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

Performing Image-like Convolution on Triangular Meshes

Claudio Tortorici[1], Naoufel Werghi[1] and Stefano Berretti[2]

[1]Khalifa University, Visual Signal Analysis and Processing Center (VSAP), United Arab Emirates
[2]University of Florence, Media Integration and Communication Center (MICC), Italy

Abstract

Image convolution with a filtering mask is at the base of several image analysis operations. This is motivated by Mathematical foundations and by the straightforward way the discrete convolution can be computed on a grid-like domain. Extending the convolution operation to the mesh manifold support is a challenging task due to the irregular structure of the mesh connections. In this paper, we propose a computational framework that allows convolutional operations on the mesh. This relies on the idea of ordering the facets of the mesh so that a shift-like operation can be derived. Experiments have been performed with several filter masks (Sobel, Gabor, etc.) showing state-of-the-art results in 3D relief patterns retrieval on the SHREC'17 dataset. We also provide evidence that the proposed framework can enable convolution and pooling-like operations as can be needed for extending Convolutional Neural Networks to 3D meshes.

CCS Concepts

•***Computing methodologies*** → ***Biometrics;*** *3D imaging; Computer vision representations; Neural networks; Mesh geometry models; Shape analysis;*

1. Introduction

The convolution operation is at the base of many computations in Mathematics, Physics, and Engineering. In particular, its discrete version has found large application in image analysis, where it is used to perform image filtering, in a broad sense. For example, convolution of an image with differently formed masks allows operations that go from edge detection (Sobel mask), to smoothing (Gaussian mask), derivatives (Laplacian mask), and so on. This success is mainly motivated by the grid structure of images, which allows the definition of masks and the effective implementation of convolution with multiplications, sums and shifts.

The capability of the convolution operation of extracting meaningful patterns from an image and its effective computation are also at the base of its extensive use in Convolutional Neural Networks (CNNs). These architectures have been known since the '80s [LBH15], but they have been now re-discovered thanks to the availability of sufficient computational power and training data. A common trait of these architectures is the application of the convolution operation to the input images through a series of convolutional layers using filters with different size, shift amount (stride) and padding. After non-linearity layers usually introduced with Rectified Linear Units (ReLu), down-sampling is also performed using some form of pooling (e.g., max-pooling). Extensions and adaptations of CNN models have been also tried in other domains than images [BBL*17]. Among them, the mesh manifold support is of particular interest since it is largely used for modeling 3D objects either obtained synthetically or acquired with 3D scanners. Thus, replicating convolution and pooling on the mesh can be the base for extending CNNs to the mesh domain. However, direct application of the convolution operation to 3D meshes is not possible, due to the lack of the regular grid structure of images.

In this paper, we propose a new framework that enables convolutions on 3D meshes, thus opening the way to a wide spectrum of filtering operations in this domain. We moved by the idea of Ordered Ring Facets [WRK12] that, given a facet on the mesh, allowed us to provide a local ordering of its neighbors. With this ordering, and a local reference frame, extension of the convolution becomes possible by emulating the 2D-like shift operation. In addition to reporting the results obtained by applying several filtering operations on the mesh, we also experimented the use of our proposed framework in the task of 3D relief patterns retrieval. Results show that the proposed solution outperforms state-of-the-art on the SHREC'17 dataset. Furthermore, we provide evidence that the proposed approach can be successfully used to perform convolution and pooling operations on 3D meshes. This opens the way to the extension of CNN architectures to this domain.

The rest of the paper is organized as follows: In Section 2, we summarize related work on convolutional-like operations on the mesh; Our proposed approach for performing convolution on the mesh is described in Section 3; Experimental results and comparative evaluation for the task of relief patterns retrieval are given in Section 4, where we also show results of convolution and pooling operations on the mesh; Conclusions and perspective for future work conclude the paper in Section 5.

DOI: 10.2312/3dor.20181060

2. Related Work

Most of the work on extending the convolution operation to non-Euclidean domains (e.g., graphs, manifolds, meshes, etc.) has been developed recently, riding the wave of success of CNN models.

One option that has been practiced in the literature is that of extending convolution to graphs. For example, Bruna et al. [BZSL14] considered generalizations of CNNs to signals defined on more general domains, without the action of a translation group. They showed that for low-dimensional graphs it is possible to learn convolutional layers with a number of parameters independent of the input size, resulting in efficient deep architectures. Kipf and Welling [KW17] followed the idea of spectral convolution on graphs, which is defined as the multiplication of a signal (a scalar for every node) with a filter parameterized in the Fourier domain (a function of the eigenvalues of the Laplacian). This might be prohibitively expensive so a localized first-order approximation of spectral graph convolution was used. Defferrard et al. [DBV16] presented a formulation of CNNs in the context of spectral graph theory, which designs fast localized convolutional filters on graphs.

