Deep Shells with Landmarks Supervision

Yibo Jiao
University of British Columbia
jyibo@cs.ubc.ca

Abstract

This project proposed a semi-automatic learning approach that includes landmark locating and supervision for finding 3D human scan point-to-point correspondences. Current axiomatic unsupervised methods Smooth Shells [4] and its learning approach Deep Shells [5] are both challenged by self-similarity, especially symmetrical problems during the matching pipeline. This work introduces a landmark supervision method based on Deep Shells by adding linear soft constraints to minimize cases of symmetrical problem. To that end, we derive a simple yet efficient pipeline based on the two Shells that better distinguishes self-similarities yet has similar overall matching quality.

1. Introduction

Finding point-to-point correspondences on meshes is a core operation in geometry processing. Traditional methods for dense shape matching problems like FSPM [8] and 3D-CODED [6] requires knowledge about deformation and noise of 3D shapes. Fortunately, Smooth Shells and Deep Shells proposed novel unsupervised algorithms that do not need ground truth labels and still handles various types of noises and deformations.

Both two Shells used hierarchical matching algorithm which iteratively aligns approximated shapes in a coarse-to-fine manner, thus Shells need relatively good alignments as a start to optimize. This initial alignment finding is called initialization in Shells. However, finding a good enough initial alignment was challenging because self-similarities like symmetries and self-touching cases for 3D scans are still difficult to be detected and distinguished for these two pure unsupervised methods. On the one hand, Smooth Shells used surrogate based Markov Chain Mote Carlo sampling for initialization, which generates a large number of proposed alignments and select the one that has lowest cost. On the other hand, Deep Shells used exactly the same hierarchical matching algorithm but it introduced a fully differentiable settings and use learning to find refined local features to initialize the matching pipeline.

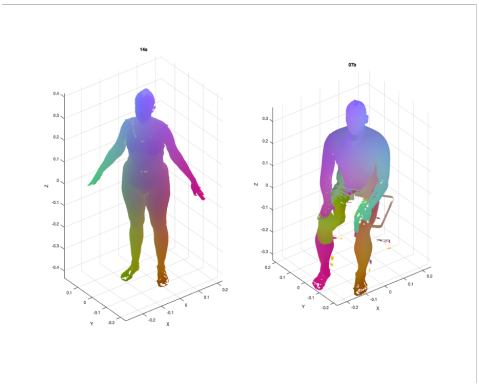

Figure 1. Deep Shells with a bad initialization is challenged by self-similarities. This trained model aligns the upper body of the two shapes on the opposite side.

Although taking advantage of deep learning made Deep Shells more computationally efficient than Deep Shells, the learned refined local features were not guaranteed to be able to disambiguate self-similarities because Deep Shells only randomly sampled some points on the input shape for training and it used SHOT [11] descriptors which are too local and do not include global information. A failure case for trained Deep Shell is in Figure 1.

Contribution In this project, we introduced landmark supervision during initialization and iterative matching pipeline. This work includes locating anthropometric landmarks for gathering supervision labels and apply landmark distances constraints onto optimization. Constraints of landmark distances penalise alignments that matches landmarks on the the other side of human shape, which forces the initialization that has symmetrical problem always has larger objective energy. Therefore, an optimal initial alignment will always has minimum possibility of symmetrical problem. This method is considered as a semi-automatic approach i.e. only requires a small number of ground truth labels of landmarks for training and still take advantage of unsupervised learning for efficiency.

2. Background

2.1. Smooth Shells and MCMC

Self-similarities were solved by running surrogate based Markov chain Monte Carlo sampling initialization. This approach samples various initial alignments τ and pick the one that has minimal energy result. The energy in this initialization step is [4]:

$$E_{init}(P, C, \tau) = \|PX_K^* - Y_K\|_F^2 \tag{1}$$

Where X^*, Y are approximated source and target shapes, P, is correspondences and (C, τ) are alignments to be initialized. τ is the displacement parameter, i.e. point-wise translation. C is functional map [9]. Since the correspondence P is computed by nearest neighbour searching in extrinsic coordinates, this energy only requires an optimal τ. Ideally, if we sample enough number of alignments and pick a reasonable threshold of energy to select optimal alignment, $(\tau_{best}, X_{best}^*)$ are guaranteed to initialize the matching and find P in coarsest level. However, exploring initial poses $\tau_{prop} \in \mathbb{R}^{K\times 3}$ are very computational costly, and Deep Shells used learning to replace this step to improve efficiency.

