Garment Ideation: Iterative View-Aware Sketch-Based Garment Modeling

Pinaki Nath Chowdhury[1, 3] Tuanfeng Wang[3] Duygu Ceylan[3] Yi-Zhe Song[1]
Yulia Gryaditskaya[1,2]

[1]SketchX, CVSSP, University of Surrey, United Kingdom
[2]Surrey Institute for People Centred AI and CVSSP, University of Surrey, United Kingdom
[3]Adobe Research, London, United Kingdom

Abstract

Designing real and virtual garments is becoming extremely demanding with rapidly changing fashion trends and increasing need for synthesizing realistically dressed digital humans for various applications. However, traditionally designing real and virtual garments has been time-consuming. Sketch based modeling aims to bring the ease and immediacy of drawing to the 3D world thereby motivating faster iterations. We propose a novel sketch-based garment modeling framework that is specifically targeted to synchronize with the iterative process of garment ideation, e.g., adding or removing details from different views in each iteration. At the core of our learning based approach is a view-aware feature aggregation module that fuses the features from the latest sketch with the thus far aggregated features to effective refine the generated 3D shape. We evaluate our approach on a wide variety of garment types and iterative refinement scenarios. We also provide comparisons to alternative feature aggregation methods and demonstrate favorable results. The code is available at https://github.com/pinakinathc/multiviewsketch-garment.

1. Introduction

Being one of the most natural mediums for humans to demonstrate ideas, sketch is widely used in design workflows. Specifically, fashion industry has a long tradition to start a design process with a sketch, and eventually convert it to 3D to demonstrate how it drapes over the body either by using physical patterns or virtual draping [1]. However, getting the 3D shape is not trivial and even more often the designer needs to iterate between the ideation, i.e., sketching from various viewpoints, and the 3D draping stages until the desired look is achieved. Our goal in this work is to provide an interactive solution that can make this iterative process more intuitive by mapping (multiple) input sketches into a 3D model. We believe that a successful solution should satisfy various properties: 1) View awareness: Allowing the designer to model the 3D garment from arbitrary views (instead of predefined fixed viewpoints such as frontal) offers an extra degree of freedom for the garment ideation. 2) Iterative refinement from different views: A single sketch may not capture all the geometric details (due to invisible regions and the inevitable ambiguity of 3D to 2D projection). Hence, the designer may want to iteratively update their design by providing different details from different viewpoints. 3) View-specific edits: When performing iterative refinements, it is expected that the latest sketch to be faithfully reflected in the 3D output, while keeping the previous edits in the invisible regions not affected if possi-

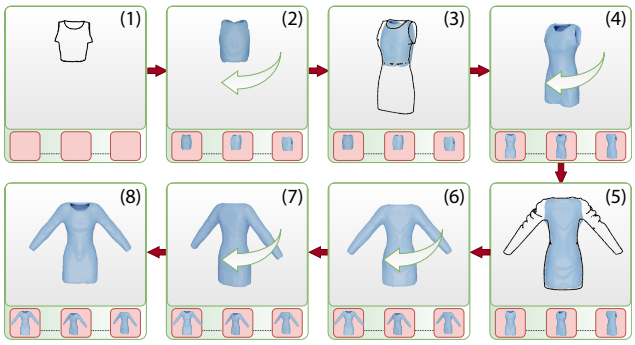

Figure 1. Our proposed system supports iterative multi-view garment design. In this example, the user starts by sketching in an arbitrary chosen view (1), and the garment geometry (2) is generated after the sketch. Next, the user rotates the reconstructed garment to sketch from a different viewpoint (3), and changes the design by augmenting it with a skirt (4). Exploring the design further from a different viewpoint, the user adds the sleeves in the back view (4). Our system efficiently updates the prediction, carefully matching it to the most recent view, while retaining details sketched in earlier views: compare the collar shape in views (1) and (8).

ble. For instance, while a newly added wrinkle in the front view of the skirt is not expected to change the back, an increase in the length of a sleeve results in a updating the 3D shape globally.

While there exist previous approaches [38, 6, 5, 8] for sketch based modeling, they do not satisfy the aforementioned properties of an iterative tool. To our knowledge, handling view conditioned editing is an unexplored area which we focus on in our work. At the core of our approach is a module to enforce view disentanglement in the feature space extracted from input sketches. Specifically, we propose a network that takes the sketch as input together with a view direction. We warp the features extracted from the sketch into a canonical space based on the viewpoint. When fusing the canonical features from multiple views, we use a binary mask which further encourages view disentanglement. Instead of averaging features from different views, we show that the binary mask helps to enforce view specific changes more effectively. We train our network on a dataset of garments composed of different parts (e.g., sleeves, collars etc.) to simulate changes that can be observed in a typical workflow (e.g., adding a sleeve from one view point). Finally, we utilize an implicit representation based on winding numbers [27] to represent the 3D geometry of the garments which we find to capture more geometric details compared to other representations such as occupancy fields.

We evaluate our method on a wide variety of garment types and show that our method enables to design a 3D garment in an iterative manner by providing sketches from various viewpoints (see Figure 1). We also provide comparisons with alternative approaches and demonstrate the effectiveness of our view disentangled feature fusion strategy.

In summary, our main contributions include (i) a sketch based garment modeling system that satisfies the needs of an iterative garment ideation tool, (ii) a novel view-aware feature aggregation strategy that faithfully reflects the changes in the last sketch in the 3D garment output.

2. Related Work

Sketch-based 3D shape modeling In our work, we explore Sketch-Based Modeling (SBM) for garment ideation and focus on multi-view iterative sketching. Since sketching is the most intuitive way of expressing a visual concept, there is a large interest in SBM for 3D content creation. For a detailed discussion, we refer the interested reader to several state-of-the-art reports on SBM [38, 6, 5, 8].

