r)$ according to (B12).
                Fill $P(m, n, o+1, 0 : 2r)$ according to (B13).
                                          ▷ Substitute $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ as $x$, $y$, and $z$.
                If some index $m$, $n$, or $o < 0$, use the value 0 instead.
            end if
            if $t$=$V$ then
                Fill $P(m+1, n, o, 0 : 2r)$ according to (B14).
                Fill $P(m, n+1, o, 0 : 2r)$ according to (B15).
                Fill $P(m, n, o+1, 0 : 2r)$ according to (B16).
                                          ▷ Substitute $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ as $x$, $y$, and $z$.
                If some index $m$, $n$, or $o < 0$, use the value 0 instead.
            end if
        end for
    end for
end for
If needed, use the balance from (B18) for $t$=$U$ or from (B19) for $t$=$V$.
```

Appendix 3: H-NeXtA: Training Setup

This Appendix provides more detailed descriptions of the experiments summarized in Table 4. Except for the last row (CNN [62]+AUG), all models are trained without augmentation, and the experiments share the same hyperparameters and regularization techniques listed in Table 5. The convolutional architecture used for comparison is based on the original LeNet [62], but as shown in Fig. 9, it has been updated with modern features like batch normalization and ReLU activation functions.

Table 4 Recognition statistics of H-NeXtA on the rotated MNIST dataset [61]

Architecture	Test accuracy	Train samples	Augmentation	Number of param
CNN [62]	93.33 ± 0.30%	10,000	✗	113 262
CNN + AUG [62]	96.55 ± 0.12%	10,000	✓	113 262
H-NeXt + A	98.16 ± 0.02%	10,000	✗	19 800

The test accuracy represents the mean and standard deviation across 10 runs with different random seeds

Table 5 Hyperparameters for H-NeXtA experiments

Hyperparameter	Value
Batch size	64
Number of epochs	50–100
Optimizer	AdamW
Learning rate	0.007
Label smoothing	0.1
Weight decay	0.01

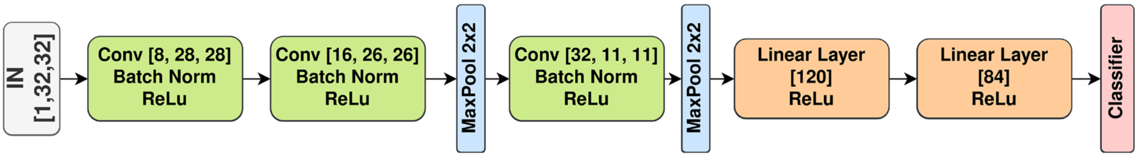


Fig. 9 Architecture of the modified LeNet with batch normalization and ReLU. Refered as CNN [62] in Table 4

Acknowledgements Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Funding Open access publishing supported by the institutions participating in the CzechELib Transformative Agreement. This work was funded by the Czech Science Foundation (GA24-10069S) and the Ministry of Education and Science of Ukraine (0119U100662) and the Ministry of Education, Youth and Sports of the Czech Republic (LM2018140).

Data availability Most of the datasets used in this study are publicly available and can be accessed through the references provided in the manuscript. The proprietary dataset, "The Statues" is not publicly accessible but can be provided upon reasonable request to the corresponding author.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Maturana D, Scherer S. VoxNet: a 3D convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS); 2015. p. 922-8.
2. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J. 3D ShapeNets: a deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1912–20.
3. Dai A, Ritchie D, Bokeloh M, Reed S, Sturm J, Nießner M. ScanComplete: large-scale scene completion and semantic segmentation for 3D scans. In: Proceedings of the IEEE conference on computer vision and pattern recognition CVPR; 2018. p. 4578–87.
4. Sinha A, Unmesh A, Huang Q, Ramani K. SurfNet: generating 3D shape surfaces using deep residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition CVPR; 2017. p. 6040–9.
5. Hatamizadeh A, Tang Y, Nath V, Yang D, Myronenko A, Landman B, Roth H, Xu D. UNETR: transformers for 3D medical image segmentation. In: Proceedings of the IEEE/CVF Winter conference on applications of computer vision; 2022. p. 574–84.
6. Liu Z, Tang H, Lin Y, Han S. Point-voxel CNN for efficient 3D deep learning. In: Advances in neural information processing systems. USA: Curran Associates, Inc.; 2019. p. 32.
7. Brock A, Lim T, Ritchie JM, Weston NJ. Generative and discriminative voxel modeling with convolutional neural networks. In: Neural information processing conference: 3D deep learning; 2016.
8. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in neural information processing systems, vol. 29. USA: Curran Associates, Inc.; 2016.
9. Liu J, Yu F, Funkhouser T. Interactive 3D modeling with a generative adversarial network. In: 2017 international conference on 3D vision (3DV); 2017. p. 126–34.
10. Xie J, Zheng Z, Gao R, Wang W, Zhu S-C, Wu YN. Generative VoxelNet: learning energy-based models for 3D shape synthesis and analysis. IEEE Trans Pattern Anal Mach Intell 2020;44(5):2468–84.
