Creative Birds: Self-Supervised Single-View 3D Style Transfer

Renke Wang1*, Guimin Que1*, Shuo Chen2†, Xiang Li3, Jun Li1†, Jian Yang1

1PCA Lab, Nanjing University of Science and Technology, China  
2RIKEN, 3Nankai University

Abstract

In this paper, we propose a novel method for single-view 3D style transfer that generates a unique 3D object with both shape and texture transfer. Our focus lies primarily on birds, a popular subject in 3D reconstruction, for which no existing single-view 3D transfer methods have been developed. The method we propose seeks to generate a 3D mesh shape and texture of a bird from two single-view images. To achieve this, we introduce a novel shape transfer generator that comprises a dual residual gated network (DRGNet), and a multi-layer perceptron (MLP). DRGNet extracts the features of source and target images using a shared coordinate gate unit, while the MLP generates spatial coordinates for building a 3D mesh. We also introduce a semantic UV texture transfer module that implements textural style transfer using semantic UV segmentation, which ensures consistency in the semantic meaning of the transferred regions. This module can be widely adapted to many existing approaches. Finally, our method constructs a novel 3D bird using a differentiable renderer. Experimental results on the CUB dataset verify that our method achieves state-of-the-art performance on the single-view 3D style transfer task. Code is available at https://github.com/wrk226/creative_birds.

1. Introduction

Neural image style transfer has received increasing attention in computer vision communities due to its remarkable success in many automatic creations. These applications mainly belong to a 2D style transfer, which transfers the artistic style of one reference image to another content image. Recently, it is extended to transfer the shape and texture style of one 3D object to another for editing 3D content in augmented reality and virtual reality [41]. However, this extension needs to acquire the 3D information on objects which suffers from the significant problem of being arduous, high-cost, and time-consuming. In contrast, taking a single-view object image is easier and cheaper than 3D data as cameras (e.g., mobile phones) are used widely in everyday life. Therefore, we address a new task of generating a 3D object with novel shape and texture by transferring one single-view object image to another, instead of acquiring 3D input information. We refer to this task as a single-view 3D style transfer.

Prior works have primarily handled either 3D style transfer or single-view 3D reconstruction but have not simultaneously considered both for the aforementioned task. The former heavily depends on the 3D information of the object as an input, which is often expensive and time-consuming. For example, 3DStyleNet [41] requires training on a large dataset of 3D shapes, while Neural Renderer [20] transfers the style of an image onto a 3D mesh. In addition, 3D scene stylization [12, 27] generates stylized novel view synthesis from multiple RGB images given an arbitrary style. The latter approach can easily implement 3D object reconstruction using single-view images. Classical methods predict the reconstructed information of a 3D object from the single-view...
images [18, 10, 19, 39, 11]. Specifically, category-specific mesh reconstruction (CMR) [18] learns to recover the 3D mesh shape, texture, and camera pose of an object from the single images of one object category. Following CMR, unsupervised mesh reconstruction (UMR) [22] uses self-supervision with semantic part consistency to predict a 3D object. As UMR does not require 3D information, a union between the two approaches becomes a viable strategy to tackle the single-view 3D style transfer task.

In this paper, we primarily focus on the novel 3D bird generation as it is a popular example [18, 2, 38, 22] in single-view 3D reconstruction, especially UMR [22]. Specifically, given a pair of single-view source and target bird images within and between species, we explore plausible 3D bird creation by transferring the geometric and textured styles of the target bird to the source, as shown in Fig. 2. First, we employ the reconstruction network in UMR [22] as an encoder to output the shape features, UV textures, and camera poses of the target and source bird images, respectively. Second, we propose a shape transfer generator for implementing the geometric transfer which consists of a dual residual gated network (DRGNet), a scale factor, and a multi-layer perceptron (MLP). DRGNet is designed with a shared coordinate gate unit to select both the target and source shape features, separately cascading single-layer networks and incorporating residual connections into their networks in Fig. 3. The scale factor controls the degree of the transformation from the source to the target. MLP generates the spatial coordinates necessary for building a 3D mesh. Third, we introduce a semantic UV texture transfer that exploits switched gates to respectively control whether the part textural transfer or not using AdaIN [13], LST [21] and EFDM [44]. Finally, a novel 3D bird is constructed by using a differentiable renderer [24] with the pose of the source, the mesh, and the semantic stylized UV map. Overall, our contributions are summarized as follows:

- We propose a shape transfer generator consisting of a dual residual gated network, a scale factor, and a multi-layer perceptron for the geometric style transfer.
- We present a semantic UV transfer method to utilize switched gates to respectively control whether the part textural transfer or not by using AdaIN [13], LST [21], and EFDM [44], and employing semantic UV segmentation for the texture style transfer.
- To the best of our knowledge, we are the first to address the single-view 3D style transfer task by combining the shape and texture transfer to automatically create novel 3D birds from images in Fig. 1.

2. Related Work

Here, we mainly review some related tasks about the style transfer from single-view images to 3D objects, including single-view 3D reconstruction & deformation, and 2D & 3D style transfer.

**Single-view 3D reconstruction & deformation.** Single-view 3D reconstruction aims to generate a 3D model from a single 2D image. Early works [36, 28] trained the model using the image and ground truth mesh pairs, which are usually limited and expensive to obtain. Based on the differentiable renderers [20, 24, 29], weakly supervised methods become possible [39, 46, 33], but typically require multi-view images or known camera parameters. CMR [18] avoids these requirements by constructing inter-class template mesh, but still needs to be initialized by the landmark. UMR [22] avoids the need for landmarks by maintaining semantic correspondence between objects of the same class. Using UMR, our model gains the capability to perform the 3D reconstruction.

The goal of 3D deformation is to deform the shape of the target object to the source. Some early works tried to set controllable points [32, 31] or cage [4] to deform the shape of the object with manual control. To avoid human intervention, some works tried to find the correspondence between source and target through building deformable cage [37], encoding deformation process [34] and detecting 3D keypoints [17]. We will compare our shape transfer module with some of these works using the 3D reconstructed shape.

**2D & 3D style transfer.** 2D style transfer has been extensively explored with many proposed methods such as AdaIN [13], LST [21], Adaattention [25], InSt [40], and EFDM [45]. These methods can be easily extended to UV texture maps as they are also 2D images. Following this strategy, we explore a semantic UV texture transfer method for the diversity of the stylized UV texture.

For 3D style transfer, early works [20, 23, 26] reconstructed 3D meshes from single-view or multi-view images and used differential rendering methods to transfer a style image into the 3D mesh. 3DStyleNet [41] focuses on shape transformation and still uses the texture of the source in the transferred mesh. Other methods perform 2D-like style transformation for point clouds [12, 27] or implicit fields [14, 5, 42] without considering the style of the 3D shape.

In contrast, we study both 3D shape and UV texture transfer from the target image to a source image for creating a 3D object. The work most similar to ours is 3D portrait stylization [8], which produces a stylized 3D face composed of geometric and texture style transfer. The geometric transfer reconstructs a 3D geometry from the source face image, utilizing facial landmarks translation between the source and target face images to guide the deformation of the dense 3D face geometry. The texture transfer performs style transfer on the target texture by using a differentiable renderer in a multi-view manner. In contrast to this work, we design a shape transfer network to directly generate a novel 3D mesh and introduce a semantic UV texture.
transfer method to obtain a new UV texture. More importantly, our work can handle the single-view 3D style transfer task for novel 3D object generation.

3. Single-view 3D Style Transfer

In this section, we aim to extract 2D information from two bird images and fuse these features to construct a novel 3D bird. To achieve this, we propose a single-view 3D style transfer framework that automatically generates a plausible 3D bird, as illustrated in Fig. 2. In our method, we utilize the pre-trained ResNet-18 from the UMR paper [22] as a feature extractor to extract shape features. We also adopt the texture predictor and camera predictor from UMR to predict the corresponding textures and camera parameters. In the following sections, we will refer to these modules as the encoder in our network. Given a source bird image \( S \in \mathbb{R}^{256 \times 256 \times 3} \) and a target bird image \( T \in \mathbb{R}^{256 \times 256 \times 3} \), the encoder outputs the camera pose \( C_S \in \mathbb{R}^7 \), \( C_T \in \mathbb{R}^7 \), shape features \( F_S \in \mathbb{R}^{512} \), \( F_T \in \mathbb{R}^{512} \) and the UV texture flows \( U_S \in \mathbb{R}^{128 \times 256 \times 3} \), \( U_T \in \mathbb{R}^{128 \times 256 \times 3} \), respectively. Concurrently, we construct a semantic UV segmentation \( U_{seg} \in \{1, 2, 3, 4, 5\}^{128 \times 256} \) from UMR [22] using SCOPS [16]. Following the encoder and the UV segmentation, our framework mainly consists of a shape transfer generator from \( F_S \) to \( F_T \) and a semantic texture transfer from \( U_S \) to \( U_T \).