Since our goal is to enable an interactive garment ideation workflow, our primary requirement for such a workflow is to be fast, responsive and intuitive. Therefore, we follow the recent trend in SBM and leverage deep learning [36, 48, 33, 16, 47, 57, 55, 54, 23], which allows for fast and robust shape inference. Majority of the existing works focus on a single-view modeling [47, 23, 56] or assume that multiple views are given simultaneously [36, 33, 57, 25]. The closets to our work is the work by Delanoy et al. [16] who train a single-view CNN and an updater CNN to generate occupancy of a voxel grid from the sketch. The updater network takes as input the concatenation of the encoding of the current reconstruction and the embedding of the most recent view. We show that this approach provides much less control over the results compared to our method, as the network tends to 'forget' what was drawn in earlier views. In order to achieve a multi-view coherent solution, the updater proposed by Delanoy et al. repeatedly loops over all available drawings. In contrast, our network produces coherent results in a single pass, supports iterative design review, and does not modify non-edited view-specific garment features.

Various 3D representations were considered in the context of deep SBM, such as voxels [16], points clouds [55] and implicit shape representations [23]. In our work, we for the first time in the context of SBM use generalized winding numbers [28, 3, 10] to represent 3D garments.

Sketch-based garment modeling Dedicated garment sketch-based modeling systems often require users to draw on top of a 2D view of a predefined 3D mannequin, which provides additional context about the position of the drawn strokes in 3D space [45, 44, 41, 14]. While our system can potentially be extended to support drawing on top of a reference 3D body, this is not necessary and it enables free sketching in its current form.

Dedicated strategies were developed for modeling folds in garments [15, 30, 34, 20]. While our system captures main folds as shown in Figure 1, we are primarily focused on capturing large geometric features such as the appearance of the entire garment, the shape of the collar region, or the length and shape of the sleeves to help the ideation stages of the design process. Previously, Wang et al. [49] explored deep learning for garment modeling. Their method brings sketches, 2D patterns and draped 3D garments into a shared latent space. However, their model is garment template specific and the workflow supports single sketch input only. On the contrary, our framework is designed to support a wide variety of garments, allowing artists to iterate over their designs via multi-view sketching.

Image-based garment modeling Closely related to our work is image-based garment modeling, which has become popular in the last few years. The existing work [42, 2, 7] targets single and multi-view reconstruction, but unlike our work, assumes that the views are fully consistent. The image-specific solution [2] uses a texture map as a bridge between the image domain and the 3D shape domain, effectively capturing normal changes along the garment surface. However, such solutions cannot be applied to sketch input.

Several methods [53, 13, 29] rely on existence of specific cloth templates, which limits the generalization of these approaches to arbitrary garments. The latter is essential for a sketch-based garment ideation system. Since many of these works are aimed at virtual try-on applications, garments are often modeled with respect to a human body. Thus, a common approach is to represent garments as offsets from the body mesh [4, 24, 37] which is limited to represent loose garments. To overcome this limitation, Saito et al. [42] use occupancy fields, an implicit geometry representation, to predict the surface of a human from single or multiple input images. Corona et al. [12] exploit unsigned distance fields and propose to encode the distance of a 3D point relative to a set of point samples from a body template. After experimenting with different representations, we have selected winding numbers to represent the 3D shape. We observe that such representation provides better reconstruction quality than alternative implicit shape representations, and avoids the problem of reconstructing double surfaces, as is the case for unsigned distance fields (see the supplementary material).

Multi-view image-based 3D shape modeling Single and multi-view image-based modeling is a popular research topic in vision and graphics and more recently learning based approaches have shown impressive results. For a detailed overview, we refer the reader to the respective state-of-the-art reports [21, 51]. Here, we discuss a few representative approaches to demonstrate why they are not directly applicable to our problem. A common approach for learning-based multi-view reconstruction is to employ an RNN based architecture [31] to fuse information across images. However, such approaches do not allow for view-disentanglement, and RNNs tend to 'forget' the important features from early inputs. Another approach is to use max pooling [26] or average pooling [39] to fuse multi-view features. However, these are not suitable for inconsistent views, which is an important requirement for a system with iterative design refinement, as shown in Figure 1. A more similar idea is to use attention-based fusion [52, 50, 46, 58]. However, these methods are tailored to handle order-invariance for the input views. In contrast, in our setting the order of the sketches is critical since the geometry depicted in the last sketch should result in expected updates. The closest to our work is the method [52], which combines views according to the learned attention vectors. In contrast to their work, we propose quantized attention vectors additionally conditioned on camera directions. This enables view-disentanglement required for our system. In our experiments, we provide a detailed comparison to these previous approaches and demonstrate the effectiveness of our method.

3. Method

Our method takes a single input sketch at a time and generates the corresponding 3D garment shape either by providing an initial 3D prediction (in case of the first sketch input) or updating the existing 3D shape prediction. Specifically, we extract features from the input sketch which are then decoded for 3D prediction. At the core of our method is a feature aggregation mechanism that fuses the features from the last sketch with the aggregated features thus far using a view-aware binary weighting mechanism. We show that this approach faithfully reflects the view-specific edits provided in the last sketch in the generated 3D output. Next, we first describe the overall network architecture (Figure 2) for a single input sketch, and then explain how we perform feature aggregation across a sequence of sketches.

3.1. Architecture

3.1.1 View-aware sketch encoder

In order to effectively aggregate information across sketches drawn from different viewpoints, we aim to map each sketch to a canonical feature space. We achieve this goal by first obtaining view specific features from the input sketch by using an image encoder pre-trained on a large collection of images. Next, we propose a simple network module, AlignNet to bring such features to a canonical space.

