Neural Network Architectures for Estimating Mesh Representations of Cloth Objects

○Solvi Arnold (Shinshu University)　Kimitoshi Yamazaki (Shinshu University)

1. Introduction

We consider the problem of quickly obtaining mesh representations of cloth items for robotic manipulation. In real-world settings, it is feasible to obtain point-clouds and images of a cloth item, but hard to obtain mesh representations. For known rigid items, we may be able to match 3D models against point-cloud and/or image data, but for cloth items, deformability complicates this task enormously. Furthermore, self-occlusion is common and presents more challenges than in rigid objects.

Examples of shape estimation for deformable objects are found in [1][2], but the extent of deformation is less extreme than typically encountered in cloth manipulation (no self-occlusion, constrained deformation). The shape estimation routine in [3] handles more variation, but objects' shape is estimated by lifting objects up and then scanning while rotating them, which naturally perturbs their shape. We instead consider mesh estimation from passive observation from a single angle. We pursue a neural network-based approach for mapping voxel representations (which can be derived easily from depth images or pointclouds) to explicitly probabilistic mesh representations.

2. Architectures

Each position on a cloth item is identified (regardless of deformation) by its UV coordinates. The problem consists in assigning XYZ coordinates to UV coordinates. The task of obtaining meshes from data with occlusion is an underdetermined problem. We cannot, in principle, assign XYZ coordinates to each point on the cloth with precision. For this reason, it is desirable to represent uncertainty in a locally quantified manner. We let the output of the networks represent a multivariate Gaussian distribution over (a set of points on) the mesh. Hence the networks output means μ and standard deviations σ w.r.t. each point's X, Y and Z coordinates. The nets can be considered Mixture Density Networks (MDN) [4] with multivariate output and single-modal distributions.

We experimented with three types of architectures: 1) nets composed of dense layers, 2) nets composed of a 3D convolutional encoder and an up-convolutional decoder, and 3) nets composed of a 3D convolutional encoder and a decoder composed of parallel multilayer perceptron (pMLP) layers. In the latter two architectures, the encoder reduces spatial resolution to 1x1x1, so latent encodings have no imposed spatial structure.

The pMLP layer consists of parallel instances of an MLP that takes the encoder's output and a set of UV coordinates as input, and outputs the XYZ coordinates for those UV coordinates. The pMLP layer is inspired by and similar to the Network in Network (NiN) layer [5].

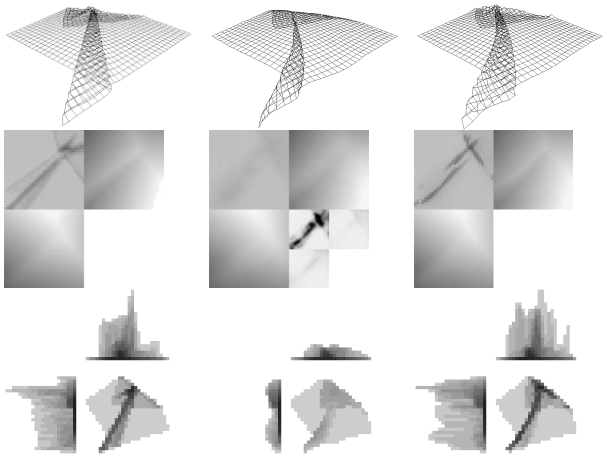

Figure 1. Representative mesh estimation and refinement result (dense architecture, test set). Top row: mesh visualisation of target, estimated μ, and refinement outcome. Middle row: vertex XYZ coordinates mapped as colour intensities in UV space for each of the meshes in the top column. Middle panel additionally shows estimated σ values. Bottom row: voxelisations (top-down and side views, with occlusion) of the meshes. The middle left image corresponds to the ground truth, the bottom left image to network input, and the middle centre image to network output.

Table 1: Architectures

	Neurons / feature maps per layer
dense	32×32×32, 4096, 4096, 4096, 4096, 32×32×3
conv & up-conv	encoder: 1, 32, 64, 128, 256, 512 fully-connected: 1024 decoder: 512, 256, 128, 64, 32, 3
conv & pMLP	encoder: 1, 32, 64, 128, 256, 512 decoder: 2050, 1026, 514, 258, 130, 66, 34, 3

The architectures are specified in Table 1. For all (up-)convolutional layers, kernel size is 3 and stride is 2. The architectures have different characteristics with respect to the sets of UV coordinates (mesh structures) that can be handled. An up-convolutional decoder assumes that points are arranged in a rectangular grid, which limits its expressiveness. In a dense architecture each point corresponds to one output neuron with no consideration of these points' spatial relations. This allows for arbitrary mesh structures, although we need to set the mesh structure before training. A pMLP decoder is most flexible. There is no correspondence between neurons and vertices. Instead, vertex UV coordinates are given as part of the input to each pMLP layer. This means that in principle, we can probe arbitrary UV coordinates at run time.

