3-D Keypoint detection with Deep Neural Networks, Sparse Autoencoders and Mesh Simplification

Elías Josué Puma Chávez

Escuela Profesional de Ciencia de la Computación,
Universidad Nacional de San Agustín,
Arequipa, PE
Email: eliasj.puma@gmail.com

3-D keypoint detection plays a fundamental role in the Computer Vision field, detection of these salient points in the local surfaces of a 3-D object is important in order to perform certain tasks such as registration, retrieval and simplification. There has been a lot of research in the field of 3-D keypoint detection, most of them take a geometrical approach which have a good performance but lack flexibility to adapt to changes such as noise and high curvature points that are not keypoints to human preference. A good approach seems to be machine learning methods that can be trained with human annotated training data. In this paper a new method is proposed using deep neural network with sparse autoencoder as the regression model due to their great ability for feature processing. The analysis shows this method would outperform other methods that are widely used.

Keywords: Keypoint detection; Deep Neural Networks; 3-D Model; Sparse Autoencoders

1. INTRODUCTION

Several computer-dependent areas are benefited of the applications that 3-D Models have in them. The growth of 3-D data has increased in the last years with the availability of low-cost 3-D capture devices [1]. The ability to analyse, process and select relevant information from them is an active research area.

3-D interest point detection is a difficult task for several reasons [1, 2]. *First*, there is not a definition of what exactly an interest point is, but a common assumption relates the level of protusion of outstanding local structures with the measure of interest of such local structure [1]. Therefore, we can say that, planar sections of an area vertices have a low interest level and local areas with diferent structures the interest level will be the opposite, the same with the edges of an object. Its main caracteristic is its invariance to transformations in the object itself. *Second*, vertex density is different for every 3-D model which makes harder the task of selecting a local area, at least an area that will have inside significant information. *Third*, information obtained from a 3-D model are only vertex positions and connectivity between them which means the interest level will depend only from the information we can retrieve from different calculations. These are not the only present challenges but are sufficient to explain the reason this method is prepared to handle these difficulties.

Previously the common approach to 3-D keypoint detection was centered in calculating geometric properties of the models, although in recent years researchers also developed machine learning techniques that with the objective to outperform the former ones by handling the problems of: Different calculations in different areas of the model [3], false positives obtained from noise or local variation and keypoint detection valuable according to human preference.

Our algorithm extends the work done with machine-learning methods, and introduces a new architecture for a more efficient Deep Neural Network making use of Sparse Autoencoders that provides better distributed input nodes, and enables it to learn more interesting features from 3-D models that were processed beforehand (k-ring analysis, best fitting calculation [1] and mesh simplification [4]). The processing done before a local area of the 3-D model is put as the entry to the DNN improves the efficiency of the training proccess by: (1) Reducing the number of inputs to the DNN, (2) reducing the number of analysed vertices, (3) making the keypoint detection dependent to the model size and (4) making it invariant to the model position.

Although our experimentation couldn't reach the state of the method proposed here, we present an approximate implementation that sustains our proposal that a well-trained neural network can detect keypoints independently to the position or transformations a model can suffer before being put to be processed.

The rest of this paper is organized in the following way: Section 2 introduces previous work done in the area, In section 3 we explain the basic principles to build a Deep Neural Network using sparse autoencoders. Section 4 presents the idea this method is based on. Results are presented in Section 5 and conclusions in Section 6.

2. RELATED WORK

In recent years researchers have proposed several techniques for 3-D keypoint detection. Most of them are based on geometric methods, that work on meshes or use surface reconstruction [5]. Techniques that follow this approach will be introduced below:

Lee [4] addresses interest point detection through the use of local curvature estimates together with a center surround scheme at multiple scales. The total saliency of a vertex is defined as the sum of Difference of Gaussian (DoG) operators over all scales.

