Eurographics Workshop on 3D Object Retrieval (2018)
A. Telea, T. Theoharis, and R. C. Veltkamp (Editors)

Non-rigid 3D Model Classification Using 3D Hahn Moment Convolutional Neural Networks

A. Mesbah[1] , A. Berrahou[2], H. Hammouchi[1], H. Berbia[2], H. Qjidaa[1], M. Daoudi[3]

[1]Sidi Mohamed Ben Abdellah University, Fes, Morocco
[2]Mohammed V University, Rabat, Morocco
[3]IMT Lille Douai, Univ. Lille, CNRS, UMR 9189 CRIStAL, F-59000 Lille, France

Abstract

In this paper, we propose a new architecture of 3D deep neural network called 3D Hahn Moments Convolutional Neural Network (3D HMCNN) to enhance the classification accuracy and reduce the computational complexity of a 3D pattern recognition system. The proposed architecture is derived by combining the concepts of image Hahn moments and convolutional neural network (CNN), frequently utilized in pattern recognition applications. Indeed, the advantages of the moments concerning their global information coding mechanism even in lower orders, along with the high effectiveness of the CNN, are combined to make up the proposed robust network. The aim of this work is to investigate the classification capabilities of 3D HMCNN on small 3D datasets. The experiment simulations with 3D HMCNN have been performed on the articulated parts of McGill 3D shape Benchmark database and SHREC 2011 database. The obtained results show the significantly high performance in the classification rates of the proposed model and its ability to decrease the computational cost by training low number of features generated by the first 3D moments layer.

Keywords: ***Classification · 3D Hahn moments · Convolutional Neural Network · Hahn Moment Convolutional Neural Network.***

1. Introduction

In recent years, the excessive production of available 3D models in different scientific fields, has led researchers to develop efficient and fast classification tools for available 3D databases. Several methods have achieved excellent results by using hand-crafted features with a machine learning classifier [DXH15, FHK04, BSC12, TLT11, GKF09]. However, due to the complexity of these algorithms, they suffer from the lack of efficient extracted features. Additionally, they require considerable domain expertise, engineering skills, and theoretical foundations. Recent state-of-art methods utilize convolutional neural networks (CNN) to overcome this problem. They have been shown to be greatly effective classifiers for object data. The majority of published work has been devoted to solving 2D problems, while a few limited exceptions that treated the 3D space either as volumetric representations [WSK*15, QSN*16] or by using Multi-view techniques [HMKLM15, ZWB*16] and other methods [QSMG17]. Conventionally the input to a convolutional neural network classifier is an image patch, which increases enormously in size for 3D object. For example, a 3D patch of $32 \times 32 \times 32$ generates an input of 32 768 voxels to the classifier. Hence, such a large big input feature vector leads to an increase in computational cost. This can be attributed to the increased numbers of filters [SEZ*14], number of layers [SZ14], smaller strides [SEZ*14, ZF14], and their combinations.

In this paper, we propose a new architecture called Hahn Moments Convolutional Neural Network (HMCNN) that incorporate discrete moments as first layer which generate features from the input object and fed them into a corresponding 3D Convolutional Neural Network. In fact, the kernel functions of Hahn moments have relatively high spatial frequency components in lower orders. Therefore, 3D Hahn moments have the ability to extract discriminant features in lower orders, hence the computed descriptor vectors can capture more information with low dimensionality and high discrimination power. The introduction of moments layer in the architecture of the CNN makes it possible to design HMCNN models that reduce considerably the computational cost by decreasing number of layers and parameters while achieving best classification results. The experimental results showed that our proposed architecture achieved high classification accuracy on 3D deformable shape dataset as compared with other methods based on moments and other algorithms. additionally, the computational complexity is enormously reduced.

The rest of this paper is organized as follows: in section 2, an overview of 3D discrete Hahn moments is presented. The details of our proposed model 3D HMCNN are explained in section 3. Section 4 is dedicated to present the main results and performances of our proposed model. Finally, summary of important conclusions are drawn out in section 5.

DOI: 10.2312/3dor.20181056

2. 3D Hahn Moments

In this section we will present the mathematical background needed for the intro-duction of 3D Hahn moments including Hahn polynomials.