3. Dataset

We generate a dataset of cloth shapes in simulation using the ARCSim simulator [6]. Cloth shapes are generated

by initially laying a square cloth out flat and then iteratively performing one- and two-handed manipulations on it. Candidate grasp points are chosen by performing corner detection on the silhouette of the cloth viewed top-down. Movement trajectories are generated randomly and follow simple arcs. We generate training and test sets of 2399 and 300 examples, respectively.

During training, data is augmented by rotation and mirroring. Cloth states are centred on the X and Y axes, and converted to binary voxel representations of resolution 32×32×32. Z-coordinates are boosted to reduce loss of height detail. We artificially apply occlusion consistent with a single top-down view. Occupied and occluded voxels both take value 1, all other voxels take value 0.

4. Results

Quantitative results are given in Table 2. We report log-likelihood (*LL*, larger is better) and mean vertex distance between μ and the target (*error*, smaller is better) scores. For scale: the size of a single voxel is 0.0625×0.0625×0.0060. Evaluation excludes the flat cloth state, as it is both very easy to interpret and overrepresented in the dataset (due to each manipulation sequence starting from it).

Table 2: Quantitative results (simulation data)

set	measure	dense	conv & up-conv	conv & pMLP
test	*LL*	2.572	2.291	2.418
	error	.0282	0.0415	0.0380
train	*LL*	2.780	2.359	2.516
	Error	0.0238	0.0394	0.0357

We obtain our best performance with the dense architecture, followed by the pMLP architecture. We hypothesize that the relative strength of the dense architecture compared to the architectures with a convolutional encoder is related to the centring of the cloth in the voxel space. The convolution operator derives its practical value in large part from its translational invariance w.r.t. feature locations, but due to centring, key features occur at specific locations in the voxel space. To the extent that shape features (or their relevance) are location-specific, the convolution operator will be inefficient. However, the use of dense architectures will likely be infeasible for higher resolution inputs. Already for the present problem setting, the dense architecture is quite large in terms of trainable parameters (226.5M parameters) compared to the up-conv architecture (15.7M parameters) and the pMLP architecture (8.5M parameters). The finding that the pMLP architecture exceeds performance of a substantially larger up-convolutional architecture, while offering higher flexibility w.r.t. mesh structures, warrants further exploration. (We note that increasing the depth and width of the up-convolutional architecture to match the pMLP architecture did not improve performance.)

Network outputs were observed to generally capture cloth shape as follows: μ values describe the global shape of the cloth quite well, but details like creases are attenuated. Meanwhile, σ values are elevated in regions with fine detail not resolved in the μ values, and in occluded parts of the cloth, which is consistent with our expectations.

Given knowledge of the resting lengths of the edges of the mesh, we can refine the estimated mesh by optimising the vertex positions (initialised to μ) under a compound loss that expresses realism (consistency with edge resting lengths) and likelihood w.r.t. the distribution given by the network. This optimisation can reduce discrepancy in surface area between the real and estimated meshes by producing artificial fine detail (creases) in appropriate areas of the cloth. Figure 1 shows example estimation and refinement results.

Figure 2 shows example outcomes for a real cloth item. We do not have ground truths for our real-world data, but we find that for sufficiently simple shapes, plausible estimates can be obtained.

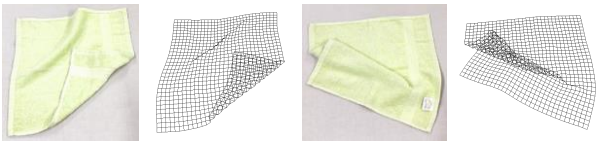

Figure 2. Examples of refined estimations for real data.

5. Future work

In future work we plan to extend our results to more complex meshes and real-world data, and integrate a mesh estimation network into our manipulation planning system [7].

References

[1] Bryan Willimon, Steven Hickson, Ian Walker, Stan Birchfield: An Energy Minimization Approach to 3D Non-Rigid Deformable Surface Estimation Using RGBD Data. IROS 2012.

[2] Tao Han, Xuan Zhao, Peigen Sun, Jia Pan: Robust Shape Estimation for 3D Deformable Object Manipulation. arXiv:1809.09802, 2018.

[3] Yinxiao Li, Yan Wang, Michael Case, Shih-Fu Chang, Peter K. Allen: Real-time Pose Estimation of Deformable Objects Using a Volumetric Approach. IROS 2014.

[4] Christopher M Bishop: Mixture density networks. Technical Report NCRG/4288, Aston University, Birmingham, UK, 1994.

[5] Min Lin, Qiang Chen, Shuicheng Yan: Network in Network. ICLR 2014.

[6] Rahul Narain, Tobias Pfaff, and James F. O'Brien: Folding and Crumpling Adaptive Sheets. ACM Transactions on Graphics, 32(4):51:1–8, July 2013. Proceedings of ACM SIGGRAPH 2013, Anaheim.

[7] Solvi Arnold, Kimitoshi Yamazaki: Fast and Flexible Multi-Step Cloth Manipulation Planning Using an Encode – Manipulate - Decode Network (EM*D Net). Frontiers in Neurorobotics. vol. 13, 2019.