Specifically, starting from a sketch S, we extract features $f \in \mathbb{R}^{512}$ using a pretrained image encoder. We adopt the VGG-16 [43] network pre-trained on ImageNet[1]. We discard the top layers starting with the global average pooling and apply adaptive pooling to obtain a $512 \times 1 \times 1$ feature vector, which we then flatten to obtain $f \in \mathbb{R}^{512}$.

The AlignNet takes as input the feature vector f and a positional encoding of a view ϕ to generate a canonical-space feature along with blending weights:

$$f_{align}, \alpha = \mathbf{AlignNet}(f, \phi), \tag{1}$$

where $f_{align} \in \mathbb{R}^{512}$. The blending weights $\alpha \in \mathbb{R}^{512}$ indicate which channels in f_{align} are informative according to the view direction ϕ. In case of sequential sketch inputs, α is utilized to fuse f_{align} with the aggregated features obtained in the previous step.

In AlignNet, the concatenated features of f and ϕ are first passed through two linear layers, each followed by a $ReLU(\cdot)$ activation function to obtain a feature vector $x \in \mathbb{R}^{1024}$. We then split this vector into two: $x^{1:512}$ and $x^{512:1024}$, which are passed through two separate sub-networks to obtain $f_{align} \in \mathbb{R}^{512}$ and $\alpha \in \mathbb{R}^{512}$, respectively. Each sub-network consists of a linear layer followed by a $ReLU(\cdot)$, and a second linear layer.

[1] https://www.image-net.org/

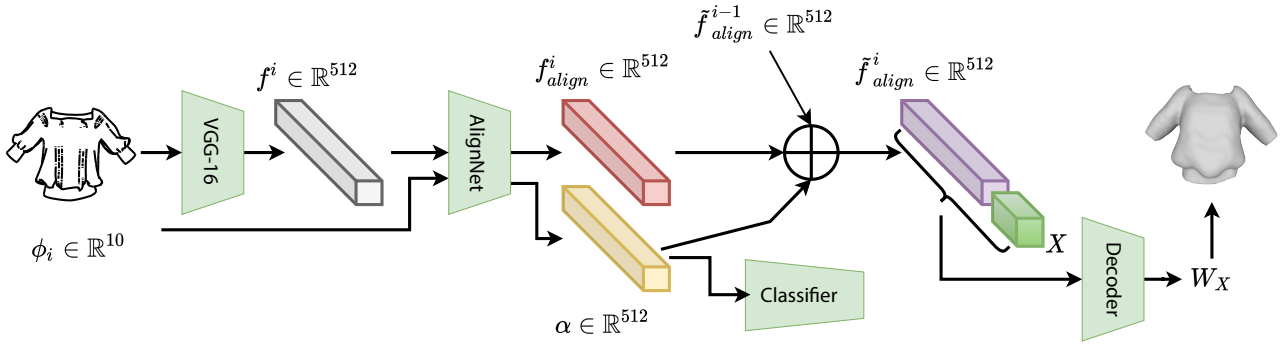

Figure 2. Starting from the input sketch and the corresponding view direction ϕ_i, we extract image feature f^i and align it to canonical domain f^i_{align}. Blending vector α is produced simultaneously to fuse the aligned feature with the feature obtained from the last step. For a given spatial location X, we concatenate it with the fused feature $\tilde{f}^i_{align}$ and pass through our decoder to predict the winding number at position X. The final 3D garment mesh will be extracted based on the densely sampled winding number field.

In our implementation, without loss of generality, we assume that the elevation angle is fixed and we vary only the camera azimuth angle. Therefore, in our implementation, the view encoding ϕ represents the camera azimuth angle.

3.1.2 View classification

To encourage the learned blending vector $\alpha \in \mathbb{R}^{512}$ to preserve view information, we introduce a classifier C_{view} during training which takes α as input and classifies the input view.

To enable view-aware feature extraction, we would like our weighting vector α to be binary so that the features extracted from a particular view affect only certain dimensions of the feature vector. Strictly enforcing binary values while still being able to back-propagate is not straightforward. We propose to use a *HardShrink* activation [40] with $\lambda = 1$ before providing α to C_{view}. This step enables us to easily obtain a binary counterpart of α, which we use for multi-view feature blending in Section 3.2.

Our view classifier network C_{view} consists of a linear layer with 512 output neurons, batch normalisation, $ReLU(\cdot)$, and a linear layer that reduces the number of output neurons to 360, followed by a $\mathrm{softmax}(\cdot)$ activation function to predict a classification probability.

3.1.3 Sketch decoder

We represent a 3D shape using generalized winding numbers [28, 3, 10]. Our decoder network takes in the concatenation of the spatial location $X \in \mathbb{R}^3$ and the view encoded in the canonical feature space $\tilde{f}_{align}$, and predicts a winding number W_X.

The decoder consists of 8 linear layers followed by a $ReLU(\cdot)$, each with hidden dimension of 512, except for the last layer, which predicts a scalar winding number.

During training, we sample $8,192$ randomly selected locations inside a pre-computed grid of resolution $128 \times 128 \times 128$. During inference, we regularly sample the 3D grid and generate the winding number field accordingly. Marching cubes algorithm [35] is applied to the numerical gradient of the winding number field to recover the 3D geometry surface. In our experiments, we adopt a threshold equal to 0.3.