2.1. Hahn polynomials

Hahn polynomials of one variable x, with the order n, defined in the interval $[0,N-1]$ as given in [YPO07], respect the following equation:

$$h_n(\alpha,\beta,N|x) = {}_3F_2\left(\begin{matrix} -n, n+\alpha+\beta, -x \\ \alpha+1, -N \end{matrix} \middle| \, 1 \right) \tag{1}$$

with $n,x = 0,1,\cdots,N-1$
where α *and* β are free parameters, and ${}_3F_2$ is the generalized hyper-geometric function given by :

$${}_3F_2\left(\begin{matrix} a_1,a_2,a_3 \\ b_1,b_2 \end{matrix} \middle| \, z \right) = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k (a_3)_k}{(b_1)_k (b_2)_k k!} z^k \tag{2}$$

Hahn polynomials satisfy the orthogonal property:

$$\sum_{x=0}^{N-1} h_n\left(\alpha,\beta,N \mid x \right) h_m\left(\alpha,\beta,N \mid x \right) \omega_h(x) = \rho_h(n)\delta_{mn} \tag{3}$$

where $w_h(x)$ is the weighting function given by

$$\omega_h(x) = \frac{(\alpha+1)_x (\beta+1)_{N-x}}{(N-x)!x!} \tag{4}$$

while ρ_h is the squared-norm expressed by

$$\rho_h(n) = \frac{(-1)^n n!(\beta+1)_n (\alpha+\beta+n+1)_{N+1}}{(-N)_n (2n+\alpha+\beta+1) N!(\alpha+1)_n} \tag{5}$$

To assure the numerical stability, the set of the weighted Hahn polynomials is defined as

$$\tilde{h}_n\left(\alpha,\beta,N \mid x \right) = h_n\left(\alpha,\beta,N \mid x \right) \sqrt{\frac{w_h(x)}{\rho_h(n)}} \tag{6}$$

The set of weighted Hahn polynomials obeys the three term recurrence relation defined as follow

$$\begin{aligned} \tilde{h}_n\left(\alpha,\beta,N \mid x \right) =& A\sqrt{\frac{\rho_h(n-1)}{\rho_h(n)}} \tilde{h}_{n-1}\left(\alpha,\beta,N \mid x \right) \\ & - B\sqrt{\frac{\rho_h(n-2)}{\rho_h(n)}} \tilde{h}_{n-2}\left(\alpha,\beta,N \mid x \right) \\ & n = 2,3,\cdots,N-1 \end{aligned} \tag{7}$$

Where

$$A = 1 + B - x\frac{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}{(n+\alpha+\beta+1)(\alpha+n+1)(N-n)} \tag{8}$$

$$B = \frac{n(n+\beta)(\alpha+\beta+n+N+1)(2n+\alpha+\beta+2)}{(2n+\alpha+\beta)(\alpha+\beta+n+1)(\alpha+n+1)(N-n)} \tag{9}$$

The initial values for the above recursion can be obtained from

$$\tilde{h}_0\left(\alpha,\beta,N \mid x \right) = \sqrt{\frac{\omega_h(x)}{\rho_h(0)}} \tag{10}$$

$$\tilde{h}_1\left(\alpha,\beta,N \mid x \right) = \left(1 - \frac{x(\alpha+\beta+2)}{(\alpha+1)N}\right) \sqrt{\frac{\omega_h(x)}{\rho_h(1)}} \tag{11}$$

2.2. 3D Hahn moments

The 3D discrete Hahn moments of order m+n+l of an image intensity function $f(x,y,z)$ are defined over the cube $[0,N-1]\times[0,N-1]\times[0,N-1]$ as:

$$\begin{aligned} H_{mnl} = \sum_{x=o}^{N-1}\sum_{y=o}^{N-1}\sum_{z=0}^{N-1} & \tilde{h}_m\left(\alpha,\beta,M \mid x \right) \tilde{h}_n\left(\alpha,\beta,N \mid x \right) \\ & \tilde{h}_l\left(\alpha,\beta,N \mid x \right) f(x,y) \end{aligned} \tag{12}$$

