Deep Learning 3D Shape Surfaces Using Geometry Images

Ayan Sinha[1(✉)], Jing Bai[2], and Karthik Ramani[1]

[1] Purdue University, West Lafayette, USA
{sinha12,ramani}@purdue.edu
[2] Beifang University of Nationalities, Yinchuan, China
bai58@purdue.edu

Abstract. Surfaces serve as a natural parametrization to 3D shapes. Learning surfaces using convolutional neural networks (CNNs) is a challenging task. Current paradigms to tackle this challenge are to either adapt the convolutional filters to operate on surfaces, learn spectral descriptors defined by the Laplace-Beltrami operator, or to drop surfaces altogether in lieu of voxelized inputs. Here we adopt an approach of converting the 3D shape into a *'geometry image'* so that standard CNNs can directly be used to learn 3D shapes. We qualitatively and quantitatively validate that creating geometry images using authalic parametrization on a spherical domain is suitable for robust learning of 3D shape surfaces. This spherically parameterized shape is then projected and cut to convert the original 3D shape into a flat and regular geometry image. We propose a way to implicitly learn the topology and structure of 3D shapes using geometry images encoded with suitable features. We show the efficacy of our approach to learn 3D shape surfaces for classification and retrieval tasks on non-rigid and rigid shape datasets.

Keywords: Deep learning · 3D Shape · Surfaces · CNN · Geometry images

1 Introduction

The ground-breaking accuracy obtained by convolutional neural networks (CNNs) for image classification [16] marked the advent of deep learning methods for various vision tasks such as video recognition, human and hand pose tracking using 3D sensors, image segmentation and retrieval [9,13,27]. Researchers have tried to adapt the CNN architecture for 3D non-rigid as well as rigid shape analysis.

The lack of a unified shape representation has led researchers pursuing deformable and rigid shape analysis using deep learning down different routes. One strategy for learning rigid shapes is to represent a shape as a probability

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46466-4_14) contains supplementary material, which is available to authorized users.

© Springer International Publishing AG 2016
B. Leibe et al. (Eds.): ECCV 2016, Part VI, LNCS 9910, pp. 223–240, 2016.
DOI: 10.1007/978-3-319-46466-4_14

distribution on a 3D voxel grid [20,32]. Other approaches quantify some measure of local or global variation of surface coordinates relative to a fixed frame of reference [26]. These representations based on voxels or surface coordinates are *extrinsic* to the shape, and can successfully learn shapes for classification or retrieval tasks under rigid transformations (rotations, translations and reflections). However, they will naturally fail to recognize isometric deformation of a shape, say the deformation of a standing person to a sitting person. Invariance to isometry is a necessary property for robust non-rigid shape analysis. This is substantiated by the popularity of the *intrinsic* shape signatures for 3D deformable shape analysis in the geometry community [31]. Hence, CNN-based deformable shape analysis methods propose the use of geodesic convolutional filters as patches or model spectral-CNN's using the eigen decomposition of the Laplace-Beltrami operator to derive robust shape descriptors [1,6,19]. In summary, the vision community has focussed on extrinsic representation of 3D shapes suitable for learning rigid shapes, whereas the geometry community has focussed on adapting CNN's to non-Euclidean manifolds using intrinsic shape properties for creating optimal descriptors. A method to unify these two complementary approaches has remained elusive.

Here we propose a 3D shape representation that serves to learn rigid as well as non-rigid objects using intrinsic or extrinsic descriptors input to standard CNNs. Instead of adapting the CNN architecture to support convolution on surfaces, we adopt the alternate approach of molding the 3D shape surface to fit a planar structure as required by CNNs. The traditional approach to create a planar surface parametrization is to first cut the surface into disk-like charts, then piecewise parameterize them in the plane followed by stitching them together into a texture atlas [18]. This approach fails to preserve the connectivity between different surfaces, vital for holistic shape analysis. In contrast, we create a planar parametrization by introducing a method to transform a general mesh model into a flat and completely regular 2D grid, which we term 'geometry image', following

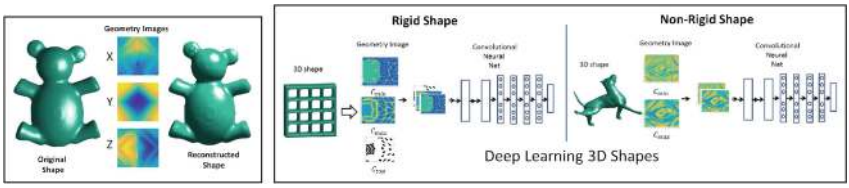

Fig. 1. Left Shape representation using geometry images: The original teddy model to the left is reconstructed (right) using geometry image representation corresponding to the X, Y and Z coordinates (center), **Right** Learning 3D shape surfaces using geometry images: Our approach to learn shapes using geometry images is applicable to rigid (left) as well as non-rigid objects undergoing isometric transformations (right). The geometry image encode local properties of shape surfaces such as principal curvatures (C_{min}, C_{max}). Topology of a non-zero genus surface is accounted for by using a topological mask (C_{top}) as in the bookshelf example.

[11] (see Fig. 1 left). The traditional approach to create a geometry image has critical limitations for learning 3D shape surfaces (see Sect. 2). We validate that an intermediate shape representation for creating geometry images in the form of an authalic parametrization on a spherical domain overcomes these limitations and is able to efficiently learn 3D shape surfaces for subsequent analysis. To this end, we develop a robust method for authalic spherical parametrization applicable to general 3D shapes. We use this parametrization to encode suitable intrinsic or extrinsic features of a 3D shape for 3D shape tasks. This encoded spherical parametrization is converted to a completely regular geometry image of a desired size. We demonstrate the use of these geometry images to directly learn shapes using a standard CNN architecture to classify and retrieve shapes. In summary our main contributions are: (1) robust authalic parametrization of general 3D shapes for creating geometry images, and (2) a procedure to learn 3D surfaces using a geometry image representation which encodes suitable features for rigid or non-rigid shape tasks (see Fig. 1 right).