3.2. Feature fusion

When the user provides a new sketch for an existing design, the canonical feature from the latest sketch f^i_{align} needs to be merged with the features obtained in the last step $\tilde{f}^{i-1}_{align}$. We desire the blending weights to be binary so that they enable the relevant channels of the feature vector to be directly affected by the latest sketch. Hence, given the predicted α^i by AlignNet, we convert α^i into a strictly binary signal, α^i_*, as:

$$\alpha^i_* = \begin{cases} 1 & \text{if } \alpha^i > \lambda \\ 0 & \text{if } otherwise \end{cases} \tag{2}$$

Then our feature blending is

$$\tilde{f}^i_{align} = \alpha^i_* \cdot f^i_{align} + (1 - \alpha^i_*) \cdot \tilde{f}^{i-1}_{align}. \tag{3}$$

Our binary activation α^i_* ensures that the learned feature will fully respect the latest sketch input S_i in the channels visible from view ϕ_i. This also resolves the issue of repetitive input, i.e., if S_i and S_{i-1} is similar on the overlapped region (if any), our feature will have consistent value on the corresponding channels.

3.3. Loss

We train our network with the two loss terms.

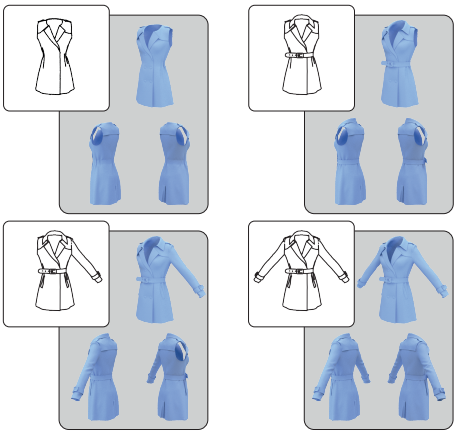

Figure 3. Example garment variations created by adding or removing garment parts. Using such variations for training allows us to meet the requirement of iterative refinement of a garment concept.

View classification loss A typical cross entropy classification loss is used to train our view classifier C_{view}. Specifically, we have:

$$\mathcal{L}_{vc} = -\phi^i \log[C_{view}(\alpha^i)] \tag{4}$$

Geometry loss We apply L_1 loss on the predicted winding number for geometry reconstruction.

$$\mathcal{L}_{geo} = |w(X) - \mathbf{Decoder}(\tilde{f}^i_{align}, X)|_1 \tag{5}$$

where $w(X)$ is the winding number at location X for the ground truth geometry. We note that $\mathcal{L}_{geo}$ is not back-propagated to the α prediction branch of AlignNet due to the strict binarization described in Section 3.2. Hence, $\mathcal{L}_{vc}$ is crucial to train blending weights α.

3.4. Implementation details

Our model is implemented in PyTorch using 11GB Nvidia RTX 2080-Ti GPUs. In each iteration, we sample 3 random sketch views for each garment in a batch. Our model is trained using Adam Optimiser [32] for 1 million iterations with batch size 8. Typically it takes 96 hours for the model to converge.

4. Experiments

Dataset We use the dataset provided by Chen *et al.* [9] which contains 240 unique garment models, containing top, bottom and full body garments. Each garment consists of several components, such as collar, sleeve, etc. We exploit this property to obtain a larger dataset by removing components as shown in Figure 3 to simulate rich variety of garment designs, which is in line with our goal of supporting iterative design evolution. In total, we obtain 2,158 3D garment models. We split them into 1,235 garment models used for training and 923 models used for testing. The splitting is done in such a way that the test set does not contain any variant of the garment from the training set. Following [10], we compute winding numbers for each garment model variant on a regular grid with spatial resolution of $128 \times 128 \times 128$.

NPR We use Non-Photorealistic Rendering (NPR) to generate the dataset of garment sketches. We render silhouette and open edge lines into 224×224 sketch views with Blender Freestyle [11], using an orthographic projection. The camera points at the center of the garment. We fix the camera elevation angle to 10° and render garment views by orbiting around the garment with 1° step. In total, we generate 360 sketch views for each garment.

Evaluation metrics We evaluate the quality of 3D reconstructions using the ubiquitously used Chamfer distance (CD) [19] metric. We uniformly sample 5,000 points from ground-truth and reconstructed 3D shapes and compute the distance between them. A smaller CD value indicates more accurate reconstruction.

To evaluate view disentanglement, we also compute view-based metrics. We render normal maps of a 3D geometry from the specified viewpoints. Next, we evaluate the structural similarity index (SSIM) between the predicted and ground-truth shape projection. A larger SSIM value indicates more accurate reconstruction.

4.1. Garment ideation workflow

In Figures 1 and 4, we demonstrate the example garment ideation workflow scenarios that our system enables. In particular, the user can start sketching from an arbitrary viewpoint and the network will generate a plausible 3D garment that matches the sketch from a given viewpoint. In each new sketch view, the user may choose to either (i) add additional details to refine the current prediction, or (ii) update the garment design by adding or removing some details, for instance, by adding a 'skirt' or 'sleeve'. Our system can update the prediction from the current view without modifying the prediction for the invisible regions that were sketched in the previous views. For instance, as is shown in Figure 4 (b), removing a skirt in the back view, does not affect the shape of a collar in the front view. Moreover, if the applied edit is global, such as removing a skirt in this example, our model updates this information globally while retaining the details that were sketched previously but are not visible in the current view.

4.2. Quantitative evaluation

4.2.1 Baselines

Due to the lack of methods that focus on iterative sketch-based modeling, we create three baselines adopting multi-view image reconstruction methods and the approach used by Delanoy et al. [17].