where $\tilde{h}_m\left(\alpha,\beta,M \mid x \right)$, $\tilde{h}_n\left(\alpha,\beta,N \mid x \right)$ and $\tilde{h}_l\left(\alpha,\beta,N \mid x \right)$ denote the weighted polynomials.
Due to the orthogonal property of the weighted polynomials, the 3D image intensity function $f(x,y,z)$ can be expressed over cube $[0,N-1]\times[0,N-1]\times[0,N-1]$ as:

$$\begin{aligned} \tilde{f}(x,y) = \sum_{x=o}^{N-1}\sum_{y=o}^{N-1}\sum_{z=0}^{N-1} & \tilde{h}_m\left(\alpha,\beta,M \mid x \right) \tilde{h}_n\left(\alpha,\beta,N \mid x \right) \\ & \tilde{h}_l\left(\alpha,\beta,N \mid x \right) H_{mn} \end{aligned} \tag{13}$$

It should be noted that 3D Hahn moments can be used as a descriptor of any 3D object, if it can be expressed as a function $f(x,y,z)$ defined in a discrete space $[0,N-1]\times[0,N-1]\times[0,N-1]$. This can be achieved if the model is expressed as a binary volumetric function. Table 1 shows some reconstructions binary objects up to orders 5, 15, 25 respectively by using 3D Hahn moments. The original binary object is a 3D STL model of human head which is converted into $128\times128\times128$ volumetric representation by utilizing an appropriate voxelization method as depicted in Fig. 1. Table 2 shows some reconstructions gray scale objects up to orders 15, 30, 60 respectively by using 3D Hahn moments from the 3D MRI image (Fig. 2) .To have best reconstruction results, we take $\alpha = \beta$=5 [MBMQ16]. We can observe more resemblance between the original object and reconstructed ones in the early orders. The reconstruction abilities of 3D Hahn moments indicate their capacity to compact more information from image that are important for classification.

Figure 1: *Original binary object with size of* $128 \times 128 \times 128$

Figure 2: *Original gray scale object with size of* $128 \times 128 \times 128$

3. 3D Hahn Moments Convolutional Neural Network

In this work, we proposed a new architecture called Hahn moments convolutional neural network (HMCNN) for classification tasks that incorporates the concept of orthogonal moments in 3D CNN structure as shown in Fig.3. Indeed, these moments can represent the image more effectively for low orders as illustrated in the previous section. This makes it possible to generate a 3D moment matrices with small size which are fed to 3D convolutional network instead of 3D image matrices. Hence, our proposed architecture reduces considerably the complexity processing and speed up the computational time. The HMCNN architecture is hierarchically structured as a stack of layers. It generally distinguishes four layers types as follow:

3D Moment layer: Through this layer, the input object is transformed to 3D matrix of moments by using Eq.12. The generated 3D matrix is generally smaller than the input 3D object matrix since the lower orders of the moments can describe well the image as demonstrated in the previous section. This matrix of moments is then fed to the following 3D convolutional layers.

3D Convolution layer: In this layer we applied 3D convolution operators on 3D moment matrix instead of 3D image matrix as in the standard 3D CNN architecture. The output activation value $a_(i,j,k)^L$ at position (i,j,k) is calculated by the following equation:

$$a_(i,j,k)^L = f\left(\sum_{x=i}^{i+N-1}\sum_{y=j}^{j+N-1}\sum_{z=k}^{k+N-1}\sum_{s=0}^{S-1} W_{s,x,y,z} M_{s,x,y,z} + b^L\right) \tag{14}$$

Where the matrix of moments M convolves with the L^{th} filter with size of $N \times N \times N$, S is the number of input channels, W is the weight matrix with size (C,N,N,N), i,j,k are the indices of the output position, x,y,z are the indices of the input position. f is activation function.

Activation functions ReLu and Elu: ReLu refers to rectified linear unit that has the mathematical expression $f(x) = max(0,x)$. In this work, we adopt this function instead of sigmoid function for some desirable reasons. Firstly, the use of this function overcomes the vanishing gradient problem found in traditional sigmoid activation functions. Secondly, convolutional neural network with ReLu need less training time and hence increase convergence speed [XWCL15]. Third, the use of this function increases the nonlinearity and avoids network saturation [GB10]. Exponential linear units (ELU) is a variant of ReLU activation functions that has negative values which allows them to push mean unit activations closer to zero. Zero means speed up learning because they bring the gradient closer to the natural gradient unit [CUH16].