2.2. Deep Shells and Spectral Convolution

Deep Shells replaces the costly initialization in Smooth Shells by learning refined local features. Instead of searching initial poses τ, Deep Shells takes SHOT descriptors as input and uses spectral convolution to get spectral information, then spectral filters are learned to extract refined local features. The energy in this initialization layer is [5]:

$$E_{init}(\pi) = \int_{\mathcal{X}\times\mathcal{Y}} \left\|G^{\mathcal{X}}(x) - G^{\mathcal{Y}}(y)\right\|_2^2 d\pi(x, y) - \lambda H(\pi) \tag{2}$$

Where $\mathcal{X}, \mathcal{Y}$ are descriptors, G are learned features computed by learned spectral filters, π is initial soft correspondences, which is simply a probability matrix indicating point-wise matching probabilities. With same hierarchical matching algorithm in Smooth Shell followed by this initialization layer, spectral filters are learned to result optimal π_{init} to reach minimal energy. However, learned spectral features are not guaranteed to be faithfully distinguish self-similarity because training spectral filters depends on input descriptors, and SHOT descriptors are unstable, local and highly depends on triangulation. In addition, Deep Shells only samples a small number of points to train, with full-resolution training i.e. take all input points of all shapes as training data learns better refined features but highly increases the training cost.

2.3. Motivation and Goal

The trade-off between training resolution and quality of refinement of learned local features in Deep Shells motivates this project. Instead of training with all points, we sample points that are more suitable for representing local features and easier to obtain ground truth labels, we call such points landmarks. This work takes advantage of unsupervised learning method and supervise a small number of located landmarks to help distinguish self-similarities.

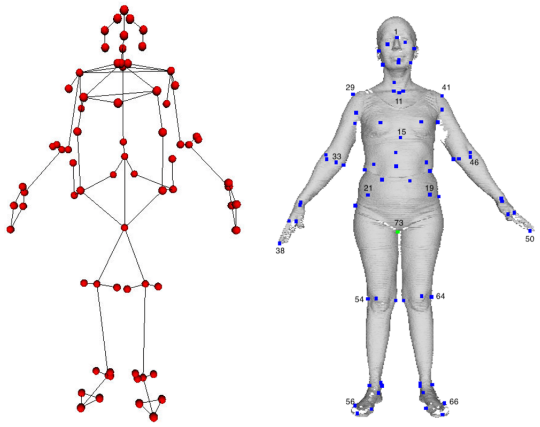

Figure 2. Structure of the landmark graph [2] and auto-locating result.

The main objective of this project is to reduce cases of symmetrical problem of Deep Shells by learning better local features using landmark supervision and yet maintain the overall quality of matching and computation time especially for denser meshes.

3. Method

3.1. Locating Anthropometric Landmarks

We choose antrhopometric landmarks for supervision. The traditional way of locating such landmarks relies on placing markers on the human body prior to scanning. To make the pipeline automatic, we use automatic locating algorithm for anthropometric landmarks [2]. Since inter-class matching tasks requires knowledge of locating landmarks on non-human models, and this method only works for 3D human models, this project focuses on matching human shapes.

The problem of auto-locating landmarks is formulated as a probabilistic inference problem and solved by two steps of locating process. In the first step, training data with ground truth landmarks that are measured by experts is used to learn the parameters of a pairwise Markov random field. In the second step, we perform probabilistic inference to find optimal labeling of the landmarks. The network is trained to be able to localise 73 anthropometric landmarks locating around human scans, this provides us truth labels for landmarks supervision. We embed the same algorithm for auto locating as a pre-processing step in our data driven pipeline.

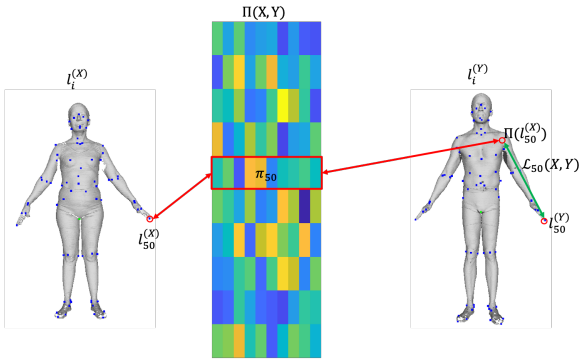

Figure 3. Example of landmark projection and landmark distance function for landmark 50.