Namely, as the first baseline, **B-RNN**, we adapt the RNN-based fusion method [31]. Each view is encoded with

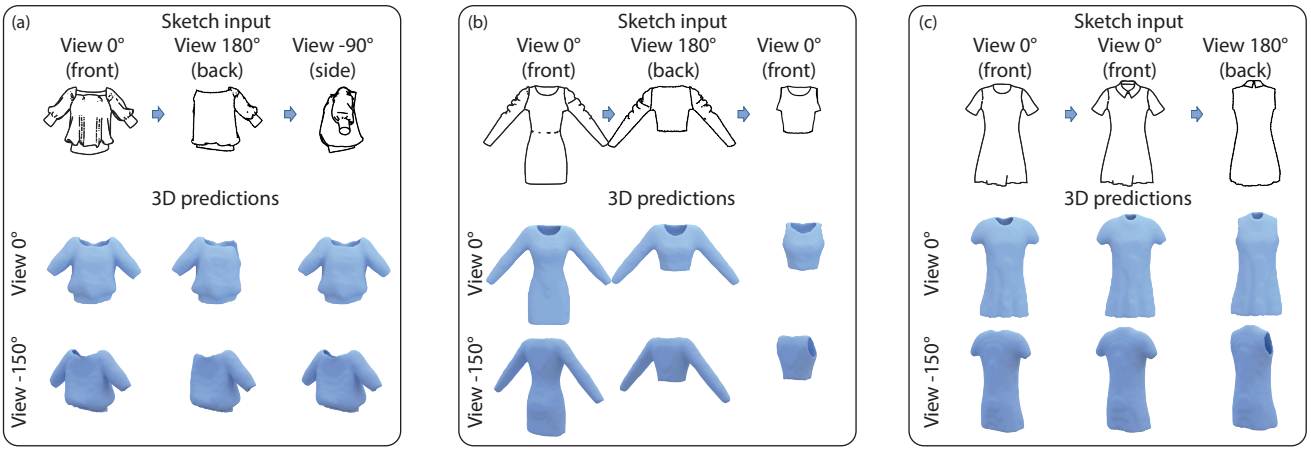

Figure 4. Garment design iterative editing. (a) The user sketches in the front view, then removes a sleeve in the back view, and then adds it back in the side view. (b) The user sketches a dress in the front view, then removes a skirt in the back view, and then removes sleeves in the front view. (c) The user sketches a dress in the front view, then updates the collar region in the same view, and then removes sleeves in the back view. Please note how all examples demonstrate that only information updated in the most recent view is updated in the 3D predictions.

our sketch encoder to obtain f_{align}, note that we do not predict any weighting vector in this case. In our implementation, we use a single layer RNN with hidden size of 512.

B-Concat follows the startegy proposed by Delanoy et al. [17]. The fused features from the last step $\tilde{f}^{i-1}_{align} \in \mathbb{R}^{512}$ are concatenated with the canonical feature from the latest sketch $f^{i}_{align} \in \mathbb{R}^{512}$ and are passed through a linear layer to obtain a new fused feature vector $\tilde{f}^{i}_{align}$.

B-Cont-α use AlignNet to predict features in the canonical space and continuous weight vectors. However, since continuous α is able to pass the gradient during training, the view classifier is not necessary in this case and the network is trained only with the geometry loss $\mathcal{L}_{geo}$. Unlike Equation (3), we perform feature fusion using softmax activation on continuous weighting vectors:

$$\begin{aligned} \alpha^i_s &= \text{softmax}(\alpha^i) \\ \tilde{f}^i_{align} &= \alpha^i_s \cdot f^i_{align} + (1 - \alpha^i_s) \cdot \tilde{f}^{i-1}_{align} \end{aligned} \tag{6}$$

4.2.2 Single-View Reconstruction

First, we show that our model produces plausible predictions for single view input and is robust to different viewpoints. Some reconstruction results are shown in Figures 4 and 5. In Table 1, we evaluate single-view reconstruction accuracy from several distinctive viewpoints: 0°, 30°, 60°, 90° and 180°. It can be observed that the highest reconstruction accuracy is achieved from the 30° and 60° views, which are often referred to as the most informative views in the sketching literature [18, 22]. Side views contain little information due to foreshortening. Similarly the back view lacks details that are typically located in the frontal part of a garment.

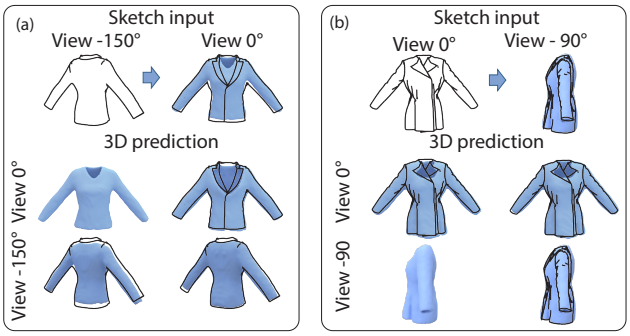

Figure 5. Multi-view Reconstruction: We show the reconstruction result when iteratively feeding 2D sketches from different viewpoints. Note, the network is capable of predicting a visually plausible geometry given a single sketch. Adding multiple sketches makes the underlying geometry more accurate.

Table 1. Chamfer distance values for single view prediction results for different viewpoints.

Method	Sketch view angle				
	Front 0°	... 30°	... 60°	Side 90°	Back 180°
Proposed	0.288	0.244	0.357	0.855	0.824

4.2.3 Multi-View Reconstruction

Although single-view reconstruction generates plausible geometry matching the input view, garment ideation might

Table 2. Quantitative evaluation using the Chamfer distance (in 10^{-2}) for multi-view reconstruction. It can be seen that adding more views reduces the Chamfer distance for all feature fusion methods. Please see Section 4.2.3 for a detailed discussion.

	1st Sketch 1-view	2nd Sketch 2-views	3rd Sketch 3-views
B-RNN	0.308	0.176	0.134
B-Concat	0.305	0.169	0.135
B-Cont-α	**0.284**	**0.159**	**0.125**
Proposed	0.288	0.160	**0.125**

require concept exploration and refinement from different viewpoints [17, 49].