Batch normalization BN: Batch normalization is used after each convolutional layer in order to further accelerate the training set, as well as reducing the gradients dependencies and avoid the risk of overfitting and divergence [IS15].

3D Max-pooling layer: is a down-sampling method applied to reduce the size of the hidden layers of the 3D CNN by an integer multiplicative factor. Through the pooling layers, only the most strongly activated information persists. This leads to faster training time and to avoid overfitting.

Fully Connected Layer: Fully-connected layers are placed generally in the last stages of the 3D CNN before the output layer and transfer the neural network into a vector with a predefined length. The formed vector can be used to construct the desired number of outputs or take it as a feature vector for follow-up processing [CMS12].

4. Experiments

In the experiment, we evaluated the performance of the proposed architecture 3D HMCNN on two databases: the articulated parts of McGill 3D shape Benchmark database [ZKCS08] and SHREC 2011. The first one consists of 255 articulated objects distributed over 10 classes: Ants, Crabs, Hands, Humans, Octopuses, Pliers, Snakes, Spectacles, Spiders, and Teddy bears. Each class contains 20 to 30 models. The second database contains 600 mesh

Table 1: *Reconstructed binary objects*

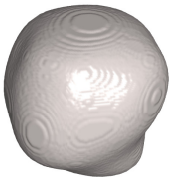
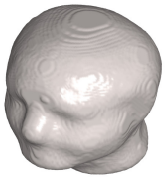
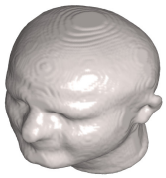
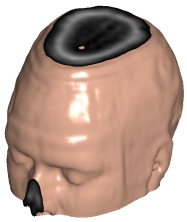
(5,5,5)	(15,15,15)	(25,25,25)
		

Table 2: *Reconstructed gray scale objects*

(15,15,15)	(30,30,30)	(60,60,60)
		

models, which are obtained from transforming 30 original models [LGB*11].The names of these 30 classes are: alien, ant, armadillo, bird1, bird2, camel, cat, centaur, dino skel, dinosaur, dog1, dog2, flamingo, glasses, gorilla, hand, horse, lamp, man, octopus, paper, pliers, rabbit, santa, scissor, shark, snake, spider, twoballs and woman. Since 3D CNN requires a fixed-size representation for the input data, we are used the voxelized format available on the McGill website. All experiments were performed on machine with NVIDIA Tesla K80, 12 GB of GDDRS memory.

4.1. Classification

In the classification tasks on McGill database we have randomly selected 10 objects from each class for training and the rest for test. Concerning the second database SHREC 2011, we have performed the classification on the original database and two others augmented SHREC 2011 datasets by applying some rotations to each object. Thus we considered the database SHREC 2011 without any augmentation noted S1 with 600 objects. A second database, noted S2, that contains 1200 objects, was generated by applying one rotation of 180 degree for each object in SHREC 2011 database. A third database, noted S3, that contains 3600 objects, was generated by applying rotations of each object from 0 to 300 degrees with a step of 60 degrees. In the experiment, for each aforementioned database, we have randomly selected 50% objects for the training set and 50% objects for the test set. The utilized network architecture is summarized in Table 5. The first layer generates low size matrix of $n \times n \times n$ where n represents the moment order. The features matrix is filtered in the second layer by applying 100 filters with size of $1 \times 1 \times 1$, stride of 1. We obtained 100 feature maps with size of $n \times n \times n$. In the third layer, we are used 60 filters with size of $3 \times 3 \times 3$ to generate 60 feature maps and 35 filters with the same size in the followed layer. Then we dawn-sampling the obtained features maps in the maxpooling layer with size of $2 \times 2 \times 2$ and stride of 2. This operation generates 35 feature maps with size of $\frac{n}{2} \times \frac{n}{2} \times \frac{n}{2}$. Two fully connected with numbers of neurones 300 and 240 are respectively used in sixth and seventh layers. The last softmax layer outputs number of labels corresponding to utilized classes. Table 3 presents the classification accuracy results on the articulated 3D McGill dataset for some lower orders. It can be seen that the high results are obtained for the moments orders up to 12. Beyond this order, the classification rate decreases progressively. The comparison to other methods is showed in Table7, we can observe the superiority of our proposed method against 5 other methods. Table 6 shows the confusion matrix across all 10 classes. Most model confusions are very reasonable showing that our model provide high quality features. The only class that we consistently misclassify are Hands which are very similar in appearance to Oc-