Our article is organized as follows. Section 2 rationalizes our choice of parametrization. Section 3 discusses our parametrization method. Section 4 is devoted to learning shapes using geometry images and CNNs followed by results in Sect. 5.

2 Frame of Reference and Related Work

In this section we first validate that authalic parametrization on a spherical domain has key advantages over alternate surface parametrization techniques in the context of learning shapes using geometry images. We briefly overview existing techniques and point the readers to [7] for a good overview of surface parametrization.

Why spherical parametrization?: Geometry images as the name suggests are a particular kind of surface parametrization wherein the geometry is resampled into a regular 2D grid akin to an image. Geometry images are advantageous for learning shapes using CNNs over free boundary or disc parameterizations as every pixel encodes desired shape information. This reduces memory and learning complexity in CNNs as the need to abstract the mask of inside/outside shape boundary is obviated. The traditional approach to create a geometry image is to cut the surface into a disc using a network of cut paths and then map the disc boundary to a square [11]. However, defining consistent *a priori* cuts over a range of shapes in a class is a hard problem. A natural solution to overcome this limitation is a data-driven approach to learn a shape over several cuts. This is computationally inefficient for cuts defined *a priori*. Another assumption of [11] is that the surface cut into a disc maps well onto a square. Different cuts lead to variation in geometry image boundaries [22], and hence, learning them requires the CNN to learn maps between image boundaries in addition to image pixels. These two limitations of traditional geometry images are overcome by geometry images created by first parameterizing a 3D shape over a spherical domain, then sampling onto an octahedron and finally cutting the octahedron along its edges to output a flat and regular geometry image. This is because: (1) Cuts are

defined *a posteriori* to the parametrization. This enables us to efficiently create many geometry images for a given shape by sampling several cuts and feed it as input to data driven learning techniques such as CNNs. (2) Spherical symmetry allows creating a regular geometry image boundaries without discontinuities. The symmetry enables us to implicitly inform the CNN that the geometry image is derived from a spherical domain via padding. Although spherical parametrization is only applicable to genus zero surfaces, we propose a heuristic extension to higher genus surface models using a topological mask.

Why authalic parametrization?: There are two strategies for spherical parametrization of a 3D shape: (a) Authalic or area conserving, (b) Conformal or angle conserving. Although, methods for conformal (angle preserving) mesh parametrization abound [4,12,25], there is relatively less work on authalic (area preserving) mesh parametrization. This is because a conformal parametrization preserves local shape, which is useful to the graphics community for feature oriented applications such as texture mapping. However, an authalic parametrization of a shape is more compatible with the notion of convolving surface patches with constant size (equi-areal) filters. Also, conformal parametrization induces severe distortion to elongated shape structures common in deformable shape models [34]. The necessity of authalic parametrization arises from the fact that the number of training samples and learning parameters in the CNN sometimes limit the input resolution of the geometry images. Under the constraint of resolution, authalic geometry images encode more information about the shape as compared to conformal geometry images (see Fig. 2). Note that a mapping that is both conformal and authalic is isometric, and must have zero Gaussian curvature everywhere. This is rare in the context of general 3D mesh models and one must choose one or the other. There exist only a handful of methods in literature that authalically parameterize a shape on a spherical domain. Dominitz and Tannenbaum [5] and Zhao et al. [34] use optimal transport for area-preserving mapping. Although efficient to

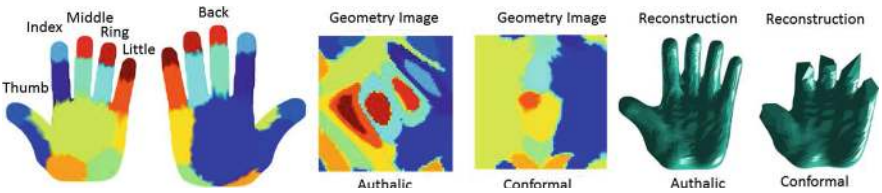

Fig. 2. Authalic vs Conformal parametrization: (Left to right) 2500 vertices of the hand mesh are color coded in the first two plots. A 64× 64 geometry image is created by uniformly sampling a parametrization, and then interpolating the nearby feature values. Authalic geometry image encodes all tip features. Conformal parametrization compress high curvature points to dense regions [12]. Hence, finger tips are all mapped to a very small regions. The fourth plot shows that the resolution of geometry image is insufficient to capture the tip feature colors in conformal parametrization. This is validated by reconstructing shape from geometry images encoding x, y, z locations for both parameterizations in final two plots. (Color figure online)

implement, these methods introduce smoothing and sharp edges get lost [29]. This is a critical drawback for CAD-like objects which contain several sharp edges. A method that implicitly corrects area distortion by penalizing large triangle sizes is proposed in [8]. However, our experiments indicate that this approach fails to work in a practical setting. A method similar in spirit to ours uses Lie advection to iteratively minimize the planar areal distortion of a parametrization [35]. However, the method frequently introduces singularities and triangle flips, highly undesirable for coherent 3D shape representation and analysis.

Why geometry images?: As discussed previously, current methods employing deep learning for 3D rigid shape analysis such as ShapeNets [32], VoxNet [20], DeepPano [26] are extrinsic representations and are not suitable for analyzing non-rigid shapes undergoing isometric deformations. Another bottleneck in voxel based approaches is that the 3rd extra dimension introduces a large computational overhead. Consequently, the voxel grid is restricted to a relatively low resolution. Also, active voxels interior to the shape are less useful if the boundary surface is well defined. Methods using CNN for 3D non-rigid shape analysis such as [1,19] focus on deriving robust shape descriptors suitable for local shape correspondence. The potential of CNN's to automatically learn hierarchical abstractions of a shape from raw input features is not realized by these approaches. In contrast to all approaches, the pixels in geometry images can encode either extrinsic or intrinsic surface property as suitable for the task at hand. A standard CNN then automatically learn discriminative abstractions of the 3D shape, useful for shape classification or retrieval.