Consistent geometry When multiple views are sketched to provide more details on a consistent design, adding more sketches from different viewpoints should result in more accurate geometry. Figure 5 demonstrates this behaviour: (a) the shape of the collar is refined and overall shape prediction is more accurate, and (b) the prediction of the sleeve is getting more accurate in the side view. We evaluate this quantitatively in Table 2, measuring Chamfer distance between the prediction results and the ground-truth when progressively more views are used. Indeed, for all fusion strategies the Chamfer distance reduces when additional views that are consistent in terms of geometric details are provided. Moreover, it can be observed that ours and **B-Cont-α** fusion approaches achieve the best performance. Our binary weighting strategy performs just slightly worse than **B-Cont-α**, but allows for the view-disentanglement that we demonstrate in the next section.

View disentanglement Figure 5 (b) shows that sketching in the side view does not change the shape of the collar sketched in the first view, and not visible in the side view. We quantitatively evaluate view-disentanglement in Table 3. First, we provide a sketch obtained from the front view to the network followed by a sketch obtained from the back view in the second iteration. We evaluate the SSIM (Structure Similarity Index) difference between the normal maps of the prediction and ground-truth 3D garments from the front and back views. It can be observed that our strategy is the most efficient in preserving the information from the first view and updating the information given the most recent view. With our fusion strategy, the reconstruction quality of the front view almost does not change, while the back view is improved significantly. This shows that our design meets the requirements on updating the geometry only with respect to what is visible in the most recent view.

An important aspect of a convenient system for garment ideation is an instant update of a 3D geometry to reflect the latest changes in the user sketch. Figure 4 demonstrates a

Table 3. Quantitative evaluation of view-disentanglement. A sketch drawn from the Front View (Inp. View-1 (FV)) is given to the network followed by a sketch drawn from the Back View (Inp. View-2 (BV)) in the second iteration. We measure the SSIM of the normal maps from the front (GT-FV) and back view (GT-BV) between the predicted and ground-truth meshes.

	Inp. View-1 (FV)		Inp. View-2 (BV)	
	GT-FV	**GT-BV**	Δ **GT-FV**	Δ GT-BV
B-RNN	0.9722	0.9724	-0.0003	0.0001
B-Concat	0.9725	0.9724	-0.0008	0.0003
B-Cont-α	0.9728	0.9730	-0.0007	0.0007
Proposed	**0.9728**	**0.9730**	**-0.0001**	**0.0013**

Table 4. Quantitative evaluation using Chamfer Distance to measure the adaptability to the latest sketch.

	No-Sleeve	One-Sleeve	Both-Sleeves
B-RNN	0.308	0.411	0.358
B-Concat	0.305	0.372	0.348
B-Cont-α	**0.284**	0.335	0.302
Proposed	0.288	**0.292**	**0.289**

number of application scenarios. In Table 4, we perform a quantitative evaluation of our approach. We select from our test set all garments that have sleeves (47 garments). At first iteration, we predict a 3D shape from the front view with both sleeves sketched. At second iteration, we predict a 3D shape from the front view with only one sleeve sketched. Finally, at the third iteration, we predict a 3D shape when again both sleeves are sketched. In each case we update the ground-truth geometry to reflect the changes in the sketch. We compute the Chamfer distance at each iteration between the predicted and ground-truth geometries. Table 4 shows that our fusion strategy is by far the most efficient in updating 3D geometry with respect to the most recent sketch. Figure 4 shows that our system supports diverse user edits and provides a good preview of the envisioned 3D geometry. We provide additional qualitative comparisons in the supplementary material.

4.2.4 Latent space analysis

In this section, we analyze our latent space. The $\mathbf{AlignNet}(\cdot)$ module used in our proposed method predicts the encoding of the input sketch into the canonical latent space f_{align} and a weighting vector α that we use to update the prediction given the most recent view.

Feature vectors in the canonical space First, we analyze the ability of our network to 'align' feature vectors obtained from different sketch views of the same garment. We visualize the t-SNE plot of f_{align} in Figure 6 (a). We represent the encodings obtained from different views of the same garment with the same designated color.

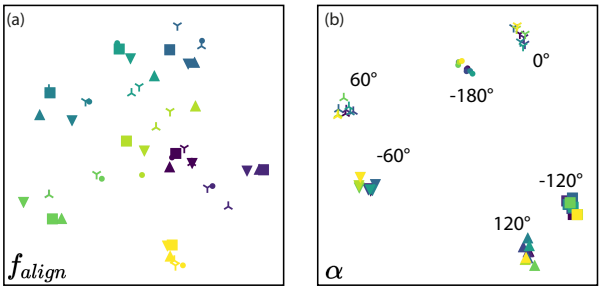

Figure 6. tSNE plot visualising f_{align} and α from 5-garments with 6-views $(0°, 60°, 120°, 180°, 240°, 300°)$ from each garment. Multiple views from the same garment are depicted using the same color and different views from each garment are depicted using different marker shapes.

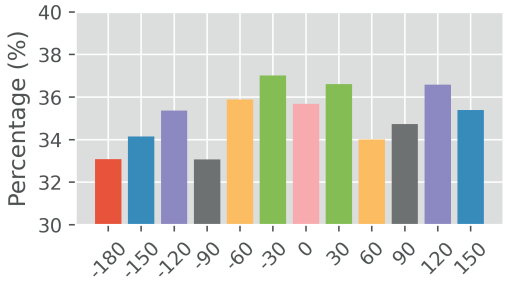

Figure 7. We select a subset of sketch viewpoints ϕ uniformly distributed around 3D garments. For each weighting vector α_*, we compute the percentage of vector components that are equal to 1. The figure shows the average percentage across different garments for each selected viewpoint.

To obtain the plot we randomly select 5 garments. For each garment we select six sketches from six viewpoints: $(0°, 60°, 120°, 180°, 240°, 300°)$. The encodings obtained from each specific view is shown with a distinctive shape. It can be observed that f_{align} from different viewpoints of the same garment (having the same color) cluster together, as shown in Figure 6 (a). Such alignment of feature vectors is crucial for efficient feature fusion.