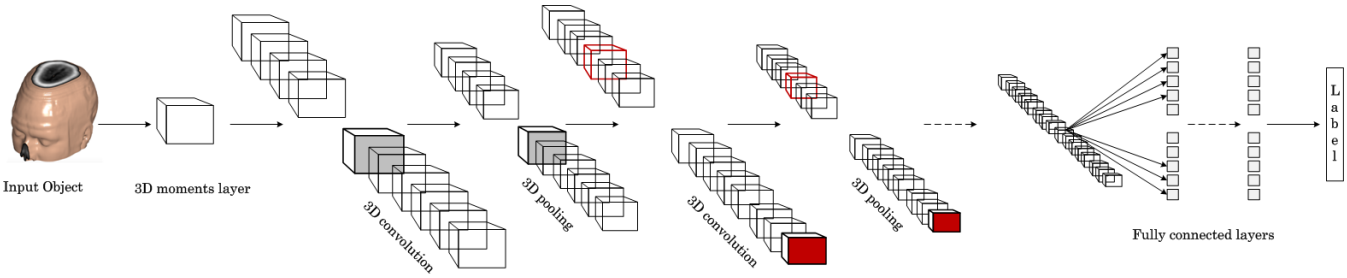


Figure 3: *3D Hahn moments convolutional neural networks architecture*

topuses. The obtained results over the three databases of SHREC 2011 are depicted in table 8. It can be observed that the classification rate increase as the order of Hahn moments increase for all databases S1, S2 and S3. Table 9 compares the performance of our proposed model to other methods that use the augmented SHREC 2011 database S3. It can be seen that our proposed model 3D HMCNN with Hahn moments layer achieves the best classification result. It should be noted that the classification rate of 89.33% is achieved without any augmentation of the SHREC 2011 database. This rate outperforms the results obtained by [SBR16] which used augmentation SHREC 2011 database.

4.2. Complexity

The proposed architecture 3D HMCNN reduces drastically the computational complexity. Indeed, many recent researches work on small voxel grids in the order of $32 \times 32 \times 32$ to reduce the complexity of 3D convolutional neural networks, which generates an input of 32 768 voxels to the classifier. The introduction of the 3D moment layer in the 3D CNN architecture reduces the generated parameters to $12 \times 12 \times 12 = 1728$, where 12 is the moments order that provides the best classification rate (see Table 3).

4.3. Visualization

The visualization technique was developed to understand the intermediate layers behavior and further to improve the model utilized [ZF14]. Indeed, the new representation for an image generated by each layer in CNN is projected back to the pixel space for visualizing constructed patches and understanding what information is kept. In this part, we attempt to provide insight into the internal representation of our model 3D HMCNN by visualizing objects in the representation spaces produced by the three main layers of our model. For this, we are utilized Eq.13 to compute the inverse transformation of convolved moments matrices extracted from layers as depicted in Table 4. The visualizations of the reconstructed objects from the output features of the three convolutional layers show the ability of 3D HMCNN to extract discriminant informations.

Table 3: *Classification accuracy on articulated McGill dataset for different orders of moments*

Order	04	08	12	16	20
Accuracy	86.15%	84.42%	**86.92**%	83.85%	83.08%

Table 4: *Examples of convoluted objects by using one arbitrary filter from each convolutional layer*

Original object	Conv1	Conv2	Conv3
			
			
			
			

Table 5: *Details of proposed 3D HMCNN*

Layer	Purpose	Filter	# of filters	Stride	Activation
0	Input object				N×N×N
1	Moment layer				n×n×n
2	Conv+BN+ELu	1×1×1	100	1	n×n×n×100
3	Conv+BN+ELu	3×3×3	60	1	n×n×n×60
5	Conv+BN	3×3×3	40	1	$n \times n \times$n×35
6	Maxpooling+ReLU	2×2×2	-	2	$\frac{n}{2} \times \frac{n}{2} \times \frac{n}{2} \times 35$
7	Fully connected				300
8	Fully connected				240
9	Softmax				number of subjects