3.1. Shape Transfer from Images to 3D Mesh

Here, we aim to generate a novel 3D shape from source and target bird images \( S \) and \( T \). Using the previously described encoder, we obtain their shape features \( F_S \) and \( F_T \). Specifically, we explore a shape transfer to generate a 3D mesh by progressively adjusting and filtering the shape features \( F_S \) and \( F_T \). It includes a dual residual gated network (DRGNet), a scale factor, and a multi-layer perceptron.

\[
g_l = \text{Sigmoid} \left( W_l \cdot [F_S^l, F_T^l] \right),
F_S^{l+1} = F_S^l + \text{ReLU} \left( \text{BatchNorm} \left( W_S^l \cdot [F_S^l \circ g_l] \right) \right),
F_T^{l+1} = F_T^l + \text{ReLU} \left( \text{BatchNorm} \left( W_T^l \cdot [F_T^l \circ g_l] \right) \right),
\]
where \( W_l \in \mathbb{R}^{512 \times 1024} \), \( W_S^l \in \mathbb{R}^{512 \times 512} \) and \( W_T^l \in \mathbb{R}^{512 \times 512} \) are the weights of the \( l \)-th unit, and Sigmoid, Tanh, and ReLU are the activation functions, respectively. To enhance the representation of specific individual traits,
we stack the gated unit in Eq. (1) \( L \) times in our DRGNet for refining the features progressively, and it outputs the features \( F_S^L \in \mathbb{R}^{512} \) and \( F_T^L \in \mathbb{R}^{512} \). In this paper, the stacked number is set to \( L = 8 \).

**Scale Factor** \( \alpha \) is to control the proportion of the features \( F_S^L \) and \( F_T^L \) during testing, while it is set to one during training. Using the scale factor, the features are computed by

\[
[F_S, F_T] = [(1 - \alpha) \cdot F_S^L, (1 + \alpha) \cdot F_T^L],
\]

where \(-1 \leq \alpha \leq 1\), \( F_S \in \mathbb{R}^{512} \) and \( F_T \in \mathbb{R}^{512} \).

**Multi-layer Perceptron** is employed to fuse the features \( F_S \) and \( F_T \). It is a simple two-layer neural network with the ReLU activation function, which is defined by

\[
O = W_o \cdot \text{ReLU} \left( W_f^2 \cdot \text{ReLU} \left( W_f^1 \cdot [F_S, F_T] \right) \right),
\]

where \( W_f^1 \in \mathbb{R}^{512 \times 1024} \), \( W_f^2 \in \mathbb{R}^{512 \times 512} \), and \( W_o \in \mathbb{R}^{5 \times 372 \times 512} \) are the network parameters, and \( O \in \mathbb{R}^{5 \times 372} \) is the output of the multi-layer perceptron.

Since most birds are symmetric, we only predict the left half of the object in the output, with a total of 372 vertices. We then obtain an additional 270 coordinates of the symmetrical vertices, as the remaining 102 vertices are on the symmetrical plane. Finally, we use all of the 642 three-dimensional coordinates to build the 3D mesh by employing a differentiable renderer, i.e., soft rasterizer [24].

### 3.2. Semantic UV Texture Transfer

Following the shape transfer, our goal is to generate a texture using the extracted source and target UV texture flows \( U_S \) and \( U_T \) obtained from the encoder. However, birds often exhibit different textures or colors on different body parts, such as sparrows that have a white belly, a messy back pattern, and fixed head spots. Treating the overall texture as a single style during transfer may result in the transfer of local textures to incorrect semantic parts. To address this issue, we propose a semantic UV texture transfer method that performs semantic segmentation of the texture before transferring style to the corresponding semantic parts. Our approach involves creating a semantic texture using a semantic UV mask, which ensures consistency of semantic distributions as shown in Fig. 4. It consists of a semantic UV mask, a VGG encoder, a semantic stylizer, and a decoder.