Weighting vectors We visualize the t-SNE plot of α in Figure 6 (b). It can be seen that weighting vectors α estimated for different garments but same viewpoints (having the same marker shape) cluster together. Moreover, we observe that view $0°$ and view $180°$ are encoded closer together than remaining views. This is intuitive as for some garments front and back views can be quite visually similar due to symmetry. This observation motivates the use of the positional encoding ϕ as an additional input to the $AlignNet(\cdot)$ module.

In addition, our representation of weighting vector in its binary form α_* allows us to study how informative each viewpoint is. In other words, we analyze if the form of the weighting vectors for different views can bring some insights into which views contain more cues about 3D garment geometry. In Figure 7, we select a subset of sketch viewpoints uniformly distributed around 3D garments, and count how many components of predicted α_*, on average, are equal to 1 per each view, across different garments. It can be observed that back view can be considered the least informative, which can be explained by the lack of details in the typical garment in our training dataset. Similarly, side views are also considered less informative by our network, which is also intuitive as these views are the most foreshortened. As many garments are symmetric, it can be observed that the learned weighting vectors exhibit symmetry with respect to the front view.

5. Limitations and future work

There are various avenues for future improvements. To leverage the challenging target domain of garments that contain thin open surfaces we adopt winding number shape representation. However, it was observed by [10] that unsigned distance fields might be better in capturing small details even though they result in thick double surface reconstructions. In the future, we would like to explore different representations to allow for accurate reconstruction of small details. Another limitations of our system is the need to create a full sketch from every new viewpoint. In future work, we would also like to extend our method to support spatial disentanglement to enable local edits in each view.

6. Conclusion

In our work, we for the first time consider the problem of iterative multi-view sketch-based garment ideation. We identified the desirable properties of the system that can support iterative garment design workflow. Namely, such a system should be able to (i) produce a plausible prediction from a single view input, (ii) efficiently aggregate the information from multi-view sketch inputs, and (iii) update the design with respect to the most recent view, while preserving the details which were sketched in the previous views and are not visible in the current view. We take inspiration from the literate on multi-view image-based modeling and propose an architecture that learns a disentangled latent space. We achieve this disentanglement by learning binary weighting vectors that indicate which part of the sketch view feature vector can be reliably predicted from a given input view. We demonstrate the efficiency of our system across various design scenarios and garment types. We consider several alternative designs of multi-view feature fusion strategies and demonstrate the superiority of our design. We believe that our system will be of high interest to a design community, where iteration between sketching and 3D modeling is one of the most time consuming steps.

References

[1] Optitext fashion design software. https://optitex.com. Accessed: 2022-06-02. 1

[2] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *ICCV*, pages 2293–2303, 2019. 2

[3] Gavin Barill, Neil G Dickson, Ryan Schmidt, David IW Levin, and Alec Jacobson. Fast winding numbers for soups and clouds. *ACM. Trans. Graph.*, 37(4):1–12, 2018. 2, 4

[4] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *ICCV*, pages 5420–5430, 2019. 3

[5] Sukanya Bhattacharjee and Parag Chaudhuri. A survey on sketch based content creation: from the desktop to virtual and augmented reality. In *Computer Graphics Forum*, volume 39, pages 757–780. Wiley Online Library, 2020. 2

[6] Alexandra Bonnici, Alican Akman, Gabriel Calleja, Kenneth P Camilleri, Patrick Fehling, Alfredo Ferreira, Florian Hermuth, Johann Habakuk Israel, Tom Landwehr, Juncheng Liu, et al. Sketch-based interaction and modeling: Where do we stand? *AI EDAM*, 33(4), 2019. 2

[7] Akin Caliskan, Armin Mustafa, Evren Imre, and Adrian Hilton. Multi-view consistency loss for improved single-image 3d reconstruction of clothed people. In *ACCV*, 2020. 2

[8] Jorge D Camba, Pedro Company, and Ferran Naya. Sketch-based modeling in mechanical engineering design: Current status and opportunities. *Computer-Aided Design*, page 103283, 2022. 2

[9] Xiaowu Chen, Bin Zhou, Feixiang Lu, Lin Wang, and Lang Bi. Garment modeling with a depth camera. *ACM Trans. Graph.*, 2015. 5

[10] Cheng Chi and Shuran Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. In *ICCV*, 2021. 2, 4, 5, 8

[11] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, 2018. 5

[12] Enric Corona, Albert Pumarola, Guillem Alenya, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *CVPR*, pages 11875–11885, 2021. 3

[13] R Daněřek, Endri Dibra, Cengiz Öztireli, Remo Ziegler, and Markus Gross. Deepgarment: 3d garment shape estimation from a single image. In *Computer Graphics Forum*, volume 36, pages 269–280. Wiley Online Library, 2017. 3

[14] Chris De Paoli and Karan Singh. Secondskin: sketch-based construction of layered 3d models. *ACM. Trans. Graph.*, 34(4):1–10, 2015. 2

[15] Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. Virtual garments: A fully geometric approach for clothing design. In *Computer Graphics Forum*, volume 25, pages 625–634. Wiley Online Library, 2006. 2

[16] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *ACM on Computer Graphics and Interactive Techniques*, 1(1):1–22, 2017. 2

[17] Johanna Delanoy, Mathieu Aubry, Phillip Isola, Alexei A. Efros, and Adrien Bousseau. 3d sketching using multi-view deep volumetric prediction. *CGIT*, 2018. 5, 6, 7

[18] Koos Eissen and Roselien Steur. Sketching: the basics; the prequel to sketching: drawing techniques for product designers. *BIS, Amsterdam. OCLC*, 756275344, 2011. 6