Table 6: *Confusion matrix*

	Ants	Crabs	Hands	Humans	Octopuses	Pliers	Snakes	Spectacles	Spiders	Teddy
Ants	12	0	1	0	0	0	2	0	0	0
Crabs	0	15	0	0	0	0	0	0	0	0
Hands	1	0	5	0	4	0	0	0	0	0
Humans	0	0	0	13	2	0	0	0	0	0
Octopuses	0	2	0	0	10	0	0	0	1	0
Pliers	0	0	0	0	0	10	0	0	0	0
Snakes	0	0	0	0	0	0	13	0	0	0
Spectacles	0	0	0	0	0	0	0	13	0	0
Spiders	0	1	0	0	2	0	0	0	13	0
Teddy	0	0	1	0	0	0	0	0	0	9

Table 7: *Comparison of classification results on McGill dataset to other methods.*

Methods	Zer [NK04]	LFD [CTSO03]	SN [WSK*15]	Conf [GWC*04]	Sph [SM06]	GI [SBR16]	Our
Accuracy	63.0%	75.0%	65.0%	55.0%	62.0%	83.0%	**86.92%**

Table 8: *Classification accuracy on S1, S2, S3 databases for different orders of moments*

Order	04	08	12	16	20
S1	74.67%	81.33%	82.67%	83.67%	**89.33%**
S2	88.49%	90.5%	90.5%	91.33%	**92.67 %**
S3	95.95%	97.57%	97.90%	98.10%	**99.38%**

5. Conclusion

This work introduces a new architecture 3D HMCNN for 3D pattern classification by combining 3D Hahn moments and 3D Convolutional Neural Networks. This technique makes use of the main advantages of the image expansion on its moments and the Convolutional Neural Networks. The 3D moment layer is used to generate effective descriptors from the input object, which are then fed to 3D CNN. Furthermore, since 3D Hahn moment provide more distinctive features in the earliest orders , the complexity of 3D HMCNN was tremendously reduced by decreasing number of layers and parameters. Experiment results showed high performance of the proposed model on small datasets of deformable 3D models as compared to other works. As future work, it will be interesting to investigate the accuracy of 3D moments convolutional neural networks on large datasets by designing architectures that can achieve very competitive accuracy.

References

[BSC12] BEHLEY J., STEINHAGE V., CREMERS A. B.: Performance of histogram descriptors for the classification of 3D laser range data in urban environment. In *IEEE International Conference on Robotics and Automation (ICRA)* (2012). doi:10.1109/ICRA.2012.6225003. 1

[CMS12] CIRESAN D., MEIER U., SCHMIDHUBER J.: Multi-column deep neural networks for image classification. In *CVPR* (2012). 3

[CTSO03] CHEN D., TIAN X., SHEN Y., OUHYOUNG M.: On visual similarity based 3D model retrieval. *Computer Graphics Forum 22* (2003), 223–232. 6, 7

[CUH16] CLEVERT D., UNTERTHINER T., HOCHREITER S.: Fast and accurate deep network learning by exponential linear units. In *ICLR* (2016). doi:arXiv:1511.07289. 3

[DXH15] DU T., XU P., HU R.: *CS 221 Final Report: 3D Shape Classification*. 2015. 1

[FHK04] FROME A., HUBER D., KOLLURI R.: Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision* (2004), vol. 1, pp. 224–237. doi:https://doi.org/10.1007/978-3-540-24672-5_18. 1

[GB10] GLOROT X., BENGIO Y.: Understanding the difficulty of training deep feedforward neural networks. In *13th International Conference*

Table 9: *Comparison of classification results on S3 database to other methods.*

Methods	Zer [NK04]	LFD [CTSO03]	SN [WSK*15]	Conf [GWC*04]	Sph [SM06]	GI [SBR16]	Our
Accuracy	43.3%	56.7%	52.7%	60.6%	59.0%	88.6%	**99.38%**

on Artificial Intelligence and Statistics, Italy (2010), vol. 9, pp. 249–256. 3

[GKF09] GOLOVINSKIY A., KIM V. G., FUNKHOUSER T.: Shape-based recognition of 3D point clouds in urban environments. In *International Conference on Computer Vision (ICCV)* (2009). doi:10.1109/ICCV.2009.5459471. 1