The THRIFT algorithm [6] is a 3-D extension of 2D algorithms like SIFT and SURF. They divide the spatial space by a uniform voxel grid and calculate a normalized quantity D for each voxel. To construct a density scale-space Flint et al. convolve D with a series of 3-D Gaussian kernels $g(\boldsymbol{\sigma})$. This gives rise to a scale-space $S(\boldsymbol{p}, \boldsymbol{\sigma}) = (D \otimes g(\boldsymbol{\sigma}))(\boldsymbol{p})$ for each 3-D point $\boldsymbol{p}$. Finally, they compute the determinant of Hessian matrix at each point of the scale space. Within the resulting $3 \times 3 \times 3 \times 3$ matrix, a non maximal suppression reduces the entries to local maxima, which become interest points.

Mian [7] related the repeatability of keypoints (extracted from partial views of an object) with a quality measure based upon principal curvatures. For each point $\boldsymbol{p}$ they rotate the local point cloud neighborhood in order to align its normal vector $n_{\boldsymbol{p}}$ to the z-axis. To calculate the surface variation they apply a principal component analysis to the oriented point cloud and use the ratio between the first two principal axes of the local surface as measure to extract the 3-D keypoints.

Sun [8] apply the Laplace-Beltrami operator over the mesh to obtain its Heat Kernel Signature (HKS). The HKS captures neighborhood structure properties which are manifested during the heat diffusion process on the surface model and which are invariant to isometric trans- formations. The local maxima of the HKS are selected as the interest points of the model.

In 2011 Sipiran and Bustos extended the Harris operator for 3-D meshes [1] using an adaptive technique to determine the neighborhood of a vertex, over which the Harris response on that vertex was calculated. Their method was said to be robust to several transformations, using the SHREC feature detection and description benchmark to measure their results.

Lin, Zhu, Zhang and Liu proposed a geometric technique [9] based in the tangencial planes traced for each vertex and other transformations in the mesh some of them can be also found in [3].

The lack of flexibility of some geometric methods led researchers to look for new approaches to achieve 3-D keypoint detection.

3. DEEP NEURAL NETWORKS USING AUTOENCODERS

3.1. Autoencoders and sparsity

An autoencoder tries to learn a function $h_{W,b}(x) \approx x$, where x is an unlabeled dataset $\{x_1, x_2, \cdots, x_m\} \in \mathbb{R}^m$, which is an approximation to the identity function, and by so having an output $\hat{x}$ similar to the input x. A representation of the framework is found in Figure 1.

By placing constraints on the autoencoder (e.g. limiting the number of hidden units) its hidden layer learns a compressed representation of the input, or in other words the internal structure of the input data.

An sparse autoencoder is a variant of autoencoder, that constraints the activation value of neurons on its hidden units, and by so learns interesting structure of the input data. The sparsity cost is added to the cost of the neural network as in (1).

$$J_{sparse}(W,b) = \frac{1}{2}\|h_{W,b} - x\|^2 + \beta \sum_{j=1}^{S} KL(\rho\|\hat{\rho}_j) \quad (1)$$

where,

$$KL(\rho\|\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \quad (2)$$

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m} a_j^2(x_i) \quad (3)$$

3.2. Deep Neural Networks with Sparse Autoencoders

An autoencoder neural network is an unsupervised learning algorithm that makes use of backpropagation, setting the targets values to be equal to the inputs.

Using deep sparse autoencoder (DSAE) can learn high-level features of the input data effectively. Each Sparse Autoencoder in a DSAE can learn features at different levels (from low level to high level). A representation of this architecture is shown in Figure 2.