The above methods mainly target very irregular and general graphs as can be generated in social networks, brain connectomes or words' embedding. However, triangular surface meshes are usually more regular, and resorting them to graphs can loose the power of CNN representations. Some researchers transformed clouds or meshes to regular 3D voxel grids, collections of images (e.g., views) or surfaces before feeding them to a deep net architecture. In [WSK*15], Wu et al. proposed to represent a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid, using a Convolutional Deep Belief Network. In the work of Sinha et al. [SBR16], the 3D shape is converted into a "geometry image" so that standard CNNs can directly be used to learn 3D shapes. Xie et al. [XFZW15] proposed a high-level shape feature learning scheme (*DeepShape*) to extract features that are insensitive to deformations via a discriminative deep auto-encoder. Fang et al. [FXD*15] developed novel techniques to extract a concise, but geometrically informative, shape descriptor and new methods of defining Eigen-shape and Fisher-shape descriptors to guide the training of a deep Neural Network. This data representation transformation, however, renders the resulting data unnecessarily voluminous, while also introducing quantization artifacts that can obscure natural invariances of the data. To overcome such limitations, Qi et al. [QSMG17] designed a novel type of Neural Network, named *PointNet*, that directly utilizes point clouds, while respecting the permutation invariance of points in the input.

Using a different view, mesh surfaces serve as a natural parametrization to 3D shapes, but learning surfaces using CNNs is a challenging task. Current paradigms to tackle this challenge are to either adapt the convolutional filters to operate on surfaces or learn spectral descriptors defined by the Laplace-Beltrami operator. Boscaini et al. [BMM*15] proposed a generalization of CNNs to non-Euclidean domains for the analysis of deformable shapes. Their construction was based on localized frequency analysis that is used to extract the local behavior of some dense intrinsic descriptor, roughly acting as an analogy to patches in images. In [MBBV15], Masci et al. extended the CNN paradigm to non-Euclidean manifolds by using a local geodesic system of polar coordinates to extract "patches". On these patches, a geodesic convolution on the mesh can be computed. Seong et al. [SPP17] proposed a geometric CNN (gCNN) that deals with data representation over a mesh surface and renders pattern recognition in a multi-shell mesh structure. To efficiently compute 3D convolutions with an arbitrary kernel size, Wang et al. [WLG*17] built a hash table to quickly construct the local neighborhood volume of eight sibling octants and computed the 3D convolutions of these octants in parallel.

3. Convolution on Mesh Manifolds

In mathematics, the convolution is an operation between two functions. The possibility to perform convolutions on images lies on the structure of the image itself. The image, indeed, is a grid of pixels that allows us to easily define a shift operator for the convolution operation. On the other hand, it is not natively possible to compute such operation on a mesh manifold. In fact, a mesh is defined by a cloud of points and connections between such points, regardless of their position: consecutive points on the stored mesh structure can be far from each other in 3D space.

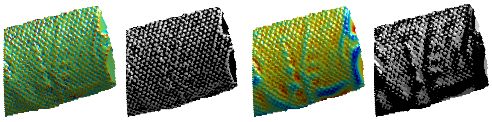
Using the Ordered Ring Facet (ORF) solution proposed in [WRK12], and already successfully used to generate Local Binary Patterns directly on a mesh manifold [WTBD15b,WTBD15a], we can derive a local order at each facet. The ORF, starting with adjacent f_{out} facet to a central facet f_c, linearly derives an ordered ring. Such linear process can be repeated for larger radii r to get multiple rings and extend the coverage of the ordered area. Since the first facet of each ring is aligned to the same axes, we can use such ordered structure as a polar grid to perform convolution facet-by-facet on the mesh.