3 Authalic Parametrization of 3D Shapes

We briefly discuss preprocessing steps to transform erroneous or high genus mesh models into a genus zero topology. These steps ensure that parametrization techniques from discrete differential geometry literature are applicable to a shape of arbitrary topology. A surface mesh, M is represented as V, F, E wherein V is the set of vertex coordinates, F the set of faces and E the set of edges constituting all faces. With abuse of notation, we term mesh models following the Euler characteristic to be accurate, given by:

$$2 - 2m = |V| - |E| + |F| \tag{1}$$

where $|x|$ indicates the cardinality of feature x and m is the genus of the surface. If a mesh model is not accurate, a heuristic but accurate procedure is discussed in the supplementary material to transform it into an accurate mesh. In our experiments we perform this procedure only for models in the Princeton ModelNet [32] benchmark. If the genus of an accurate mesh model is evaluated to be non-zero, we propose another heuristic in the supplementary material to convert the mesh into a genus-0 surface. This genus-0 shape serves as input to the authalic parametrization procedure. Note that a non genus-0 shape has an associated topological geometry image informing the holes in the original shape.

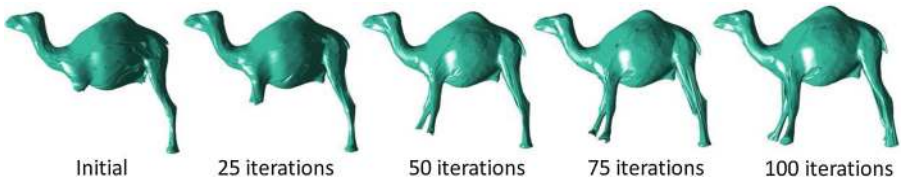

Fig. 3. Progression of our authalic spherical parametrization algorithm: Individual plots display the shape reconstructed from the geometry image corresponding to a spherical parametrization. The area distortion associated with the geometry image, and hence the spherical parametrization, progressively decreases with more iterations given an initial spherical parametrization.

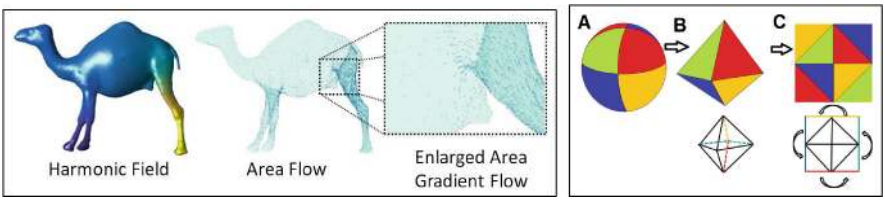

Fig. 4. Left Left: Harmonic field corresponding to area distortion on sphere displayed on the original mesh. Center: Area restoring flow on the spherical domain mapped onto the original mesh as a quiver plot. Right: Enlarged plot of area restoring flow. **Right**: Explanation of geometry image construction from a spherical parametrization: The spherical parametrization (A) is mapped onto an octahedron (B) and then cut along edges (4 colored dashed edges in line plot below) to output a flat geometry image (C). The colored edges share the same color coding as the one in the octahedron. Also the half-edges on either side of the midpoint of colored edges correspond to the same edge of the octahedron. (Color figure online)

Our method for authalic spherical parametrization takes as input any spherically parameterized mesh and iteratively minimizes the areal distortion (see Fig. 3) in 3 steps described in detail below and outputs a bijective map onto the surface of a sphere. We use the spherical parametrization suggested in [10] for initialization due to its speed and ease of implementation. We evaluated different initial parameterizations [25] and our experiments indicate that our method is robust to initialization. We now detail the 3 steps:

(1) At every iteration we first evaluate a scalar harmonic field corresponding to the areal distortion ratio of vertices in the original mesh and spherical mesh by solving a Poisson equation. Mathematically, we solve

$$\nabla^2 g = \delta h \tag{2}$$

where g is a function defined on the vertex set V, ∇^2 transforms to the Laplacian operator, L (see supplement) for a closed mesh surface [14], and δh is the areal distortion ratio wherein each element of the vector is defined

as $\delta h_u = \frac{A_u^s}{A_u} - 1$. A_u^s is the spherical triangular area associated with the Voronoi region around vertex u and A_u is the triangular area associated with vertex u on the mesh model. Equation 2 now becomes

$$Lg = \delta h \tag{3}$$

The scalar field g is evaluated using the above equation at every iteration for the vector δh (see Fig. 4 left). Due to the sparsity of L, Eq. 3 can be efficient evaluated at every iteration using the preconditioned bi-conjugate gradient method. However, we precalculate the pseudoinverse of L once, and use it for every iteration. This saves the overall computational time. Note, k-rank approximation ($k \approx 300$) of the pseudoinverse when $|V|$ is large does not noticeably affect the final result.

(2) We then evaluate the gradient field of the harmonic function on the original mesh. This field is indicative of the required vertex displacements on the spherical mesh so as to decrease the areal distortion ratio. Consider a face f_{uvw} in the original mesh with its three corners lying at u, v, w. Let n be a unit normal vector perpendicular to the plane of the triangle. The gradient vector ∇g for each face is solved as [33]:

$$\begin{bmatrix} v-u \\ w-v \\ n \end{bmatrix} \nabla g = \begin{bmatrix} g_v - g_u \\ g_w - g_v \\ 0 \end{bmatrix}$$

A unique gradient vector for each vertex is obtained as weighted mean of incident angle of each face at the vertex and the corresponding gradient value as done in [35]:

$$\nabla g_u = \frac{1}{\sum_{f_{uvw}} c_{vw}^u} \sum_{f_{uvw}} c_{vw}^u \nabla g(f_{uvw}) \tag{4}$$

f_{uvw} are the faces in the one ring neighborhood of vertex u and c_{vw}^u is the angle subtended at vertex u by the edge vw. Figure 4 shows the gradient low field using a quiver plot on the mesh model.