4. PROPOSAL

This work is inspired by the work of [3] by the use of sparse autoencoders as the regression model in order

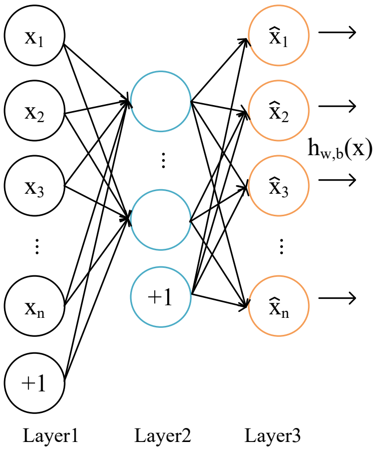


FIGURE 1. Representation of an autoencoder [10]

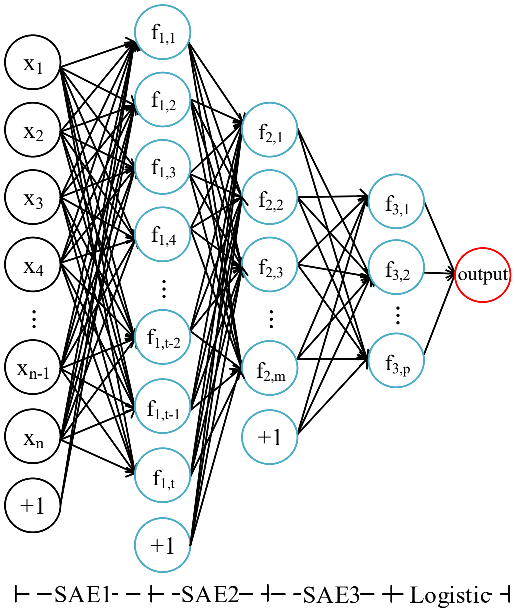


FIGURE 2. Deep Neural Network representation, composed of three SAEs as the hidden layers [3]

to learn features from local and global information generated from a human-annotated keypoint database. Also this proposal is influenced by [11], so for us to achieve the 3-D keypoint detection we train a 3-layered locally connected sparse autoencoder similarly to their technique, such technique's results revealed to be a inexpensive way to develop high-level features from unlabeled data, from that study this work presents an adapted architecture for 3-D meshes. Taking a different approach than the other techniques mentioned above, resized and simplified segments of the 3-D mesh will be used as the input for our Deep Neural Network, this will enable our DNN to learn when one of these segments has inside a keypoint.

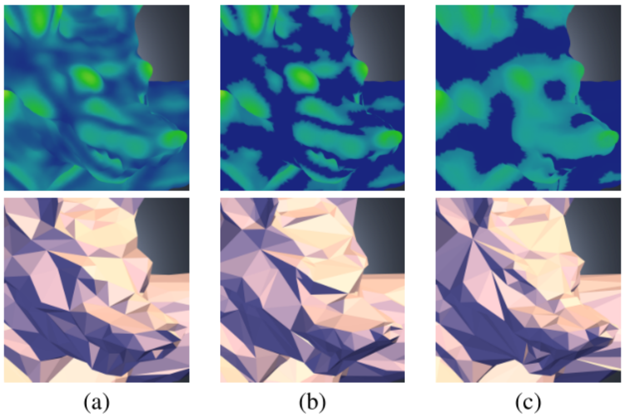


FIGURE 3. Saliency-based weights and quality of a 99% simplification for three choices of the simplification weights: (a) Original mesh saliency, (b) amplified mesh saliency and (c) smoothed and amplified mesh saliency [4]

4.1. Mesh Simplification

One of the objectives of this research is to reduce the amount of data that is processed to detect 3-D keypoints, to accomplish that we propose to simplify the input our Deep Neural Network receives. Several approaches have been proposed and discussed, we make use of Mesh Saliency's approach [4]: Guide the simplification proccess through mesh curvature obtained from local areas using a center-surround mechanism to identify regions that are different from their local context. An example is seen in Figure 3.

4.2. Architecture

This technique can be seen as a set of sparse deep autoencoders that similarly to [11] has two fields in it: local receptive fields, pooling normalization (the architecture taken as a base can be seen on the Figure 4). Local receptive fields scale the autoencoder to big inputs, connecting the autoencoder's features to a small region of the next lower layer. These sublayers are know as filtering and pooling.