An example of the structure and result is shown in figure 2.

In this work, we define landmarks on human shape X by $l_i^{(X)}, i = 1, ..., 73$, given a correspondence P that maps from source shape X to target shape Y, and landmark i, define landmark projection by $P(l_i^{(X)})$, this will be the matched position for landmark i of shape X on shape Y. The landmark distance is defined by the error of localisation of matched position and the ground truth landmark position on target shape Y:

$$\mathcal{L}_i(X,Y) = \left\| P(l_i^{(X)}) - l_i^{(Y)} \right\|_2^2 \tag{3}$$

However, with differentiable settings in Deep Shells, correspondence P is replaced by soft correspondence Π, we define soft projection as:

$$\Pi\left(l_i^{(X)}\right) = \pi^T Y, \text{for } \pi \in \Pi(X,Y) \tag{4}$$

Which we take probability vector π that indicates soft correspondence that projects landmark $l_i^{(X)}$ onto Y, taking inner product will give us a barycentric extrinsic coordinate with probabilistic weights. Thus differentiable landmark distance function is:

$$\mathcal{L}_i(X,Y) = \left\| \Pi(l_i^{(X)}) - l_i^{(Y)} \right\|_2^2 \tag{5}$$

A visual illustration for this idea of differentiable landmark projection and landmark distance definition is as shown in figure 3.

3.2. Assumption

Recall that our goal of this work is to minimize cases of symmetrical problem, we assume that large landmark distances are only caused by two possible situation: either overall matching is erroneous or the matching has symmetrical problem.

As shown in figure 3, if the large landmark distance value is caused by matching landmarks to an area that do not have similar local features, this is the case of erroneous matching because the learned spectral filters cannot even distinguish local features. We do not consider this case because Deep Shells has proved to have good quality of matching for benchmark datasets. However, if the Shells is challenged by symmetrical problem and match landmarks on the other side of human shape, that means learned parameters are able to distinguish local features but challenged by self-similarities because of the symmetrical properties of human bodies, this case still have large landmark distances.

3.3. Energy Formulation with Landmark Constraints

To apply landmark constraints, we simply add landmark distances onto original Deep Shells' energy as equation (2) for optimization. This constraints need to be applied on both the initialization layer and hierarchical matching steps.

For initialization, we formulate the energy as:

$$E_{init}(\pi) \quad \text{s.t.} \quad \min_{X^*,Y} \sum_{i=1}^{73} w_i \mathcal{L}_i(X^*, Y) \tag{6}$$

which is equivalent to:

$$E_{init}(\pi) + \mu \sum_{i=1}^{73} w_i \mathcal{L}_i(X^*, Y) \tag{7}$$

where we assign different weights w_i for each landmark because landmarks that are farther from the center axis are more affected by symmetrical problems, and hyperparameter μ controls the hardness of the constraint, when μ goes to infinity, landmark distances are hard constraints. In this way, we force the initialization layer to learn spectral filters that result initial correspondence with minimal landmark distances in order to solve symmetrical alignment in the initialization step.

For hierarchical matching steps, similar addition of landmark distances is applied:

$$E(\pi, C, \tau) + \mu \sum_{i=1}^{73} w_i \mathcal{L}_i(X^*, Y) \tag{8}$$

3.4. Iterative Optimization

By formulations of energy described as above, the initialization energy only has variable π to be optimized, we use Adaptive Moment Estimation to optimize.

However, for hierarchical matching, solving π, τ, C together are difficult, we decompose the problem by fixing deformation (τ, C) and correspondence π iteratively made the optimization problem easier. We use similar strategy for optimizing ARAP deformation energy [10]. Since the

