Industrial Style Transfer with Large-scale Geometric Warping and Content Preservation

Jinchao Yang[1*], Fei Guo[1*], Shuo Chen[2], Jun Li[1†], Jian Yang[1]

[1]PCA Lab, Nanjing University of Science and Technology   [2]RIKEN

{yangjinchao,feiguo,junli,csjyang}@njust.edu.cn   shuo.chen.ya@riken.jp

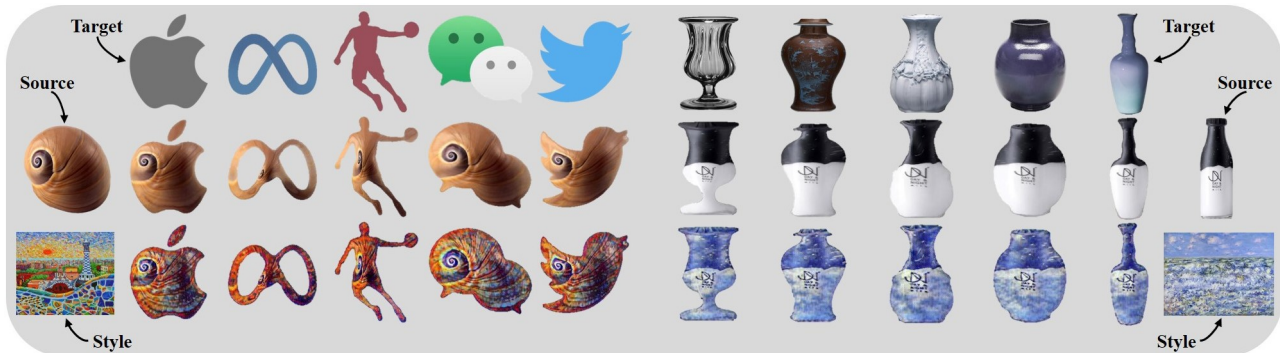*Contributes equally   †Corresponding author&project lead

Figure 1. We propose an Industrial Style Transfer method for visual product design. Our method creates new product appearances (*e.g.*, logos and Day&Night bottles) by transferring both the shape of one product (target) and art style reference to another (source).

Abstract

We propose a novel style transfer method to quickly create a new visual product with a nice appearance for industrial designers reference. Given a source product, a target product, and an art style image, our method produces a neural warping field that warps the source shape to imitate the geometric style of the target and a neural texture transformation network that transfers the artistic style to the warped source product. Our model, Industrial Style Transfer (InST), consists of large-scale geometric warping (LGW) and interest-consistency texture transfer (ICTT). L-GW aims to explore an unsupervised transformation between the shape masks of the source and target products for fitting large-scale shape warping. Furthermore, we introduce a mask smoothness regularization term to prevent the abrupt changes of the details of the source product. ICTT introduces an interest regularization term to maintain important contents of the warped product when it is stylized by using the art style image. Extensive experimental results demonstrate that InST achieves state-of-the-art performance on multiple visual product design tasks, e.g., companies' snail logos and classical bottles (please see Fig. 1). To the best of our knowledge, we are the first to extend the neural style transfer method to create industrial product appearances. Code is available at https://jcyang98.github.io/InST/home.html

1. Introduction

Visual Product Design (VPD) has been recognized as a central role in the industrial product design field as consumers' choices heavily depend on the visual appearance of a new product in the marketplace [12]. VPD often designs a novel product [11] by following different appearance roles (*e.g.*, aesthetic, functional, and symbolic). For example, designers usually produce the beautiful appearance of flying cars by referring to airplanes and cars to fuse their flying and driving functions and attractive aesthetic. However, it is difficult to quickly create high-quality product appearances due to the human intelligence in the VPD process, which is heavily dependent on designers' creative ability. Fortunately, neural style transfer (NST) [16, 21, 28, 38], aiming at transferring artistic and geometric styles of one or two reference images to a content image, has a strong opportunity to assist the designers because the art style transformation is suitable for the aesthetic value, and some geometric shape transformations can gain the functional and symbolic values, such as Beijing National Stadium (bird's nest and building). Therefore, we seek for a style transfer formulation to automatically generate many visual appearance candidates of new products for industrial designers' reference.