The possibility to inherit the order from the ORF allows us to define a shift operator on the mesh surface. While on the image the neighbors of a pixel are given by the Cartesian coordinates derived by the grid structure of the image itself, with the ORF we use polar coordinates, i.e., radius r and quantized angle θ. Therefore, the convolution between a given mesh M and a filter F is defined as follow:

$$(M * F) = \sum_r \sum_\theta m_{r,\theta} \cdot f_{r,\theta} \ , \tag{1}$$

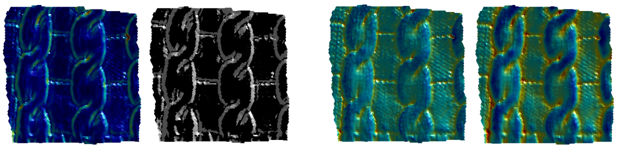
where $m_{r,\theta}$, and $f_{r,\theta}$ are, respectively, a scalar function computed on the mesh and the filter values, both at radius r and angle θ. In images, the convolution is performed at each pixel: neighbor pixels are multiplied by the filter values; in our proposed approach, instead, the convolution is performed on the facets, therefore filter values have to be determined at each facet of the ring. In image processing, it is possible to differentiate between *discrete* and *continuous* filters: 1) *discrete filters* are defined and represented by $N \times N$ matrices, as edge detectors, Sobel, etc.; 2) *continuous filters*, instead, are generated by continuous functions and quantized to fit a $N \times N$ window. Assuming to have a regular triangular mesh, the number of facets per ring is $r \cdot 12$, where r is the ring number. We can distinguish between discrete and continuous filters: for continuous filters, we can obtain the exact filter value at each facet by applying the function with polar coordinates; for discrete filters, we can devise two approaches: 1) Adapt the number of values at each ring, 2) Obtain $r \cdot 12$ values from the discrete filter, i.e., adapting the $N \times N$ values of the 2D filter to the ORF structure.

0	1	0
1	-4	1
0	1	0



(a) **(b)** **(c)**

Figure 1: *Convolution of the edge detector mask in (a) with a mesh: (b) the number of values of the ORF has been adapted to the filter; (c) the filter is adapted to the ORF structure. Both in (b) and (c), the filter response is shown on the left, while the thresholded values are given on the right.*



(a) *Sobel* **(b)** *Sharpener*

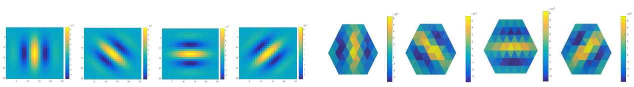
Figure 2: *(a) Convolution of the mesh with the Sobel operator, applied on local depth as a scalar function on the mesh, without and with threshold (on the left and on the right, respectively); (b) the same scalar function on the surface, and its enhancement after applying the Sharpen filter.*

Both methods can be implemented by interpolation and/or sub-sampling. A visual comparison of the two solutions is shown in Figure 1. As expected, adapting the values on the ORF to the discrete filter gives better results with respect to the second approach, since altering the 2D filter structure also modifies its original behavior. From now on, the first method will be used for discrete 2D filters.

To demonstrate that our proposed approach makes possible to reproduce any discrete filter on a mesh manifold, we replicate the Sobel and the sharpen filters. They show, respectively, a better edge detection and an enhancement of any scalar function computed on the mesh manifold (see, respectively, Figure 2a and 2b). As continuous filter, we chose Gabor. In this case, we have been able to get the $r \cdot 12$ values at each ring r, simply using the Gabor formula in polar coordinates. Figure 3 shows the 2D Gabor filter generated on a standard image, and our proposed counterpart on the mesh.

4. Experimental Results

Recently, a new database has been released for the SHREC'17 contest track on "Retrieval of surfaces with similar relief patterns" [BTA*17]. The database includes different patches of various textiles, each acquired at different poses and deformed shape situations. For each scan, three processing operations, designed to alter the mesh connectivity, have been applied to obtain, respectively, meshes with 5K, 10K and 15K vertices. The database has a total of 720 samples. To evaluate our new convolution approach, we computed the meshLBP [WTBD15b] descriptor on the response of the following filters on the mesh: Gabor, Sharpen, standard Edge Detector, and Sobel. In Table 1, our convolution algorithm is evaluated against the winner of SHREC'17 competition [BTA*17]. The assessment has been done according to the competition evaluation, with Nearest Neighbor (NN), first-Tier, second-Tier and Discounted Cumulative Gain (DCG) measures. As shown in the table, all our approaches outperform the SHREC'17 best result, with the exception of the Gabor filter for 1^{st} and 2^{nd} tier. We justify such exception with the non-optimization of Gabor parameters.



(a) *Standard 2D Gabor filter* **(b)** *Gabor filter obtained on the mesh using radius quantization 3.*

Figure 3: *Visual representation of the Gabor filter responses on a 2D image and on a 3D mesh in (a) and (b), respectively. The filter has been computed using the same wavelength and orientations at* 0^o, 45^o, 90^o *and* 135^o.