Originally the neurons in the first sublayer were connected to pixels in all input channels [11], but in order to adapt this architecture it is proposed to use the 3-D vertices and their connectivity information as the input channels and by so adding more receptive fields.

4.3. Training

As mentioned before the first layer input...

To train the Deep Neural Network what is to be done at first is to train each Sparse Autoencoder and a final logistic regression layer, then following the schema from [3] stack the four layers together and backpropagate the whole DNN to fine tune it.

The goal of this approach is to reduce the proccessing that is performed, instead of evaluating each vertex

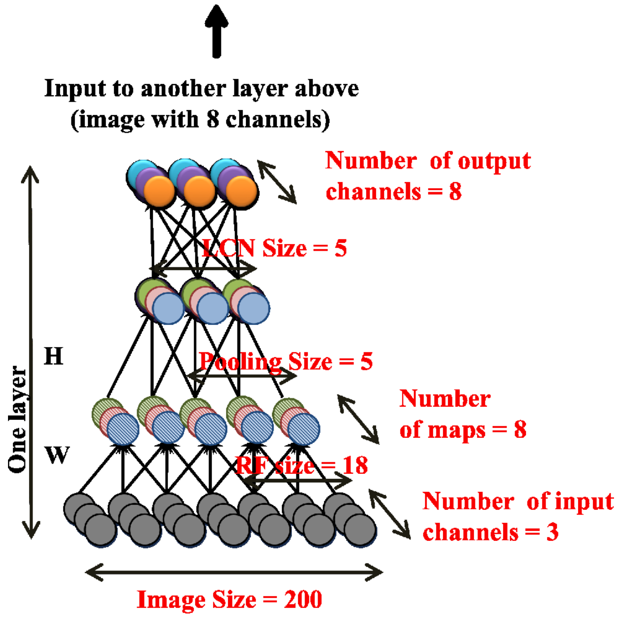
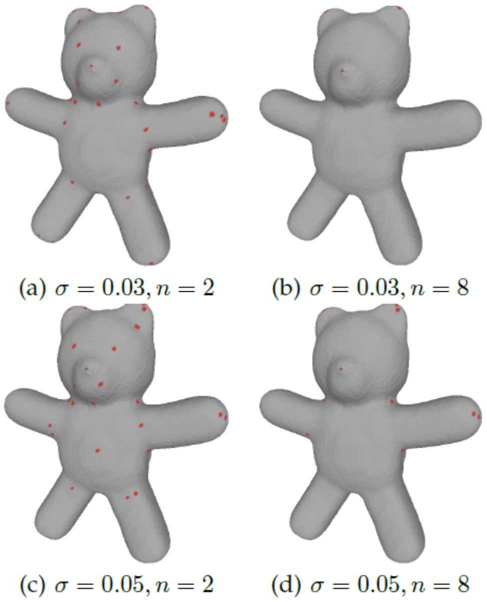


FIGURE 4. Large scale unsupervised learning architecture [11]


(a) $\sigma = 0.03, n = 2$ (b) $\sigma = 0.03, n = 8$

(c) $\sigma = 0.05, n = 2$ (d) $\sigma = 0.05, n = 8$

FIGURE 5. Annotated points for a teddy bear model using four different σ/n clustering parameters [2].

in the DNN which is expensive, we can perform the neccesary calculations just for some samples of the 3-D object and discard if those samples don't contain any keypoints, in the case we find the presence of keypoints we will perform further calculations to choose the sample keypoint.

5. RESULTS

To evaluate the performance of this keypoint detector we will compare it with five state-of-the-art methods as did in [3], they are: Harris 3D [1], HKS, Salient Points, Mesh Saliency [4] and Scale Dependent Corners.

5.1. The Data Sets

In [2], Data sets from Dutagaci were used to train their machine learning tree model. Those models were organized in two different data sets: Data Set A and Data Set B, having 24 and 43 models respectively, were users were to click on the most interesting points of the models.