However, most modern NST methods [14,25,32,59,60], including geometric NST [28, 38], are difficultly or impossibly extended into directly design visual product appearances due to the following two challenges. *One is the large-*
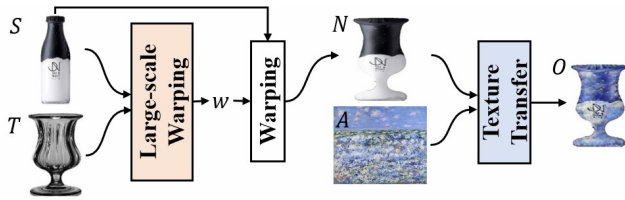
Figure 2. The pipeline of our industrial style transfer. Our method creates a new product N by warping source S to target T, and generates a final product appearance O by transferring the artistic style of reference image A to the new product N.

scale geometric shape between diverse objects (or products) since designing new products is often to fuse two objects with very different geometries, such as flying cars (airplanes and cars) and butterfly doors (butterfly's wings and car doors). *Another is that NST usually makes the content worse in the stylization process*, *e.g.*, AdaIN [21], and WCT [33], resulting in that product designers cannot refer to both rich content and novel geometry for generating creative inspirations.

To address these challenges, we develop an industrial style transfer (InST) method to create new product appearances, shown in Fig. 2. Given a source product (or object), a target product, and an art reference image, InST aims to transfer both the industrially geometric shape of the target product and the art style of the reference image to the source product. In contrast to the existing NST methods, InST consists of large-scale geometric warping (LGW) and interest-consistency texture transfer (ICTT). Unlike small-scale geometric NST [28, 38], LGW designs a neural warping field between the shape masks of the source and target products using a shape-consistency loss. This is different from a warping field between their textural pixels as it results in a worse optimization, that is, a failure deformation. In addition, we explore a mask smoothness regularization term to prevent the abrupt changes of the details of the source product. With the help of the masks, LGW works well in large-scale warping between two products even though they are semantically irrelevant.

ICTT aims to keep interesting contents of the new product when it is stylized by using the art reference image. Inspired by the SuperPoint network [15], we present an interest regularization (IR) term based on both interest points and descriptors to constrain the art stylization to minimize perceptual differences between the new product and its stylized product. Unlike the most relevant work, ArtFlow [3], we design the interesting perceptual constraint to prevent worse contents, and our IR can further improve the performance of ArtFlow. Overall, the contributions of this work are summarized as follows:

- For large-scale geometric differences in the visual product design process, we explore a large-scale geometric warping module based on mask to transfer ge-

ometric shape style from one object product to another, even though irrelevant semantics.

- For product content maintenance in the stylization process, we introduce an interest-consistency texture transfer with interesting regularization using interest point and descriptor extracted by the SuperPoint network to preserve content details.

- Combining LGW with ICTT, we propose an industrial style transfer framework to fast generate visual appearances of new products, *e.g.*, companies' logos, flying cars, and porcelain fashions. To the best of our knowledge, this work could open up a new field of style transfer, designing industrial product appearances.

2. Related Work

In this section, we mainly review visual product design, texture style transfer, and geometric style transfer since we extend the style transfer technology to a new application, product appearance design tasks.

2.1. Visual Product Design

Due to the important influence of product appearance on consumers perceptions [5, 6], visual product design (VPD) can be considered as a communication process between designers (companies) and consumers [12]. In this process, the designer aims to communicate a specific message through the product appearance by changing the geometry, art style *etc.*, and the consumers provide their responses to the designers for the product improvements when seeing the product appearance [41]. Usually, there are four popular types of product appearance for consumers: aesthetic impression, functional effect, symbolic association and ergonomic information [11, 12, 43].

However, this is a manual labour procedure with an expensive cost of communication and product design because it needs many feedback loops and designers take a lot of time to improve the product design in each loop [12]. This expensive cost drives us to explore a fast design method. More importantly, since high-quality product appearance is dependent on the designers creative ability, it encourages us to generate many product innovations to inspire the designer. Therefore, we develop a novel style transfer method to create many visual product appearance candidates to assist or inspire designers.