(3) We finally displace the vertices on the original mesh and then map these displacements onto the spherical mesh using barycentric mapping, i.e., vertex displacements on the original mesh serve as proxy to determine the corresponding displacements on the spherical mesh. Barycentric mapping is possible because the original and spherical mesh have the same triangulation. Each vertex in the original mesh is (hypothetically) displaced by:

$$v = v + \rho \nabla g_v \tag{5}$$

where ρ is a small parameter value. A large value of ρ leads to a large displacement of the vertex and may displace it beyond the its 1-neighborhood. This causes triangle flips and the error propagates through iterations. However, a small value of ρ leads to large convergence time. We empirically

set ρ equal to 0.01 in all our experiments which achieves the right tradeoff between number of iterations to convergence and accuracy. The barycentric coordinates of displaced vertices are evaluated with respect to triangles in the one-ring, and the triangle with all coordinates less than 1 is naturally chosen as the destination face. The vertex in the spherical mesh is then mapped to the corresponding destination face with the same barycentric weights. In contrast to [35] which operates directly on the spherical mesh domain, the indirect mapping procedure has the following advantages: (1) The vertex displacements minimizing areal distortion are constrained to be on the input mesh, which in turn ensure the mapped displacements onto the spherical domain are well behaved. (2) The constraint that the vertices remain on the mesh model minimize triangle flips and alleviate the need for an expensive retriangulation procedure after each iteration. The iterations continue until convergence. In practice we stop the iterations after the all areal distortion ratios fall below a threshold or the maximum number of iterations has been reached. The maximum number of iterations is set to 100. Supplementary material provides a pseudo code of the above procedure and MATLAB code for creating geometry images are available at: https://github.com/sinhayan/learning_geometry_images. Next, we discuss the geometry image and its application to deep learning.

4 Deep Learning Shapes Using Geometry Image

In this section we briefly discuss the creation of a geometry image with desirable surface properties encoded in the pixels to learn 3D shapes. We also discuss our CNN architecture for shape classification and retrieval.

4.1 Geometry Image and Descriptors

The spherical parametrization maps the surface of the mesh onto a sphere. We then project this spherical surface onto an octahedron and cut it to obtain a square, thus creating a geometry image. We consider spherical triangular area when sampling from sphere to octahedron, so that the authalic parametrization is respected, and hence, the areas are preserved after projection onto a octahedron. The advantage of mapping the surface onto an octahedron over other regular polyhedra such as a tetrahedron or cube is that the signals can be linearly interpolated onto a regular square grid [22]. For brevity, we skip details on the spherical area sampling for projecting points on the sphere onto an octahedron and refer readers to [22] for details. The edges of the octahedron cut to flatten the polyhedron are shown in Fig. 4 right. Observe the reflective symmetry of the geometry image along the vertical, horizontal and diagonal axes shown in Fig. 4 right. Due to this symmetry, we can create replicates without any discontinuities along any edge or corner of the image (see Fig. 5 right). This property is useful for implicitly informing a deep learning model about the warped mesh the image represents, further explained below. The procedure of creating the

geometry image is visually elucidated in Fig. 4 right. Additionally, a MATLAB implementation is provided in supplementary material. Having obtained a geometry image from a mesh model, we next discuss encoding the pixel values with local surface property descriptors. There exist several possibilities of which we enumerate a few:

1. Principal curvatures: The two principal curvatures, κ_1 and κ_2 measure the degree by which the surface bends in orthogonal directions at a point. They are in effect the eigenvalues of the shape tensor at a given point.
2. Gaussian Curvature: The Gaussian curvature κ is defined as the product of the principal curvatures at a point on the surface, $\kappa = \kappa_1\kappa_2$. Gaussian curvature is an intrinsic descriptor. The sign of Gaussian curvature indicates whether a point is elliptic ($\kappa > 0$), hyperbolic ($\kappa < 0$) or flat ($\kappa = 0$)
3. Heat kernel signature [31]: The heat kernel, h_t is the solution to the differential equation $\frac{\delta h_t}{\delta t} = -\Delta h_t$ (h_t is the heat kernel). The heat kernel signature (HKS) at the point is the amount of untransferred heat after time t, given by

$$h_t(u,u) = \sum_{i \geq 0} e^{-t\lambda_i} \Phi_i(u)\Phi_i(u) \tag{6}$$

Where λ and Φ are the eigenvalues and eigenvectors of the Laplace-Beltrami operator. The heat kernel is invariant under isometric transformations and stable under small perturbations to the isometry, such as small topological changes or noise, i.e., is intrinsic. Additionally, the time parameter t in the HKS controls the scale of the signature with large t representing increasingly global properties, i.e. its a multiscale signature. Variants of the heat kernel include the GMS [28], GPS [23]which differ in the weighting of the eigenvalues. Figure 5 left discusses the difference between intrinsic HKS and point coordinates which are extrinsic in the context of analyzing articulated shapes. The invariance of intrinsic descriptors to articulations of a deformable object such as a hand is further demonstrated in Fig. 5 center. In our experiments we use the HKS for non-rigid shape analysis and the two principal curvatures for rigid-shape analysis.

4.2 Convolutional Neural Net

We discuss four aspects of learning rigid and non-rigid shapes using geometry images created using the authalic parametrization method discussed in the previous section as input to a CNN, i.e., encoding a property, padding the image, robustness to cut and the CNN architecture which takes geometry images as inputs and performs shape analysis tasks.