[GWC*04] GU X., WANG Y., CHAN T., THOMPSON P., YAU S.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging 23* (2004), 949–958. 6, 7

[HMKLM15] HANG S., MAJI S., KALOGERAKIS E., LEARNED-MILLER E.: Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE international conference on computer vision* (2015). 1

[IS15] IOFFE S., SZEGEDY C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *2nd International Conference on Machine Learning, France* (2015), pp. 448–456. 3

[LGB*11] LIAN Z., GODIL A., BUSTOS B., DAOUDI M., HERMANS J., KAWAMURA S., KURITA Y., LAVOUÉ G., NGUYEN H., OHBUCHI R., AL E.: shape retrieval on non-rigid 3D watertight meshes. In *3DOR* (2011). 4

[MBMQ16] MESBAH A., BERRAHOU A., MALLAHI M. E., QJIDAA H.: Fast and efficient computation of three-dimensional hahn moments. *J. Electron. Imaging 25* (2016). doi:10.1117/1.JEI.25.6.061621. 2

[NK04] NOVOTNI M., KLEIN R.: Shape retrieval using 3D zernike descriptors. *Computer Aided Design 36* (2004), 1047–1062. 6, 7

[QSMG17] QI C. R., SU H., MO K., GUIBAS L.: Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Computer Vision and Pattern Recognition (CVPR)* (2017). 1

[QSN*16] QI C. R., SU H., NIESSNER M., DAI A., YAN M., GUIBAS J.: Volumetric and multi-view cnns for object classification on 3d data. In *Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 5648–5656. doi:10.1109/CVPR.2016.609. 1

[SBR16] SINHA A., BAI J., RAMANI K.: Deep learning 3D shape surfaces using geometry images. In *Computer Vision-ECCV 2016. Lecture Notes in Computer Science* (2016), Springer, Cham, pp. 223–240. doi:10.1007/978-3-319-46466-4_14. 5, 6, 7

[SEZ*14] SERMANET P., EIGEN D., ZHANG X., MATHIEU M., FERGUS R., LECUN Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR2014)* (2014), vol. arxiv.org/abs/1312.6229. 1

[SM06] SHEN L., MAKEDON F.: Spherical mapping for processing of 3-d closed surfaces. *Image and Vision Computing 24* (2006), 743–761. doi:10.1016/j.imavis.2006.01.011. 6, 7

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. In *CVPR* (2014). doi:arXiv:1409.1556v6. 1

[TLT11] TEICHMAN A., LEVINSON J., THRUN S.: Towards 3D object recognition via classification of arbitrary object tracks. In *IEEE International Conference on Robotics and Automation (ICRA)* (2011). doi:10.1109/ICRA.2011.5979636. 1

[WSK*15] 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), pp. 1912–1920. doi:10.1109/CVPR.2015.7298801. 1, 6, 7

[XWCL15] XU B., WANG N., CHEN T., LI M.: Empirical evaluation of rectified activations in convolutional network. In *Proceedings of the IEEE international conference on computer vision* (2015). doi:arXiv:1505.00853. 3

[YPO07] YAP P. T., PARAMESRAN R., ONG S. H.: Image analysis using hahn moments. *IEEE Trans. Pattern Anal. Mach. Intell 29*, 11 (2007), 2057–2062. doi:10.1109/TPAMI.2007.70709. 2

[ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional neural networks. In *CVPR* (2014). doi:arXiv:1311.290. 1, 5

[ZKCS08] ZHANG J., KAPLOW R., CHEN R., SIDDIQI K.: Retrieving articulated 3D models using medial surfaces. *Machine Vision and Applications 19* (2008), 261–274. 3

[ZWB*16] ZHU Y., WANG X., BAI S., YAO C., BAI X.: Deep learning representation using autoencoder for 3D shape retrieval. *Neurocomputing 204* (2016), 41–50. doi:https://doi.org/10.1016/j.neucom.2015.08.127. 1