[19] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613, 2017. 5

[20] Amélie Fondevilla, Damien Rohmer, Stefanie Hahmann, Adrien Bousseau, and Marie-Paule Cani. Fashion transfer: Dressing 3d characters from stylized fashion sketches. In *Computer Graphics Forum*, volume 40, pages 466–483. Wiley Online Library, 2021. 2

[21] Kui Fu, Jiansheng Peng, Qiwen He, and Hanxiao Zhang. Single image 3d object reconstruction based on deep learning: A review. *Multimedia Tools and Applications*, 80(1):463–498, 2021. 3

[22] Yulia Gryaditskaya, Mark Sypesteyn, Jan Willem Hoftijzer, Jan Willem Hoftijzer, Sylvia Pont, Fredo Durand, and Adrien Bousseau. Opensketch: A richly-annotated dataset of product design sketches. *ACM. Trans. Graph.*, 2019. 6

[23] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua. Sketch2mesh: Reconstructing and editing 3d shapes from sketches. In *ICCV*, 2021. 2

[24] Erhan Gundogdu, Victor Constantin, Shaifali Parashar, Amrollah Seifoddini Banadkooki, Minh Dang, Mathieu Salzmann, and Pascal Fua. Garnet++: Improving fast and accurate static 3d cloth draping by curvature loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[25] Zhizhong Han, Baorui Ma, Yu-Shen Liu, and Matthias Zwicker. Reconstructing 3d shapes from multiple sketches using direct shape optimization. *IEEE Transactions on Image Processing*, 29:8721–8734, 2020. 2

[26] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, pages 2821–2830, 2018. 3

[27] Alec Jacobson, Ladislav Kavan, and Olga Sorkine. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph.*, 32(4), 2013. 2

[28] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM. Trans. Graph.*, 32(4):1–12, 2013. 2, 4

[29] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *ECCV*, pages 18–35. Springer, 2020. 3

[30] Amaury Jung, Stefanie Hahmann, Damien Rohmer, Antoine Begault, Laurence Boissieux, and Marie-Paule Cani. Sketching folds: Developable surfaces from non-planar silhouettes. *ACM. Trans. Graph.*, 34(5):1–12, 2015. 2

[31] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NeurIPS*, pages 364–375, 2017. 3, 5

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[33] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM. Trans. Graph.*, 37(6):1–12, 2018. 2

[34] Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. Foldsketch: enriching garments with physically reproducible folds. *ACM. Trans. Graph.*, 37(4):1–13, 2018. 2

[35] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 4

[36] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3D shape reconstruction from sketches via multi-view convolutional networks. In *3DV*, 2017. 2

[37] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *CVPR*, pages 6469–6478, 2020. 3

[38] Luke Olsen, Faramarz F Samavati, Mario Costa Sousa, and Joaquim A Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009. 2

[39] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *CVPR*, pages 3897–3906, 2018. 3

[40] PyTorch. HardShrink Activtion. https://pytorch.org/docs/stable/generated/torch.nn.Hardshrink.html, 2019. [Online; accessed 30-May-2022]. 4

[41] Cody Robson, Ron Maharik, Alla Sheffer, and Nathan Carr. Context-aware garment modeling from sketches. *Computers & Graphics*, 35(3):604–613, 2011. 2

[42] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *CVPR*, pages 2304–2314, 2019. 2, 3

[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[44] Emmanuel Turquin, Jamie Wither, Laurence Boissieux, Marie-Paule Cani, and John F Hughes. A sketch-based interface for clothing virtual characters. *IEEE Computer graphics and applications*, 27(1):72–81, 2007. 2

[45] Charlie CL Wang, Yu Wang, and Matthew MF Yuen. Feature based 3d garment design through 2d sketches. *Computer-Aided Design*, 35(7):659–672, 2003. 2

[46] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z Jane Wang, and Rabab Ward. Multi-view 3d reconstruction with transformer. In *ICCV*, 2021. 3

[47] Jiayun Wang, Jierui Lin, Qian Yu, Runtao Liu, Yubei Chen, and Stella X Yu. 3d shape reconstruction from free-hand sketches. *arXiv preprint arXiv:2006.09694*, 2020. 2

[48] Lingjing Wang, Cheng Qian, Jifei Wang, and Yi Fang. Unsupervised learning of 3D model reconstruction from hand-drawn sketches. In *ACM MM*, 2018. 2

[49] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 2018. 2, 7

[50] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: multi-scale context-aware 3d object reconstruction from single and multiple images. *IJCV*, 128(12):2919–2935, 2020. 3

[51] Xiaoqiang Yan, Shizhe Hu, Yiqiao Mao, Yangdong Ye, and Hui Yu. Deep multi-view learning methods: A review. *Neurocomputing*, 448:106–129, 2021. 3

[52] Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction. *IJCV*, 128(1):53–73, 2020. 3

[53] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. Physics-inspired garment recovery from a single-view image. *ACM. Trans. Graph.*, 37(5):1–14, 2018. 3

[54] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR*, 2021. 2

[55] Yue Zhong, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. Deep sketch-based modeling: Tips and tricks. In *3DV*, pages 543–552. IEEE, 2020. 2

[56] Yue Zhong, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. A study of deep single sketch-based modeling: View/style invariance, sparsity and latent space disentanglement. *Computers & Graphics*, 106:237–247, 2022. 2

[57] Yue Zhong, Yonggang Qi, Yulia Gryaditskaya, Honggang Zhang, and Yi-Zhe Song. Towards practical sketch-based 3d shape generation: The role of professional sketches. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 2

[58] Pierre Zins, Yuanlu Xu, Edmond Boyer, Stefanie Wuhrer, and Tony Tung. Data-driven 3d reconstruction of dressed humans from sparse views. In *3DV*, pages 494–504. IEEE, 2021. 3