(1) **Encoded Property:** After parameterizing the shape, we are interested in encoding the geometry image with a suitable property. These are the RGB pixel values in images which are fed as input to a CNN. Unlike traditional deep architectures, CNN's have the attractive property of weight sharing reducing the number of variables to be learned. The principle of weight sharing in convolutional filters extensively applied to image processing is

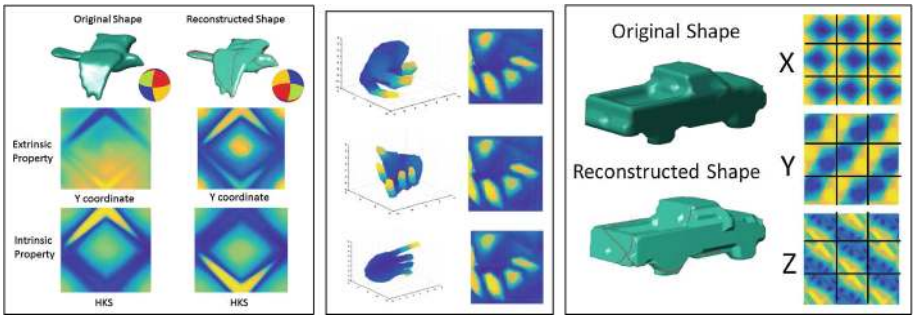

Fig. 5. Left Intrinsic vs. Extrinsic properties of shapes. Top left: Original shape. Top Right: Reconstructed shape from geometry image with cut edges displayed in red. The middle and bottom rows show the geometry image encoding the y coordinates and HKS, respectively of two spherical parameterizations (left and right). The two spherical parameterizations are symmetrically rotated by 180 degrees along the Y-axis. The geometry images for Y-coordinate display an axial as well as intensity flip. Whereas, the geometry images for HKS only display an axial flip. This is because HKS is an intrinsic shape signature (geodesics are persevered) whereas point coordinates on a shape surface are not. **Center** Intrinsic descriptors (here the HKS) are invariant to shape articulations. **Right** Padding structure of geometry images: The geometry images for the 3 coordinates are replicated to produce a 3× 3 grid. The center image in each grid corresponds to the original geometry image. Observe no discontinuities exist along the grid edges.

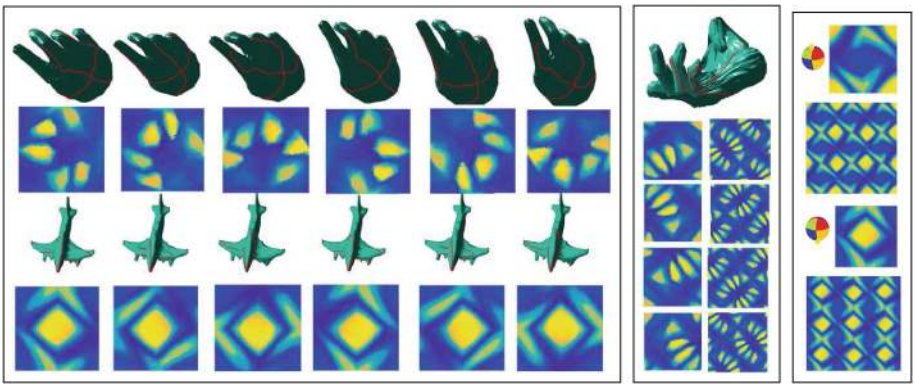

Fig. 6. Left Geometry images created by fixing the polar axis of a hand (top) and aeroplane (bottom), and rotating the spherical parametrization by equal intervals along the axis. The cut is highlighted in red. **Center** Four rotated geometry images for a different cut location highlighted in red. The plots to the right show padded geometry images wherein the similarity across rotated geometry images are more evident and the five finger features coherently visible **Right** Changing the viewing direction for a cut inverts the geometry image. The similarity in geometry images for the two diametrically opposite cuts emerges when we pad the image in a 3×3 grid (Color figure online)

applicable to learning 3D shapes using geometry images as well. This is because shapes like images are composed of atomic features and have a natural notion of hierarchy. However, we encode different features in the pixels of the geometry image for rigid and non-rigid shapes as it helps a CNN to discriminatively learn shape surfaces. The Gaussian curvature is the most atomic and intrinsic property suitable for non-rigid shape analysis. The heat kernel signature too can be interpreted as an extension to gaussian curvature [31]. We use the HKS for our experiments on non-rigid datasets as it enforces long-range consistency to geometry images. In rigid shape analysis, the principal curvatures serve as the atomic local descriptors for points on a surface. Although, the intrinsic HKS can be used for rigid shape analysis, HKS has a high computational cost unsuitable for large datasets like the Princeton Shape Benchmark.

(2) **Padding:** We now have a geometry image with a suitably encoded property. It is naturally beneficial to inform the CNN that this flat geometry image stems from a compact manifold. The spherical symmetry of our parametrization allows us to implicitly inform the CNN about the genus-0 surface via padding. There are no edge and corner discontinuities if we connect replicates of a geometry image along each of the 4 edges of the image which are rotated by 180 degrees (or flipped once along the x-axis and y-axis each). This is due to spherical symmetry and orientation of edges in the derived octahedral parametrization. This is visually illustrated for the geometry images encoding the x, y, and z coordinates of the mesh model in Fig. 5 right. No subsequent layer in the CNN is padded so as to not distort this information.