**Semantic UV Mask** \( U_{seg} \in \{1, 2, 3, 4, 5\}^{128 \times 256} \) is generated by SCOPS [16]. To match the size of the subsequent VGG feature map, \( U_{seg} \) is downsampled, but it is still referred to as \( U_{seg} \in \mathbb{R}^{64 \times 128} \) for convenience. It comprises five parts, including four semantic UV parts: red head \( U_1^{seg} \in \{0,1\}^{64 \times 128} \), green neck \( U_2^{seg} \in \{0,1\}^{64 \times 128} \), blue belly \( U_3^{seg} \in \{0,1\}^{64 \times 128} \), and yellow back \( U_4^{seg} \in \{0,1\}^{64 \times 128} \), and one non-semantic part \( U_5^{seg} \in \{0,1\}^{64 \times 128} \) to preserve the source UV texture. Each element \( u^{i}_{jk} \) in \( U_{seg} \) is defined as follows:

\[
u^{i}_{jk} = \begin{cases} 1, & \text{if } u^{i}_{jk} = i, \\ 0, & \text{otherwise}, \end{cases}
\]

where \( 1 \leq i \leq 5 \), \( u^{i}_{jk} \) is the element of the UV mask \( U_{seg} \).

**VGG Encoder** [30] is used to extract the feature of the UV texture, which has 512 channel maps of size \( 64 \times 128 \). Given the source and target UV texture flows \( U_S \) and \( U_T \), the encoder outputs the features \( V_S \in \mathbb{R}^{512 \times 64 \times 128} \) and \( V_T \in \mathbb{R}^{512 \times 64 \times 128} \), respectively.

**Semantic Stylizer** is introduced to incorporate the semantic UV mask, represented as \( U_{seg} \), into existing UV texture transformation methods for enhanced stylization. Specifically, we focus on the instance normalization approach, which aligns the channel-wise mean and variance of the masked source feature \( V_S \) with those of the corresponding masked target feature \( V_T \), without requiring additional parameters. We extend this approach to various methods such as AdaIN [13], linear style transfer (LST) [21] and EFDM [44]. To handle the five semantic parts, semantic AdaIN (SAdaIN) is computed by uniting these corresponding AdaIN feature indexed by \( U_{seg} \), that is:

\[
\text{SA}(V_S, V_T, U_{seg}) = \sum_{i=1}^{5} \text{SA}^i(V_S^i, V_T^i, U_{seg}^i);
\]

\[
\text{SA}^i = \begin{cases} \sigma(V_T^i) \frac{V_S^i \odot \mu(V_S^i)}{\sigma(V_S^i)} + \mu(V_T^i), & \text{if } i \neq 5, \\ V_S^5, & \text{if } i = 5, \end{cases}
\]

where \( V_S^i = V_S \odot \overline{U}_{seg}^i \), \( V_T^i = V_T \odot \overline{U}_{seg}^i \). \( \overline{U}_{seg}^i \) is a binary matrix to repeat the index matrix \( U_{seg}^i \) 512 times and reshape it to \( 512 \times 64 \times 128 \). \( \odot \) is an element-wise multiplication, \( \odot, \sigma \) and \( \mu \) are the subtraction, variance, and mean computations only depend on the index binary matrix \( \overline{U}_{seg}^i \). Here, \( V_S^5 = V_S^5 \) that means SAdaIN holds the...
source feature for the non-semantic part. Clearly, these local statistics are respectively computed across spatially masked locations, are merged into a new feature SA.

Similar to SAdaIN, semantic LST (SLST) is defined as
\[
SL^i = \begin{cases} 
\text{cov}(V^i_S) \text{cov}(V^i_T) V^i_S, & \text{if } i \neq 5, \\
V^i_S, & \text{if } i = 5, 
\end{cases}
\]

where \(\text{cov}(V^i_S) = V^i_S (V^i_S)^T\) is the centered covariance of \(V^i_S\).