2.2. Texture Style Transfer.

As a hot topic, texture style transfer has developed for a long time. The initial works [16, 17, 34, 45] pay attention to iterative optimization. Later, numerous works [9, 26, 52, 62] based on feed-forward network improve both quality and quantity, such as visual effect and computation time. Although improving texture style transfer greatly,

these methods transfer only one style via a trained model. A lot of works including AdaIN [21], WCT [33], Avatar-Net [48], LinearWCT [32], SANet [42], MST [63] and recent [20, 31, 35, 37, 54, 55, 57, 58], are extended to an arbitrary style transfer. However, these methods are limited to preserving the details of the content image. The problem of the worse content in style transfer has attracted the attention of many scholars. A structure-preserving algorithm [10] is introduced to preserve the structure of the content image. ArtFlow [3] preserves more details from content image via reversible neural flows. However, their visual qualities are still to be improved. We propose an interest regularization computed by a SuperPoint network [15], which takes the content image as input and outputs the corresponding interest points and descriptors to make the content better.

2.3. Geometric Style Transfer

Traditional geometric matching approaches involve detecting and matching hand-crafted interest points, such as SIFT [40], Shape Context Matching [4] or HOG [13]. While these approaches work well for instance-level matching, it is sensitive to appearance variation and noise disturbance. Later, the convolutional neural network has been popular in geometric matching due to its ability to extract powerful and robust features. The current best methods follow the network paradigm proposed by [46] which consists of feature extraction, matching layer and regression network, and make various improvements [18, 27, 38, 46, 47] based on it. All the above methods act on two RGB images and attempt to estimate a warp field to directly match them. Although performing well between semantically similar images, they fail to deal with objects of different categories with large-scale warping. In the absence of semantic relevance, computing the correlation between two RGB images is unreasonable and defining a match metric is also difficult. DST [28] achieves warping by matching NBB keypoints [2] and estimating thin-plate spline (TPS) [7] transformation. It is also limited to class-level warping because NBB can only extract key points between similar objects. Some methods are restricted to specialized semantic class such as faces [61], caricature [49] or text [60]. Compared to the above geometric matching method, we achieve large-scale warping even between arbitrary objects of different categories. Overall, different from the above methods, we aim to broaden the style transfer application for product design tasks, and our method obtains amazing industrial product appearances to inspire the designers.

3. Industrial Style Transfer

In this section, we develop an industrial style transfer (InST) framework to create new visual product appearances in Fig. 2, consisting of two modules: large-scale geometric warping (LGW) in subsection 3.1 and interest-consistency

texture transfer (ICTT) in subsection 3.2. We denote a source product (or object) by S, a target product T, an art reference image by A, a new warping product using LGW by N, and a final output by O.

3.1. Large-scale Geometric Warping

The goal of LGW is to warp a source product S to match the geometric shape of a target product T for a new product generation even if large-scale shape differences and irrelevant semantics. To achieve this goal, we design a neural warping filed between their shape masks inspired by an optical flow method, recurrent all-pairs field transforms (RAFT) [51]. Specially, Fig. 3 shows our LGW module including a mask RAFT and an unsupervised warping loss.

3.1.1 Mask RAFT Network

The mask RAFT network can be distilled down to five stages: (1) mask extraction, (2) feature extraction, (3) position embedding, (4) correlation computation, (5) recurrent updates. More details are described as follows.

Mask Extraction. We employ an object segmentation network, denoted as $\mathcal{F}_m : \mathbb{R}^{H \times W \times 3} \rightarrow \{0,1\}^{H \times W} \xrightarrow{\text{repeat it 3 times}} \{0,1\}^{H \times W \times 3}$, to extract the masks of the products. Given the products S and T as inputs, their masks are $M_s = \mathcal{F}_m(S)$ and $M_t = \mathcal{F}_m(T)$, respectively. Here, we use a fixed Resnet50+FPN+PointRend (point-based rendering) network, which has been pre-trained in [30].

Feature Extraction. Mask features are extracted from the input masks M_s and M_t using a convolutional encoder network, denoted as $\mathcal{F}_f : \{0,1\}^{H \times W \times 3} \rightarrow \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times D}$, where D is set to 256. In order to compute the correlation between M_s and M_t, the network is similar to the feature encoder network of RAFT [51], which consists of 6 residual blocks, 2 at 1/2, 1/4, and 1/8 resolutions, respectively. Then, we have the mask multi-scale features, $F_s = \mathcal{F}_f(M_s)$, and $F_t = \mathcal{F}_f(M_t)$.