Table 1: *Comparison between the mesh-LBP descriptor computed on the mesh for the proposed filters, and the best performing method at SHREC'17.*

Method	**NN**	1^{st}**-Tier**	2^{nd}**-Tier**	**DCG**
Gabor	99.72%	31.70%	44.27%	76.01%
Sharpen	99.86%	45.72%	61.17%	83.97%
EdgeDetector	99.86%	45.72%	61.17%	83.97%
Sobel	100.00%	46.65%	61.16%	83.97%
KLBO [BTA*17]	98.60%	33.30%	44.90%	75.90%

Replicating CNN operations on the mesh: Potentially, our technique can be also used to extend the idea of CNNs to a real 3D environment. One basic operation in CNNs is performed by the convolutional layer where the size of the filter is a crucial parameter. Inheriting the ordering and the linear complexity of the ORF it is possible to perform convolution at different radial resolution efficiently. As an example, Figure 4 shows the application of differential filters, while varying the filter resolution (i.e., radius).

Down-sampling, or "pooling" is another important layer in CNNs. Often placed after convolutional layers, it reduces the feature dimensionality, at the same time increasing efficiency and performance. We propose two approaches for computing pooling like operations on the mesh: (a) *ring pooling*, and (b) *first-order ring neighbors*. The first approach uses the polar structure of the ORF to down-sample information in the ring (i.e., at each ring r any pooling operand is applied ring-wise, with different azimuthal quantizations). Although the ring pooling approach is easy and fast to compute, it restricts the pooling to the facets of a same ring. With the second approach, instead, we can select the entire neighborhood of a facet, thus performing the pooling across the rings (i.e., the pooling operation is done exploiting the ORF, which also allows for different strand values). In Figure 5, we show the pooling area of radius $r = 1$ on the region displayed in Figure 5a. Using max-pooling as down-sampling operand, we can determine the value $P(f_{r,\theta})$ at a given facet $f_{r,\theta}$ and a pool area size p as:

$$P(f_{r,\theta}) = \max N_p(f_{r,\theta}) , \quad (2)$$

where $N_p(f)$ gives the neighborhood of the facet f within a radius p. This approach allows us to easily select values between different rings. In Figure 5b, 5c and 5d, we show the pooling area of radius $p = 1$ sliding at different stride rates.

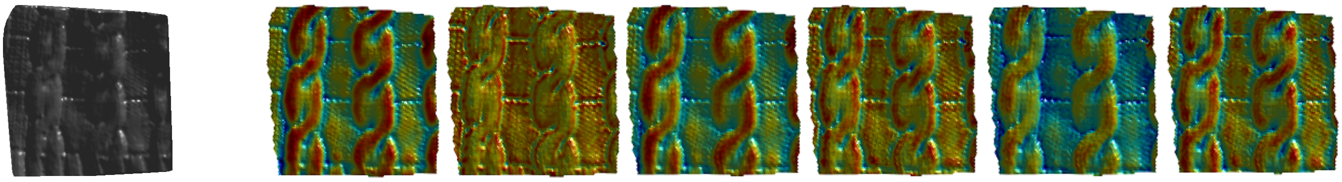
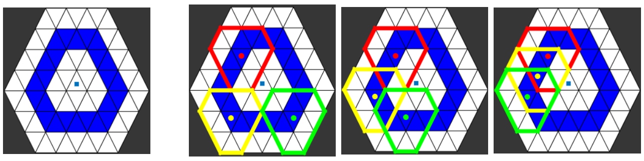


Figure 4: *Application of the differential filter at different radius on the mesh on the left using the local depth as scalar function. Radius of the mask varies from 2 to 7 ORF from left to right, while color values show the intensity of the filter response.*



(a) **(b)** **(c)** **(d)**

Figure 5: *Pooling using ORF: (a) region of radius $r = 3$ and azimuthal quantization $\theta = r \cdot 12$; referring to Eq. (2), plots (b) to (d) show a pooling area of radius $p = 1$, sliding at a strand rate of, respectively, 6, 4, and 2.*