(3) **Cut:** Recall that the octahedral edges cut to create a geometry image are dependent on the orientation of the spherical parametrization. We implicitly inform the CNN that different cuts resulting in slightly different geometry images stem from the same shape. When the shape is known to be upright as in the Princeton shape benchmark, we realign north pole of the derived spherical parametrization to be coincident with the highest point along the centroid axis to make the north pole to be approximately co-located for the same class of shapes. The directed axis connecting the north and south pole can be thought of as a viewing direction of the sphere, and hence the geometry image. Rotation around this polar axis of the sphere will result in different cuts of the octahedron and hence slightly different geometry images which are rotationally related. This rotational relationship between geometry images for the same object is learnt by rotating the spherical parametrization in equal intervals about the polar axis for a shape (see Fig. 6 left). This is analogous to the procedure of augmenting data by rotation along the gravity direction as done in voxel based approaches such as [20,30,32] to create models in arbitrary poses, and hence, remove pose ambiguity. The rotational variance along the polar axis for geometry images of upright objects can be further resolved by incorporating an additional feature map in CNN architecture as the geometry image encoding the angle between a vertex normal and the gravity direction [9]. When there is no information about

orientation of the shape, we naturally set multiple radial axes of the sphere to be the directed polar axes (we set six orthogonal directed axes of the sphere to be the polar axes in our experiments with non-rigid datasets) and then rotate the sphere by equal intervals along each polar axes to holistically augment the training data along different viewing directions of the spherical parametrization. Figure 6 left and center show the rotated geometry images for an articulated hand for two different polar cutting axes. Observe that although the geometry images appear very different for the two cuts, they are functionally related as they are just projections along different viewing directions of spherical parametrization onto the flat geometry image. For example there are 5 primary features in both geometry images corresponding to the 5 fingers and their relative locations are similar in both images. The mild stretch variations among geometry images would not appear if the parametrization was isometric. Indeed, the accuracy of our approach stems from the power of CNNs to automatically abstract these similarity in patterns robust against different cut locations in the augmented data across articulations of a deformable object or variations of objects in a class.

(4) **Resolution and architecture:** There are two determining factors for the resolution of a geometry image: (i) The number of training samples (ii) Features in the mesh model. Currently there are no large databases for non-rigid shapes, and hence, a large resolution will lead to a large number of weight parameters to be learnt in the CNN. Although we have large databases for rigid shapes, the number of geometry features (eg. protrusion, corners etc.) in rigid shapes is typically much lower compared to images and even articulated objects. We set the size of the geometry image to be 56×56 for all our experiments on rigid and non-rigid datasets which balances the number of weights to be learnt in CNN and capturing relevant features of a mesh model. The number of layers in CNN is determined by the size of the training database. Hence, we choose a relatively shallow architecture for non-rigid database compared to the rigid database. The precise architecture of the CNNs are discussed in the supplementary section.

5 Experiments

In this section we first compare our parametrization scheme. Then we discuss results for 3D shape analysis tasks on rigid as well as non-rigid datasets.

Authalic parametrization: We compare our authalic spherical parametrization scheme to other area correcting methods. We qualitatively adjudge the parametrization in terms of the geometry image created from the corresponding spherical parameterizations on some prototypical meshes. The methods compared to are the lie advection based method in [35], and the penalty-term based method proposed in [8], both of which are iterative methods. For fair comparison, the maximum number of iterations was fixed to 100 for all methods along with suggested parameter settings. Figure 7 left shows the comparison. We observe that our method is the only method to consistently complete the shape while

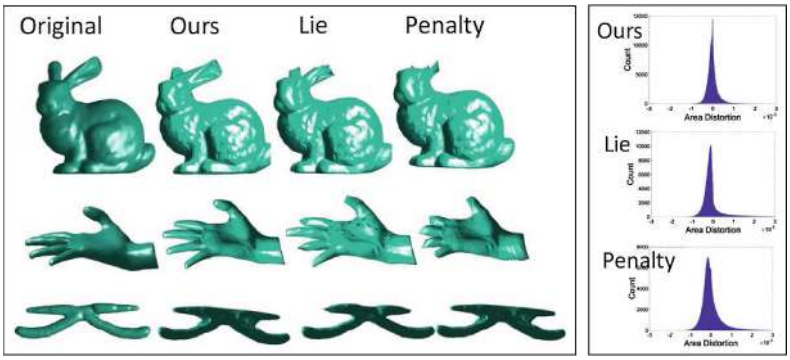

Fig. 7. Left Comparison of authalic surface parametrization methods in terms of shape reconstruction using geometry image. Left to Right: Original mesh model, Our authalic parametrization, Lie advection based method in [35], Penalty based method proposed in [8]. **Right** Top to Bottom: Area distortion viewed as a histogram over triangles for ours, [8,35].

keeping extraneous noise at a minimum. For example no method apart from ours is able to complete the bunny's ears or completely reveal all 5 fingers. This validates our approach in the context of geometry image creation and authalic spherical parametrization in general. Next we quantitatively evaluate the accuracy of our authalic parametrization by comparing the area distortion across all triangles in all 148 shapes in TOSCA database. The distortion metric is δhA. Figure 7 right shows area distortion as a histogram as done in [35]. A perfect authalic parametrization would manifest as a delta function in this plot. Hence we evaluate the variance of these three approaches. Observe that our method has the sharpest peak and the variance is evaluated to be 9.8e-8 for our method compared to 5.2e-7 for [35] and 2.65e-7 for [8], i.e., lowest among all.

Non-rigid shapes: We evaluated our approach for surface based intrinsic learning of shapes on two datasets. We used 200 shapes from the McGill 3D shape benchmark consisting of articulated as well as non-articulated shapes from 10 classes (20 in each class). To test the robustness of our approach, we also evaluated our approach on the challenging SHREC-11 [17] database of watertight meshes consisting of 20 shapes from 30 classes (600 in total). For each of the 2 databases, we performed classification tasks on 2 splits: (1) 10 randomly chosen shapes from each class were used for training and 10 were test (2) 16 randomly chosen shapes were in the train set and the rest were test cases. Due to the small size of the database, we kept our CNN relatively shallow (3 convolutional, 1 fully connected layer and a classification layer) so as to limit the number of training parameters. We augment the data in order to be robust to cut location by inputting 36 geometry images for a shape created by (1) fixing the six directed intersections of the three orthogonal coordinate axes with the spherical parametrization as the polar axes and then (2) creating a geometry image for each incremental rotation of the sphere along the polar axes by 60 degrees