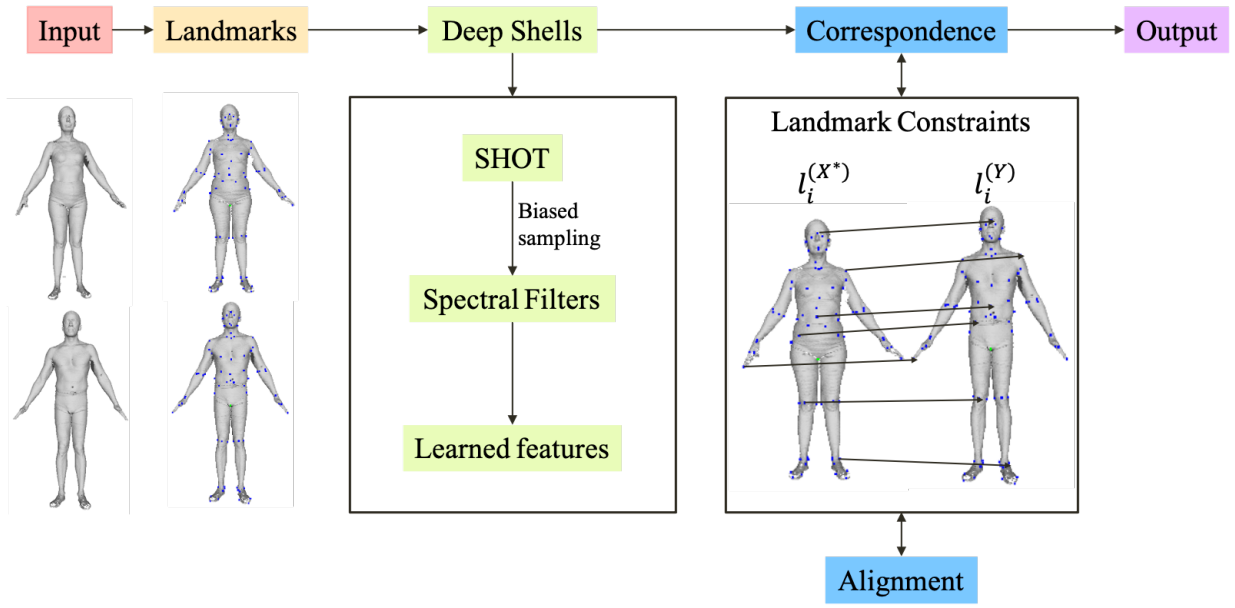

Figure 4. Pipeline overview of Deep Shells with landmark Supervision.

constraint term is only dependent on π, and we also have initial soft correspondence π_{init}, we firstly fix π and optimize (τ, C), the energy is a non-linear least square problem, which can be solved by either Gauss-Newton optimization or in closed form. Then we fix optimized (τ, C), π is also differentiable. We repeat above steps several times until the result converges.

3.5. Pipeline Overview

See Figure (4) for the overview of the project, for a given pair of training shapes, we firstly use auto-locating of anthropometric landmarks as part of pre-processing to prepare landmark and SHOT descriptor information. In initialization layer, instead of randomly sample points, we sample nearest points around each landmarks. Landmark distance constraints are also applied on both initialization layers and hierarchical matching steps to solve symmetrical problems.

4. Results

4.1. Implementation Details

We implemented the network based on Deep Shells, instead of using synthetic human mesh dataset like SCAPE [1] and FAUST [3], we train our model with CAESAR [7] dataset, which are real-world scans with higher resolution(more than 200K vertices). We normalized the meshes by centering the object on the origin and re-sizing the meshes to have same overall surface area.

	SS	DSFR	DSRS	Ours
FAUST	725	680	**221**	230
SCAPE	742	613	**189**	201
CAESAR	1058	1343	372	**367**

Table 1. Runtime comparisons between methods, SS stands for Smooth Shells, DSFR is Deep Shells in full resolution, DSRS is Deep Shells with 1000 random sampled points.

4.2. Computational Efficiency Comparison

One of the main goal of this project is to make training more efficient, we trained our and comparative networks on 3 datasets discussed above using 4 gpus. We recorded the time of running one pair of matching. Our model is trained with sampling 20 nearest points around each landmark, thus 1460 points are selected. See Table (1) for runtime comparisons. Our method has similar running time compared to Deep Shells because landmark operations are efficient and optimizations are least square problems.