Data Set A had 23 people performing the annotations, and Data Set B had 16. There were some variances in the annotation, user clicks were clustered using geodesic distances [12] to improve the consensus. An example of the clustering with different parameters is observer at Figure 5. We will use this data set to train our Deep Neural Network.

5.2. Evaluation criteria

Datagaci also proposed a new criteria for performance evaluation, he stated that a keypoint is rightly identified if the geodesic distance between the point given by the algorithm and the keypoint obtained from the dataset is less than an error tolerance.

He also defined the Intersection Over Union (IOU) criterion for object detection in images. It is calculated with the equation (4).

$$IOU(r) = \frac{TP}{FN + FP + TP} \tag{4}$$

Let N_C be the number of correctly detected points and N_A represent the number of detected interest points by the algorithm, $FP = N_A - N_C$ is the number of false positives and $FP = N_G - N_C$ is the number of false negatives. $TP = N_C$ being the true positives. N_G is the number of ground truth points.

5.3. Results

Our results are based in a implementation of a method similar to the proposed in this paper: Similar to [3], we select two thirds of Data Sets A and B to train our Neural Network, and the remaining third of each dataset is used for testing. Our Neural Network is composed of a layer trained through linear regresion linked to another logistic layer that retrieves a value that represents the likelihood of one vertex to be a keypoint, the first layer input dimensions are: 50 × 50, the output has a dimension of: 7 × 7 that is the size of the input for the logistic layer, for each model we count with the positions of the human-annotated keypoints, for this experiment we limit the processing to evaluate each one of the vertices of the model, we mark as a keypoint to the closest vertex to these different positions, this enables the DNN to learn some features but with the experiments we confirmed our assumptions that there would be several vertices closed to each other

	IOU	*FNE*	*FPE*
3D-Harris	0.102	0.343	0.879
HKS	0.216	0.530	0.530
DNN	0.275	0.561	0.561
Our method	**0.071**	**0.211**	**0.915**

TABLE 1. Average IOU, FNE, FPE values on the test dataset.

marked as keypoints. At first we calculate the $V_k(v)$ neighborhood for each vertex, this is the k-ring of this vertex using 7 as k, then we calculate the centroid of this sub-mesh to translate the set of points to a impartial position, where we apply Principal Component Analysis to calculate the best fitting plane, we achieve that by finding the less significative component and transform this into a 2-D plane that is the entry for our DNN, after a training of seven hours in a core i5 computer with no GPU processing enabled, we stored the values obtained from the logistic layer using the testing models and picked only the vertices with the highest obtained values, we present two examples of the keypoints detected over a model with transformations in Figure 6 and the comparatives using IOU criteria in Table 1.

As seen in the results, we reached a good performance even when it doesn't outperform the state-of-the-art current methods it is able to detect a good amount of interest points with a implementation that roughly approximates to the method proposed, with the right clustering process these values would present a greater improvement.

6. CONCLUSIONS

We presented a new approach to detect keypoints of a 3D model making use of machine learning, specifically autoencoders in deep neural networks, to have a Multilayer Deep Neural Network trained with local surfaces of simplified meshes of models found in human annotated keypoint Data Sets, this approach leads to a simplification in the proccessing to be done previous to the training, as well as this trained DNN will be able to perform the detection of keypoints depending of the size of the object making it suitable compared to the visual characteristics humans notice in these models.