In addition, we further extend to a histogram-based style transfer method, \textit{i.e.}, Exact Histogram Matching (EHM) [6]. Similarly, semantic EFDM (SEFDM) is defined as
\[
SE^i = \sum_{i=1}^{5} \text{SL}^i (V^i_S, V^i_T, U^i_{seg}).
\]

To increase the diversity of the texture, we also design a switch gate (SG) to stochastically exchange the part source feature \(V^i_S\) and the part target feature \(V^i_T\) in both the Eqs. (5) and (7).

Decoder is identical to the decoder of AdaIN [13], LST [21] and EFDM [44] for our SAdaIN, SLST and SEFDM, respectively.

3.3. Training Loss

Our training loss is divided into two parts: shape transfer and texture transfer. The former mainly includes single-view 3D shape reconstruction with mask and perceptual loss and shape transfer with 3D key points loss. The latter part consists of traditional style transfer with style and content loss. These two parts are trained separately. The shared encoder and VGG-19 are pre-trained using UMR [22] and [30], respectively. Given the source and target bird images \(S\) and \(T\), the shared encoder outputs the shape features \(F_S\), \(F_T\), and the UV maps \(U_S, U_T\) respectively.

Shape Transfer. During source shape reconstruction, the model uses only the source shape feature \(F_S\). However, our model requires both shape features \((F_S, F_T)\). Thus, we set both shape features to \(F_S\) during reconstruction, and it outputs the result \(O_S\). Similarly, they are set to \(F_T\) for the target reconstruction and it outputs \(O_T\). Following UMR [22], the mask and perceptual losses are introduced as follows.

Mask Loss aims to compute the negative IoU [19] between the ground truth instance mask \(M\) and the predictive mask \(\overline{M}\), which is produced by using [24] to render \(O\) based on the camera pose predicted by the shared encoder [22]. Considering the source and target masks, it can be defined as:
\[
\mathcal{L}_{\text{mask}} = \mathcal{L}_m(M_S, \overline{M}_S) + \mathcal{L}_m(M_T, \overline{M}_T),
\]

where \(\mathcal{L}_m(M, \overline{M}) = 1 - \frac{\|M \odot \overline{M}\|_1}{\|M\|_1}\).

Perceptual Loss calculates the perceptual distance [43] between the input image \(I\) and the predictive image \(T\) produced by its UV map using the method [22]. Considering the source and target images \(S\) and \(T\), it is described as:
\[
\mathcal{L}_{\text{per}} = \mathcal{L}_p(S, \overline{S}) + \mathcal{L}_p(T, \overline{T}),
\]

where \(\mathcal{L}_p(I, T) = \|\text{VGG}(I) - \text{VGG}(T)\|_2\). This loss can improve the visual quality of texture by capturing its details.

3D Keypoints Loss is, more importantly, proposed to implement the shape transfer between source and target. The 3D key points are obtained by detecting the 2D key points [7] from an image, determining their corresponding vertices on the reconstructed 3D shape using the predicted UV texture, and projecting them onto the symmetry plane. Since the symmetry plane is identical for all output 3D models, we can calculate the distance between key points based on their projections. This loss is defined as the \(\ell_2\) distances of \(\{p^i_s, p^i_t, p^i_f\}_{i=1}^N\), which are the 3D keypoints of the transferred, source, and target shapes. It is calculated by
\[
\mathcal{L}_{\text{key}} = \frac{1}{2N} \sum_{i=1}^{N} (|p^i_s - p^i_t|_2 + \lambda |p^i_s - p^i_f|_2),
\]

where \(N\) is the number of 3D keypoints, \(\lambda\) is a balance parameter, and in this paper we set \(N = 15\) and \(\lambda = 1\). In summary, the overall objective for shape transformation is
\[
\mathcal{L}_{\text{shape}} = \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{per}} + \mathcal{L}_{\text{key}}.
\]

Texture Transfer. We extend AdaIN [13], LST [21], and EFDM [44] into SAdaIN, SLST, and SEFDM respectively on the UV maps for the texture transfer. We utilize their style loss \(\mathcal{L}_{\text{style}}\) and content loss \(\mathcal{L}_{\text{content}}\) as our texture transfer loss \(\mathcal{L}_{\text{texture}} = \mathcal{L}_{\text{style}} + \mathcal{L}_{\text{content}}\) on the UV textures \(U_S\) and \(U_T\).