4.3. Symmetrical Problem Results

Our method solves symmetrical problems during matching, we trained our model using 25 shapes(625 pairs) and tested on 30 shapes(900 pairs) and compute the maximum error of localisation of landmarks for test results, relatively large errors indicates symmetrical problems, and we record the number of matching pairs that has such result. See Table (2) for symmetrical cases comparison and Figure (5) for an example case that Deep Shells failed and our method

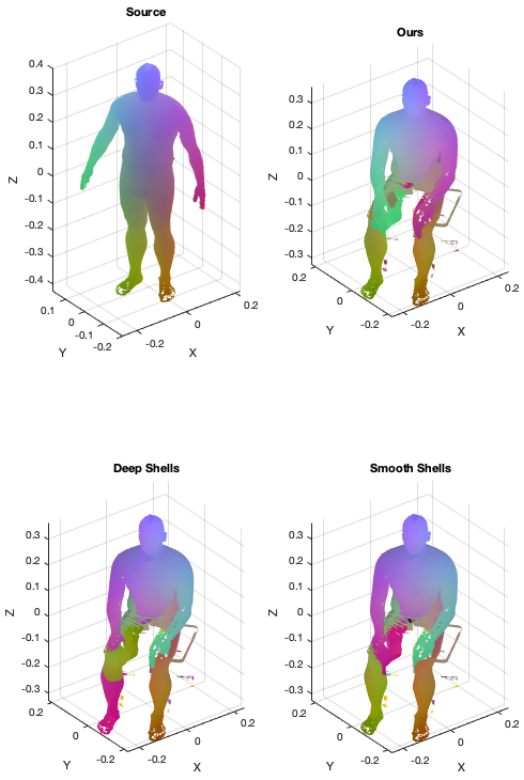

Figure 5. An example of noisy and partial scan for matching, both DS and SS were challenged by symmetrical problems while ours worked.

	SS	DS	Ours
FAUST	59	98	**23**
SCAPE	61	103	**17**
CAESAR	92	178	**25**

Table 2. Number of cases that models failed to distinguish symmetrical problems.

worked for the same pair. We choose the scan that has extreme degrees of noise and partiality.

4.4. Overall Matching Quality

Our method achieves a higher accuracy on distinguishing symmetrical problem, it is at the same time able to obtain similar overall high quality correspondences that are comparable with the Shells, we evaluate matching quality by recording average geodesic error for all tested data pairs. See Figure (6) for details.

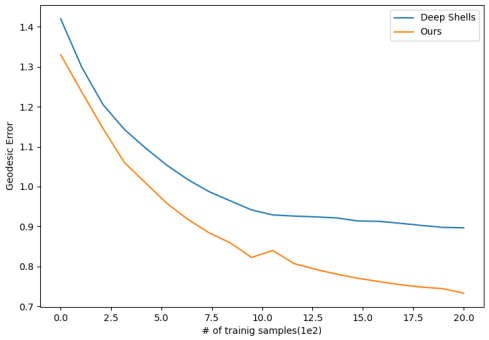

Figure 6. A quantitative comparison of our method and Deep Shells, we train with different number of training samples and show average geodesic error tested with same 900 testing scans.

4.5. Limitation

Although our method is more efficient and solved symmetrical cases, like all other shape matching methods, self-touching problem is still a challenging problem. Our method is highly dependent on accuracy of autolocating for landmarks, thus we need a more reliable algorithm to provide us more precise landmarks locations. In addition, obtaining landmarks from non-human scans are difficult, inter-class shape matching problems are still more suitable for pure unsupervised methods.

5. Conclusion

We proposed a stable and effective method based on Shells that solves symmetrical problem by optimizing constraints problems. Our method takes advantage of spectral convolution for feature extracting and learns more refined features that can distinguish self-similarities and used landmark supervision to constrain optimization. We show that constraining landmark distances not only improve overall matching quality but also better easy to implement and computational effective.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. In *ACM SIGGRAPH 2005*, pages 408–416, 2005. 4

[2] Zouhour Ben Azouz, Chang Shu, and Anja Mantel. Automatic locating of anthropometric landmarks on 3d human models. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 750–757, 2006. 2

[3] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 3794–3801, 2014. 4

[4] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020. 1, 2

[5] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems*, 34, 2020. 1, 2

[6] Thibault Groueix, Matthew Fisher, Valdimir G. Kim, Bryan C. Russel, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. 2018. 1

[7] Robinette Kathleen, Daanen H, and Paquet Eric. The caesar project: a 3-d surface anthropometry survey. *3-D Digital Imaging and Modelling*, pages 380–386, 1999. 4

[8] Or Litany, Emanuele Rodola, Alex Bronstein, and Michael Brostein. Fully spectral partial shape matching. 36(2):1681–1707, 2017. 1

[9] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 2

[10] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. 2007. 3

[11] Federico Tombari, Samuele Salti, and Luigi Di Stefano. 3unique signatures of histogram for local surface description. 16(9):356–369, 2010. 1