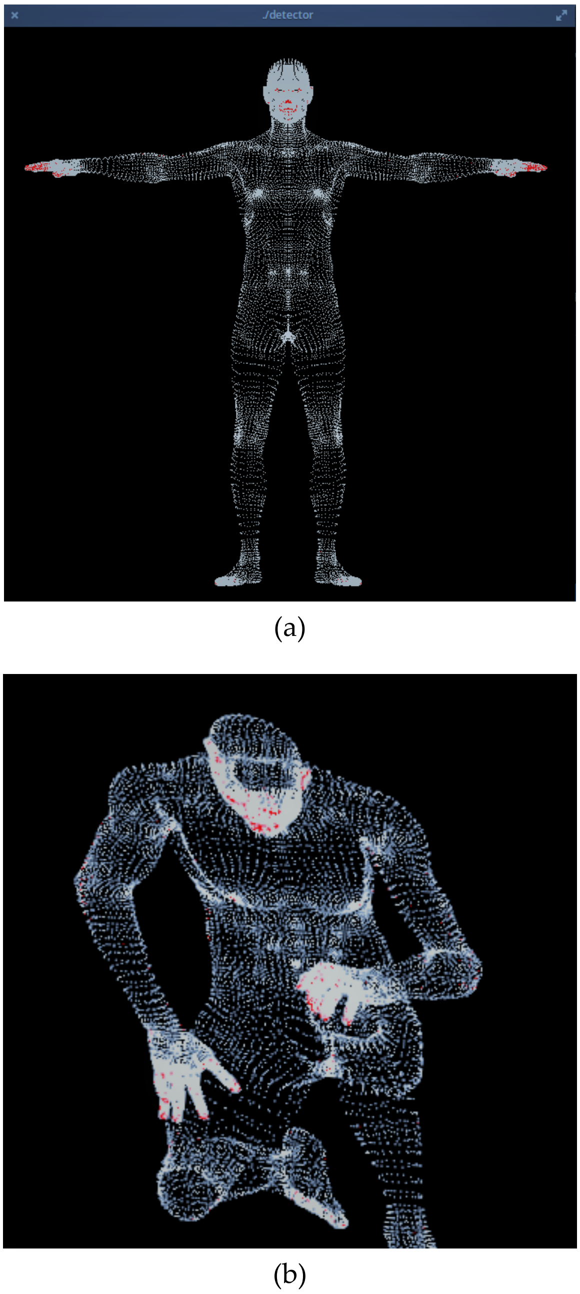
ACKNOWLEDGEMENTS

This research was undertaken as part of the motivation received from my family and future wife that I probably met already.

REFERENCES

[1] Sipiran, I. and Bustos, B. (2011) Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes. *Visual Computer*, **27**, 963–976.

[2] Teran, L. and Mordohai, P. (2014) 3D Interest Point Detection via Discriminative Learning. *Eccv* , **?**, 1–8.



(a)

(b)

FIGURE 6. Detected interest points in (a) the model in a null position and (b) the model with an scaling transformation.

[3] Lin, X., Zhu, C., Zhang, Q., and Liu, Y. (2016) 3D Keypoint Detection Based on Deep Neural Network with Sparse Autoencoder. , **?**, 1–13.

[4] Lee, C. H., Varshney, A., and Jacobs, D. W. (2005) Mesh saliency. *ACM Transactions on Graphics*, **24**, 659.

[5] Garstka, J. and Peters, G. (2015) Fast and robust keypoint detection in unstructured 3-d point clouds. *Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on*, pp. 131–140. IEEE.

[6] Flint, A., Dick, A. R., and Van Den Hengel, A. (2007) Thrift: Local 3d structure recognition. *dicta*, pp. 182–

188.

[7] Mian, A., Bennamoun, M., and Owens, R. (2010) On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, **89**, 348–361.

[8] Sun, J., Ovsjanikov, M., and Guibas, L. (2009) A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum*, pp. 1383–1392. Wiley Online Library.

[9] Lin, X., Zhu, C., Zhang, Q., and Liu, Y. (2016) 3D Interest Point Detection Based on Geometric Measures and Sparse Refinement. , **?**, 1–9.

[10] Ng, A. (2011) Sparse autoencoder. *CS294A Lecture notes* , **?**, 1–19.

[11] Le, Q. V. (2013) Building High-Level Features Using Large Scale Unsupervised Learning. , **?**, 8595–8598.

[12] Surazhsky, V., Surazhsky, T., Kirsanov, D., Gortler, S. J., and Hoppe, H. (2005) Fast exact and approximate geodesics on meshes. *ACM transactions on graphics (TOG)*, pp. 553–560. ACM.