starting from 0 to cover a full 360 degrees. Images of size 56×56 were padded as described in Sect. 4.2 to produce a 64×64 image as input to the CNN. For features, we used HKS sampled at 5 logarithmically sampled time scales to produce a 5 dimensional feature map. Due to the small training sample, the CNN using only gaussian curvature failed to converge. CNNs using principal curvatures naturally failed to converge as the principal curvatures are not intrinsic properties for non-rigid shapes. Training using the HKS features converged after 30 epochs. We compare our approach to 4 other methods: ShapeGoogle (SG)[2], Zerkine moments (Zer) [21], Light Field Descriptor (LFD) and 3DShapeNets (SN) [32] for classification and retrieval. A class was assigned to each shape in our method by simply pooling predictions from the softmax layer over the 36 views and then selecting the one with the highest overall score. Multi-view CNN architecture [30] can be directly employed for a more principled way to pool and learn across different cuts within the CNN architecture itself, which we wish to investigate in the future when larger non-rigid databases are available. We trained a linear SVM classifier for SG, LFD and Zer methods.[1] We see that our method significantly outperforms all other methods on both splits for the 2 databases (Table 1) indicating that our geometry image representation was able to *learn* the shape structure of each class. Our method performs significantly better than SN [32] on these benchmarks because voxels capture extrinsic shape information, and hence, confuse shape articulations. It performs better than SG [2] because of the same reason that CNNs outperform bag of feature (BOF) based approaches on image tasks, i.e., CNNs are better able to automatically abstract relevant information for tasks than BOFs. We also quantitatively validate that authalic parametrization is more suitable for shape analysis compared to conformal (Conf) parametrization [12] or Spharm (Sph) [24] which minimizes length distortion. Performance of authalic parametrization is a lot higher than others for non-rigid shapes, as expected because the other two parameterizations do not robustly capture elongated protrusions. We use the L2 distance to measure the similarity between all pairs of testing samples and retrieval accuracy was measured in terms of mean average precision (MAP) as standard in literature. The penultimate 48-dimensional activation vector in the fully connected layer was used for measuring the retrieval accuracy of our method. We perform best in all but one dataset, i.e., 2nd to SG for SHREC2, inspite our feature vector being $1/50^{th}$ the size of SG. This highlights that our method can be used to output highly informative shape signatures. Figure 8 shows precision-recall curves for the 4 splits.

Rigid shapes: We evaluate our approach for surface-based learning of 3D shape classification on the two versions of the large scale Princeton ModelNet dataset: ModelNet40 and ModelNet10 consisting of 40 and 10 classes respectively following the protocol of [32]. We use four feature maps encoded in geometry images: 2 principal curvatures, topological mask along with a height field encoded as angle to the positive gravity direction. Additionally, each spherical parametriza-

[1] Note we do not report the scores for SG on Mcgill because the author provided implementation failed on several shapes and produced spurious results.

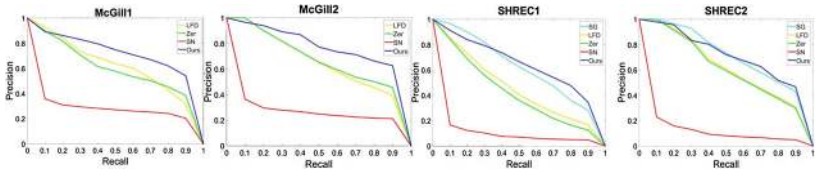

Fig. 8. Precision recall curves for shape retrieval on non-rigid datasets

Table 1. Classification/Retrieval accuracy of our method compared to 4 other methods and compared to 2 other surface parameterizations.

Database	SG [2]	Zer [21]	LFD [3]	SN [32]	Conf [12]	Sph [24]	Ours
McGill1	NA	63.0/0.64	75.0/0.67	65.0/0.29	55.0/0.36	62.0/0.35	**83.0**/**0.75**
Mcgill2	NA	57.5/0.69	72.5/0.68	57.2/0.28	80.0/0.58	82.5/0.58	**92.5**/**0.72**
SHREC1	62.6/**0.65**	43.3/0.47	56.7/0.50	52.7/0.10	60.6/0.45	59.0/0.65	**88.6**/ **0.65**
SHREC2	70.8/**0.74**	50.8/0.64	65.8/0.65	48.4/0.13	85.0/0.45	82.5/0.66	**96.6**/0.72

Table 2. Classification/Retrieval accuracies of our method on the ModelNet40 and ModelNet10 database compared to 5 other 3D learning methods and two alternate surface parameterizations.

Database	VN [20]	DP [26]	LFD [3]	SN [32]	SH [15]	Conf [12]	Sph [24]	Ours
ModelNet10	**92.0**/NA	85.5/**84.1**	79.8/49.8	83.5/69.2	79.9/45.9	78.2/67.4	79.9/65.2	88.4/74.9
ModelNet40	83.0/NA	77.6/**76.8**	75.4/40.9	77.3/49.9	68.2/34.4	75.6/46.2	75.9/44.8	**83.9**/51.3

tion is augmented by incrementally shifting by 30 degrees along the centroid axes described in Sect. 4.2 to create 12 replicates. The size and structure of the geometry image is the same as the ones used for non-rigid testing. Supplementary material validates technical parameter settings on the ModelNet10 dataset. Table 2 shows the classification accuracies (same method as non-rigid) and retrieval results (MAP %) relative to 5 methods (VN is VoxNet, DP is DeepPano, SH is spherical harmonic) and 2 alternate parameterizations. We employ the procedure in [32] to use the L2 distance between the penultimate 96-dimensional activation vectors in the fully connected layer for retrieval. We achieve the best classification accuracy on ModelNet40 dataset. Our MAP retrieval is second only to DeepPano on both splits, however our classification accuracies are higher suggesting the a panoramic representation may be more suitable for retrieval with high intra-class discrimination, whereas geometry images are highly robust for classification. Our method performs better than SN [32] on these benchmarks because (i) encoding local principal curvatures in geometry images is analogous to pixel intensities in images, which suit CNN's architecture. (ii) Learning is harder for voxel locations compared to surface properties. Indeed training required about 3 hours on the ModelNet40 benchmark compared to 2 days for SN [32].