11. Xie J, Zheng Z, Gao R, Wang W, Zhu S-C, Wu YN. Learning descriptor networks for 3D shape synthesis and analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 8629–38.
12. Mao J, Xue Y, Niu M, Bai H, Feng J, Liang X, Xu H, Xu C. Voxel transformer for 3D object detection. In: Proceedings of the IEEE/CVF international conference on computer vision; 2021. p. 3164–73.
13. Fan L, Pang Z, Zhang T, Wang Y-X, Zhao H, Wang F, Wang N, Zhang Z. Embracing single stride 3D object detector with sparse transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022. p. 8458–68.
14. Park C, Jeong Y, Cho M, Park J. Fast point transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022. p. 16949–58.
15. Lahoud J, Cao J, Khan FS, Cholakkal H, Anwer RM, Khan S, Yang M-H. 3D vision with transformers: a survey; 2022. arXiv:2208.04309
16. Geiger A, Lenz P, Stiller C, Urtasun R. Vision meets robotics: the KITTI dataset. Int J Robot Res (IJRR). 2013;32(11):1231–7.
17. Zbontar J, et al. fastMRI: an open dataset and benchmarks for accelerated MRI. 2019. arXiv:1811.08839
18. Mumuni A, Mumuni F. CNN architectures for geometric transformation-invariant feature representation in computer vision: a review. SN Comput Sci. 2021;2(5):1–23.
19. Yu H-X, Wu J, Yi L. Rotationally equivariant 3D object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2022. p. 1456–64.
20. Wu H, Wen C, Li W, Li X, Yang R, Wang C. Transformation-equivariant 3D object detection for autonomous driving. 2022. https://doi.org/10.48550/arXiv.2211.11962
21. Flusser J, Suk T, Zitová B. 2D and 3D image analysis by moments. Chichester: Wiley; 2016.
22. Suk T, Flusser J. Tensor method for constructing 3D moment invariants. In: Computer analysis of images and patterns CAIP'11. Lecture notes in computer science. Berlin, Heidelberg, Germany: Springer; 2011. p. 6854–5, 212–9.
23. Suk T, Flusser J, Boldyš J. 3D rotation invariants by complex moments. Pattern Recogn. 2015;48(11):3516–26.
24. Canterakis N. 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. In: Proceedings of the 11th scandinavian conference on image analysis SCIA'99. Lyngby: Pattern Recognition Society of Denmark.; 1999. p. 85–93.
25. Yang B, Flusser J, Suk T. 3D rotation invariants of Gaussian-Hermite moments. Pattern Recogn Lett. 2015;54(1):18–26.
26. Bedratyuk L, Flusser J, Suk T, Kostková J, Kautský J. Non-separable rotation moment invariants. Pattern Recogn. 2022;127:108607–12.
27. Bedratyuk L. 2D geometric moment invariants from the point of view of the classical invariant theory. J Math Imaging Vis. 2020;62:1062–75.
28. Yang B, Flusser J, Kautsky J. Rotation of 2D orthogonal polynomials. Pattern Recogn Lett. 2018;102(1):44–9.

29. Dunkl CF, Xu Y. Orthogonal polynomials of several variables. Encyclopedia of mathematics and its applications. Cambridge: Cambridge University Press; 2014. p. 155.
30. DIP: 3D Rotation Moment Invariants 2011. http://zoi.utia.cas.cz/3DRotationInvariants. Accessed 3 Dec 2024.
31. Mahmmod BM, Abdulhussain SH, Naser MA, Alsabah M, Hussain A, Al-Jumeily D. 3d object recognition using fast overlapped block processing technique. Sensors. 2022. https://doi.org/10.3390/s22239209.
32. Koschutnig K. Openneuro. uploaded 2021; 2021. https://openneuro.org. Accessed 3 Dec 2024.
33. Long F, Peng H, Liu X, Kim SK, Myers E, Kainmüller D, Weigert M. 3D nuclei instance segmentation dataset of fluorescence microscopy volumes of *C. elegans*; 2022. https://zenodo.org/record/5942575#.YoYxYVTP0uV. Accessed 3 Dec 2024.
34. Long F, Peng H, Liu X, Kim SK, Myers E. A 3D digital atlas of *C. elegans* and its application to single-cell analyses. Nat Methods. 2009;6(9):667–72.
35. Stanford Vision Lab: ILSVRC 2015. http://www.image-net.org/challenges/LSVRC/. Accessed 3 Dec 2024.
36. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE CVPR; 2016. p. 770–8. https://doi.org/10.1109/CVPR.2016.90.
37. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. USA: Curran Associates, Inc.; 2012. p. 25. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. Accessed 3 Dec 2024.
38. Azulay A, Weiss Y. Why do deep convolutional networks generalize so poorly to small image transformations? J Mach Learn Res. 2019;20(184):1–25.
39. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13. Springer; 2014. p. 818–33.