4. Experiments

In this section, we conduct extensive experiments to evaluate the transfer ability of birds across populations of the same species and between species. More results are provided in supplementary materials.

Datasets. Our method is validated on the CUB-200-2011 dataset [35]. The dataset provides the bounding boxes and instance masks for each bird, and we obtain semantic segmentation using SCOPS [16]. Based on the provided bounding boxes, each image is cropped and resized to 256x256 to form one training sample and randomly flipped with a probability of 0.5.

Training. Our training schedule is divided into two steps due to the inclusion of texture transfer and shape transfer modules in our model. We first train the shape transfer network using the source and target images. Then, we use the
Figure 5: Visual comparisons on shapes using 3D deformation methods (e.g., NC [37], DSN [34], KPD [17] and NT [15]). Both the source and target birds in the row 1-3 are from the same species, while they become from different species in the row 4-6. "Recon_s" and "Recon_t" represent the reconstruction of the source and the target respectively. Our method effectively achieves novel shape creation from these reconstructions and maintains reasonability.

UV textures produced by the encoder for the source and target to train the texture transfer network. In our experiment, these two steps are trained for 100k/80k iterations with an 8/8 batch size. We use Adam optimizer with a learning rate of 1e-4/initial 1e-4 and decay by 0.9995 per iteration. On a single GTX 2080ti GPU, training takes about 3/30 hours.

4.1. Main Results

To show the transfer capacity of our method for 3D bird creation, we compare with four shape deformation methods (i.e., NC [37], DSN [34], KPD [17] and NT [15]) and three texture style transformation methods (i.e., AdaIN [13], LST [21] and EFDM [44]).

Visual comparisons. We validate the effectiveness of our model from two aspects: shape and texture transfer.

Shape Transfer. Our work is not directly comparable to the 3D style transfer methods, even those involving 3D portrait stylization [8] as they focus only on the face. For potential comparisons, we initially use the UMR to reconstruct the 3D models and then compare them with the shape deformation methods (e.g., NC, DSN, KPD, and NT). Fig. 5 showcases the results of the shape transformation. We observe that our method achieves a reasonable shape transformation, better matching the characteristics of the source and target. It demonstrates that when the shape difference between the source and target objects from different species is large, comparison methods can produce unreasonable distortions. This is likely due to the semantic part being difficult to align. Conversely, our method learns to reconstruct the 3D model to prevent shape distortion and transfer their shapes for novel shape generation simultaneously.

Texture Transfer. Fig. 6 shows the results of the style transfer algorithms (e.g., AdaIN, LST, and EFDM), after utilizing the modules of semantic transfer. Fig. 7 shows the result after adding different switch gates. Comparing Fig. 6 columns (c) and (d), we see that semantic transfer improves the effect of style transfer on each semantic part for all algorithms because the semantic mask prevents influence between different semantic parts. Furthermore, in Fig. 7 switch gate further enhances the diversity of results based on the semantic mask, making our transfer more consistent with natural evolutionary laws.

Comparison with real hybrid birds. To further validate the effectiveness of our model, we simulate the morpholog-
Figure 6: Visual comparisons on textures using style transfer methods (e.g., AdaIN [13], LST [21] and EFDM [44]) associate with our method (e.g., semantic UV mask (+M)). Both the source and target birds in the left picture are from the same species, while they become from the different species in the right picture. We can see that a semantic UV mask can increase the diversity of the texture compared to the original methods. More results are shown in Fig. 7.

Figure 7: Visual comparisons with different switch gates (+SG), the subscript are the switch signals for each semantic part (i.e. head, neck, belly, and back). It is obvious that the switch gate further increases the diversity of the transformed texture.

Table 1: Quantitative evaluations of shape transfer methods.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask IoU†</td>
<td>0.6670</td>
<td>0.6937</td>
<td>0.5699</td>
<td>0.5209</td>
<td>0.7316</td>
</tr>
</tbody>
</table>

Table 1: Quantitative evaluations of shape transfer methods.