5. Conclusions and Perspectives

In this paper, we have proposed an original framework for computing convolution-like operations on the mesh domain. To the best of our knowledge this is the first attempt of extending, in an effective and easy way, the 2D computational framework defined on the image grid to the vertices and facets of the mesh. One interesting aspect of our proposal is that it keeps most of the computational structure of the 2D domain (shift, filter mask, etc.) thus making quite straightforward the extension to the mesh of common and consolidated practices. This is clearly illustrated by the possibility to replicate Sobel, Sharpen, and Gabor-like filters. Experiments performed on the 3D relieves used in the SHREC'17 contest show that applying mesh-LBP descriptors on 3D meshes convolved with different filter masks (i.e., Sharpen, Sobel, etc.) results into performance measures that outperform state-of-the-art solutions. Finally, we provide the evidence that our proposed framework allows convolution and pooling operations on the mesh using the same setting as used in CNNs when applied to the image domain. This opens the way to the direct extension of CNN architectures to the mesh support. This will be part of our future work.

References

[BBL*17] Bronstein M., Bruna J., LeCun Y., Szlam A., Vandergheynst P.: Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine 34*, 4 (2017), 18–42. 1

[BMM*15] Boscaini D., Masci J., Melzi S., Bronstein M. M., Castellani U., Vandergheynst P.: Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In *Eurographics Symposium on Geometry Processing* (2015), Eurographics Association, pp. 13–23. 2

[BTA*17] Biasotti S., Thompson E. M., Aono M., Hamza A. B., Bustos B., Dong S., Du B., Fehri A., Li H., Limberger F. A., Masoumi M., Rezaei M., Sipiran I., Sun L., Tatsuma A., Forero S. V., Wilson R. C., Wu Y., Zhang J., Zhao T., Fornasa F., Giachetti A., Velasco-Forero S., Wilson R. C., Wu Y., Zhang Y., Zhao T., Formasa F., Giachetti A.: Shrec'17 Track: Retrieval of surfaces with similar relief patterns. In *Eurographics Work. on 3D Object Retrieval* (2017), The Eurographics Association. 3

[BZSL14] Bruna J., Zaremba W., Szlam A., LeCun Y.: Spectral networks and locally connected networks on graphs. In *Int. Conf. on Learning Representations* (April 2014), pp. 1–14. 2

[DBV16] Defferrard M., Bresson X., Vandergheynst P.: Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems* (2016), pp. 1–9. 2

[FXD*15] Fang Y., Xie J., Dai G., Wang M., Zhu F., Xu T., Wong E.: 3d deep shape descriptor. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2015), pp. 2319–2328. 2

[KW17] Kipf T. N., Welling M.: Semi-supervised classification with graph convolutional networks. In *Int. Conf. on Learning Representations* (2017), pp. 1–14. 2

[LBH15] LeCun Y., Bengio Y., Hinton G.: Deep learning. *Nature 521* (May 2015), 436–444. 1

[MBBV15] Masci J., Boscaini D., Bronstein M. M., Vandergheynst P.: Geodesic convolutional neural networks on riemannian manifolds. In *IEEE Int. Conf. on Computer Vision Workshop* (2015), pp. 832–840. 2

[QSMG17] Qi C. R., Su H., Mo K., Guibas L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2017), vol. 1, pp. 652–660. 2

[SBR16] Sinha A., Bai J., Ramani K.: Deep learning 3d shape surfaces using geometry images. In *European Conf. on Computer Vision* (2016), pp. 223–240. 2

[SPP17] Seong S., Pae C., Park H.: Geometric convolutional neural network for analyzing surface-based neuroimaging data. *CoRR abs/1708.00587* (2017). 2

[WLG*17] Wang P.-S., Liu Y., Guo Y.-X., Sun C.-Y., Tong X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph. 36*, 4 (July 2017), 72:1–72:11. 2

[WRK12] Werghi N., Rahayem M., Kjellander J.: An ordered topological representation of 3D triangular mesh facial surface: concept and applications. *EURASIP Journal on Advances in Signal Processing 2012*, 1 (2012), 144. 1, 2

[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 *IEEE Conf. on Computer Vision and Pattern Recognition* (2015), pp. 1912–1920. 2

[WTBD15a] Werghi N., Tortorici C., Berretti S., Del Bimbo A.: Computing local binary patterns on mesh manifolds for 3D texture retrieval. *Eurographics Work. on 3D Object Retrieval* (2015), 91–94. 2

[WTBD15b] Werghi N., Tortorici C., Berretti S., Del Bimbo A.: Local binary patterns on triangular meshes: Concept and applications. *Comp. Vis. and Im. Understanding 139* (2015), 161–177. 2, 3

[XFZW15] Xie J., Fang Y., Zhu F., Wong E.: Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2015), pp. 1275–1283. 2