6 Conclusion

We introduce geometry images for intrinsically learning 3D shape surfaces. Our geometry images are constructed by combining area correcting flows, spherical parameterizations and barycentric mapping. We show the potential of geometry images to flexibly encode surface properties of shapes and demonstrate its efficacy for analyzing both non-rigid and rigid shapes. Furthermore, our work serves as a general validation of surface based representations for shape understanding. In the future we wish to build upon these insights for generative modeling of 3D shapes using geometry images instead of traditional images using deep learning. We believe that deep learning using geometry images can potentially spark a closer communion between the 3D vision and geometry community.

Acknowledgements. This work was partially supported by the NSF Award No.1235232 from CMMI as well as the Donald W. Feddersen Chaired Professorship from Purdue School of Mechanical Engineering. Dr. Jing Bai was supported by the National Natural Science Foundation of China (No. 61163016) and by the China Scholarship Council. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

1. 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. Comput. Graph. Forum **34**, 13–23 (2015)
2. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. (TOG) **30**(1), 1 (2011)
3. Chen, D.-Y., Tian, X.-P., Shen, Y.-T., Ouhyoung, M.: On visual similarity based 3D model retrieval. Comput. Graph. Forum **22**(3), 223–232 (2003)
4. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. Comput. Graph. Forum **21**, (2002)
5. Dominitz, A., Tannenbaum, A.: Texture mapping via optimal mass transport. IEEE Trans. Vis. Comput. Graph. **16**(3), 419–433 (2010)
6. Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., Wong, E.: 3D deep shape descriptor. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2319–2328 (2015)
7. Floater, M.S., Hormann, K.: Surface parameterization: a tutorial and survey. Advances in Multiresolution for Geometric Modelling, pp. 157–186. Springer, Heidelberg (2005)
8. Friedel, I., Schröder, P., Desbrun, M.: Unconstrained spherical parameterization. J. Graph. Tools **12**(1), 17–26 (2007)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580–587. IEEE (2014)
10. Gotsman, C., Gu, X., Sheffer, A.: Fundamentals of spherical parameterization for 3D meshes. In: Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, 14–17 March 2006, pp. 28–29 (2003)

11. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2002, pp. 355–361. ACM, New York (2002)
12. Gu, X., et al.: Genus zero surface conformal mapping and its application to brain surface mapping. IEEE Trans. Medical Imaging (2003)
13. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of International Computer Vision and Pattern Recognition (CVPR 2014) (2014)
14. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP 2006, pp. 61–70. Eurographics Association, Aire-la-Ville (2006)
15. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3D shape descriptors, June 2003
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
17. Laga, H.T., Schreck, A., Ferreira, A., Godil, I.P., Meshes, W., Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavou, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H.: SHREC 2011 Track: shape retrieval on non-rigid 3D (2011)
18. Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. ACM Trans. Graph. **21**(3), 362–371 (2002)
19. Masci, J., Boscaini, D., Bronstein, M.M., Vandergheynst, P.: Shapenet: Convolutional neural networks on non-euclidean manifolds. arXiv preprint arXiv:1501.06297 (2015)
20. Maturana, D., Scherer, S.: Voxnet: a 3D convolutional neural network for real-time object recognition. In: Signal Processing Letters (2015)
21. Novotni, M., Klein, R.: Shape retrieval using 3D zernike descriptors. Comput. Aided Design **36**, 1047–1062 (2004)
22. Praun, E., Hoppe, H.: Spherical parametrization and remeshing. In: ACM Transactions on Graphics (TOG), vol. 22, pp. 340–349. ACM (2003)
23. Rustamov, R.M.: Laplace-beltrami eigen functions for deformation invariant shape representation. Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP 2007, pp. 225–233, Aire-la-Ville (2007)
24. Shen, L., Makedon, F.: Spherical mapping for processing of 3-D closed surfaces. In: Image and Vision Computing (2006)
25. Shen, L., Makedon, F.: Spherical mapping for processing of 3D closed surfaces. Image Vis. Comput. **24**(7), 743–761 (2006)
26. Shi, B., Bai, S., Zhou, Z., Bai, X.: DeepPano: deep panoramic representation for 3-D shape recognition. IEEE Signal Process. Lett. **22**(12), 2339–2343 (2015)
27. Sinha, A., Choi, C., Ramani, K.: DeepHand: robust hand pose estimation by completing a matrix imputed with deep features. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
28. Sinha, A., Ramani, K.: Multi-scale kernels using random walks. Comput. Graphics Forum **33**(1), 164–177 (2014)
29. Solomon, J., de Goes, F., Studios, P.A., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., Guibas, L.: Convolutional wasserstein distances: efficient optimal transportation on geometric domains. ACM Transactions on Graphics (Proceeding SIGGRAPH 2015) (2015)

30. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G.: Multi-view convolutional neural networks for 3D shape recognition. In: Proceeding ICCV (2015)
31. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: Proceedings of the Symposium on Geometry Processing, SGP 2009, pp. 1383–1392, Aire-la-Ville (2009)
32. 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, pp. 1912–1920 (2015)
33. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.-Y.: Mesh editing with poisson-based gradient field manipulation. ACM Trans. Graph. **23**(3), 644–651 (2004)
34. Zhao, X., Su, Z., Gu, X.D., Kaufman, A., Sun, J., Gao, J., Luo, F.: Area-preservation mapping using optimal mass transport. IEEE Trans. Visual Comput. Graphics **19**(12), 2838–2847 (2013)
35. Zou, G., Hu, J., Gu, X., Hua, J.: Authalic parameterization of general surfaces using lie advection. IEEE Trans. Visual Comput. Graphics **17**(12), 2005–2014 (2011)