Figure 8: Visual comparisons to the real hybrid birds.

User Study. We conducted a user study to better assess our model’s performance compared to existing models. This study consisted of three main components: shape transformation, texture transformation, and realism judgment, which included 25 questions in total. As a result, we collected 102 questionnaire responses with a total of 2550 votes. Table 2 shows the number of votes received for each option. In the shape transformation comparison, 52.5% of users prefer our results compared to 27.8% for NC, 11.8% for DSN, 3.5% for KPD, and 4.3% for NT. In the texture transformation comparison, 76.9% of users prefer the results of SAdaIN over AdaIN, 71.2% prefer the results of SLST results over LST, and 64.1% prefer the results of SEFDM results over EFDM. In the realism judgment, 73.7% of users think that our result is more realistic.

4.2. Ablation Study

Since the modules of semantic transfer and switch gate have been compared in the previous subsections. Here we conduct ablation experiments on DRGNet, 3D keypoints loss $L_{key}$ and scale factor $\alpha$, respectively.

Effect of DRGNet. We evaluate the importance of DRGNet by replacing it with MLP. From Fig. 10, (e) ours using DRGNet can restore more edge details than (c) w/o DRGNet (that is MLP) because it helps in the cooperation of shape features. Moreover, Fig. 11 also shows that...
Table 2: Quantitative evaluations of user study.

<table>
<thead>
<tr>
<th>Method</th>
<th>Texture</th>
<th>Shape</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake</td>
<td>Real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Votes↑</td>
<td>118</td>
<td>392</td>
<td>147</td>
</tr>
</tbody>
</table>

Figure 9: Transfer degree of the shape controlled by the scale factor $\alpha$ from $-1$ to $1$.

Figure 10: Shape ablation study of DRGNet and 3D keypoints loss $L_{\text{key}}$ using our SLST+SG as texture transformation method.

Figure 11: Masked IoU using DRGNet and MLP.

DRGNet has a higher masked IoU than the MLP.

Effect of 3D keypoints loss $L_{\text{key}}$. We evaluate the effectiveness of $L_{\text{key}}$ by replacing it with masked IoU loss, where the masks of the source and the target are used to constrain the result together. Then the results using the mask IoU loss instead of $L_{\text{key}}$ are shown in Fig. 10 (d) w/o $L_{\text{key}}$, indicating significantly poorer outputs. The reason is that when the shape difference between the source and target object is large, their masks are more likely to have shape conflicts, resulting in an averaged result.

Scale factor $\alpha$. We use $\alpha$ to adjust the ratio between source and target features from $-1$ to $1$. Fig. 9 demonstrates that the scale parameter $\alpha$ effectively controls the presentation of the source and target characteristics in the results. When $\alpha = -1$ or $1$, the results are exactly the source or target 3D reconstruction. When $\alpha = 0$, the results are to combine the half source with the half target. In addition, we also conducted experiments on DRGNet with different layers, which are included in the supplementary materials.

Limitation. In this section, we discuss the limitations of our model. As we intend to perform style transformation on 2D images at a 3D level, the model needs to understand the 3D structure of the input image correctly. The model will produce problematic results when it incorrectly perceives the three-dimensional structure of the input image. According to Fig. 12, the model improperly treats the sky as part of the bird, leading to a blue appearance. In the second row, the incorrect reconstruction of the complex target directly fails our style transfer algorithm.

5. Conclusion

In this paper, we proposed a style transfer method to automatically create novel 3D birds from single-view images.
We obtain the transformed shape features by filtering and coordinating the image features progressively and then utilizing a semantic transfer module to enhance the expressiveness of semantic texture style and the diversity of evolution. Experimental results on the CUB dataset demonstrate that our method achieves state-of-the-art performance. Note that this is the first meaningful try to extend the style transfer into the novel 3D bird creations from the images.

6. Acknowledgments

This work was partially supported by the National Science Fund of China (Grant Nos. 62072242, 62206134). Note that the PCA Lab is associated with, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, China.

References

[25] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adattn: Revisit attention mechanism in arbitrary neural style transfer. In ICCV, pages 6649–6658, 2021. 2