40. Mumuni A, Mumuni F. CNN architectures for geometric transformation-invariant feature representation in computer vision: a review. SN Comput Sci. 2021. https://doi.org/10.1007/s42979-021-00735-0.
41. Nguyen DT, Pham TD, Baek NR, Park KR. Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors. Sensors. 2018;18:699.
42. Nanni L, Ghidoni S, Brahnam S. Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recogn. 2017;71:158–72.
43. Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Proceedings of the 27th international conference on neural information processing systems (NIPS'14), vol. 1; 2014. p. 568–76.
44. Anwer RM, Khan FS, Weijer J, Molinier M, Laaksonen J. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. ISPRS J Photogramm Remote Sens. 2018;138:74–85.
45. Hao Y, Li Q, Mo H, Zhang H, Li H. AMI-Net: convolution neural networks with affine moment invariants. IEEE Signal Process Lett. 2018;25:1064–8.
46. Levi G, Hassner T. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In: Proceedings of the 2015 ACM on international conference on multimodal interaction (ICMI). New York, USA: ACM; 2015. p. 503–10.
47. Connie T, Al-Shabi M, Cheah W, Goh M. Facial expression recognition using a hybrid CNN-SIFT aggregator. In: Phon-Amnuaisuk S, Ang S-P, Lee S-Y (eds.) Multi-disciplinary trends in artificial intelligence. Berlin, Heidelberg, Germany: Springer; 2017. p. 139–149. https://doi.org/10.1007/978-3-319-69456-6_12.
48. Tsourounis D, Kastaniotis D, Theoharatos C, Kazantzidis A, Economou G. Sift-cnn: when convolutional neural networks meet dense sift descriptors for image and sequence classification. J Imaging. 2022. https://doi.org/10.3390/jimaging8100256.
49. Xu K, Qin M, Sun F, Wang Y, Chen Y-K, Ren F. Learning in the frequency domain. In: Proceedings of the IEEE/CVF CVPR 2020.
50. Cohen T, Welling M. Group equivariant convolutional networks. In: Proceedings of The 33rd international conference on machine learning. New York, USA: PMLR; 2016. p. 2990–9.
51. Marcos D, Volpi M, Komodakis N, Tuia D. Rotation equivariant vector field networks. In: 2017 IEEE international conference on computer vision (ICCV); 2016. p. 5058–67.
52. Laptev D, Savinov N, Buhmann JM, Pollefeys M. TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. 2016 IEEE conference on computer vision and pattern recognition (CVPR); 2016. p. 289–97.
53. Freeman WT, Adelson EH. The design and use of steerable filters. IEEE Trans Pattern Anal Mach Intell. 1991;13(9):891–906. https://doi.org/10.1109/34.93808.
54. Cohen TS, Welling M. Steerable CNNs. In: International conference on learning representations; 2017. https://openreview.net/forum?id=rJQKYt5ll. Accessed 3 Dec 2024.
55. Worrall DE, Garbin SJ, Turmukhambetov D, Brostow GJ. Harmonic networks: deep translation and rotation equivariance. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society; 2017. p. 7168–7177. https://doi.org/10.1109/CVPR.2017.758.
56. Weiler M, Hamprecht FA, Storath M. Learning steerable filters for rotation equivariant CNNs. In: 2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR). USA: IEEE Computer Society; 2018. p. 849–58. https://doi.org/10.1109/CVPR.2018.00095.
57. Weiler M, Cesa G. General E(2)-equivariant steerable CNNs. In: Advances in neural information processing systems, vol. 32. USA: Curran Associates, Inc.; 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/45d6637b718d0f24a237069fe41b0db4-Paper.pdf. Accessed 3 Dec 2024.
58. Singh J, Singh C. Learning invariant representations for equivariant neural networks using orthogonal moments. In: 2022 international joint conference on neural networks (IJCNN). IEEE; 2022. p. 1–8.
59. Singh J, Singh C, Rana A. Orthogonal transforms for learning invariant representations in equivariant neural networks. In: Proceedings of the IEEE/CVF Winter conference on applications of computer vision (WACV); 2023. p. 1523–30.
60. Karella T, Šroubek F, Blažek J, Flusser J, Košík V. H-NeXt: the next step towards roto-translation invariant networks. In: 34th British machine vision conference 2023, BMVC 2023, Aberdeen, UK, November 20–24, 2023. UK: BMVA; 2023. p. 922–8. https://papers.bmvc2023.org/0578.pdf.
61. Larochelle H, Erhan D, Courville A, Bergstra J, Bengio Y. An empirical evaluation of deep architectures on problems with many factors of variation. In: Proceedings of the 24th international conference on machine learning, vol. 227; 2007. p. 473–80. https://doi.org/10.1145/1273496.1273556.
62. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324. https://doi.org/10.1109/5.726791.
63. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR); 2016. p. 770–8. https://doi.org/10.1109/CVPR.2016.90.
64. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Bengio Y, LeCun Y (eds.) 3rd

international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track Proceedings; 2015. arXiv:1409.1556.