Position Embedding. Due to the lack of color information, there have too many similar or same features between source and target masks, resulting in weak correlation computation and deformation. To avoid such a situation, adjacent position information can improve the deformation field because it updates changes of every pixel (position) of the object product. Thus, we consider position embedding P [53] of the feature maps F_s and F_t by using the popular residual operation, and define the new position+feature as:

$$\begin{cases} \hat{F}_s = F_s + P \\ \hat{F}_t = F_t + P \end{cases}, \qquad (1)$$

where $P(i,j,k) = \begin{cases} \sin\left(\frac{j*\frac{W}{8}+i}{10000^{\frac{k}{D}}}\right), & k \bmod 2 = 0 \\ \cos\left(\frac{j*\frac{W}{8}+i}{10000^{\frac{k-1}{D}}}\right), & k \bmod 2 = 1 \end{cases}$,
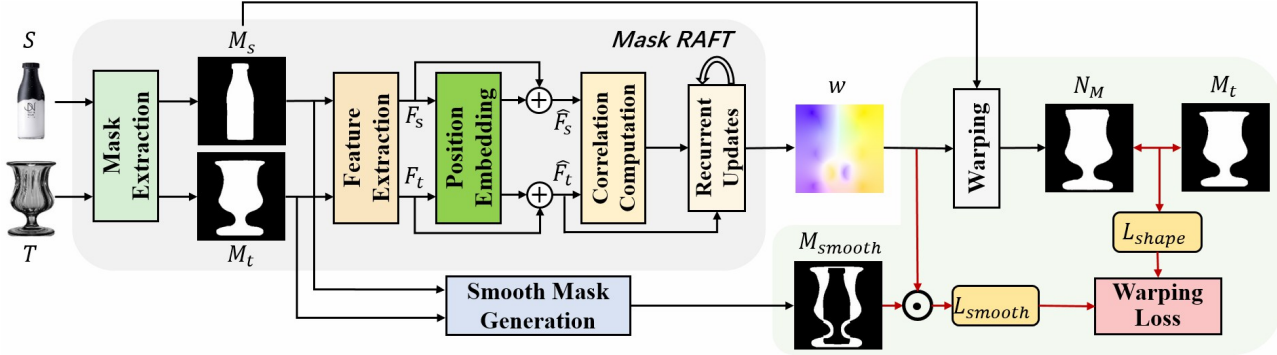$0 \leq i \leq \frac{H}{8}, 0 \leq j \leq \frac{W}{8}$, and $0 \leq k \leq D = 256$.

Figure 3. Our proposed large-scale geometric warping module including a mask RAFT network and an unsupervised warping loss.
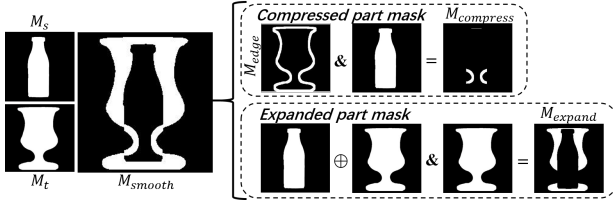

Figure 4. Given two shapes, we design different smoothness masks for the two parts of compression (upper right) and expansion (lower right). The smoothness mask in the middle is what we use in the smoothness regularization.

Correlation Computation and Recurrent Updates. Here, we follow the computing visual similarity and iterative updates of RAFT [51] to compute the multi-scale correlations and recurrently update the warping field. In this paper, these two steps are denoted as $\mathcal{F}_{cr}$: $(\mathbb{R}^{H/8 \times W/8 \times D}, \mathbb{R}^{H/8 \times W/8 \times D}) \to \mathbb{R}^{H \times W \times 2}$. Overall, our mask RAFT network is described as: $\omega = \{\omega_r\}_{r=1}^R$

$$= \mathcal{F}_{cr}\left(\mathcal{F}_f\left(\mathcal{F}_m(S)\right) + P, \ \mathcal{F}_f\left(\mathcal{F}_m(N)\right) + P\right), \quad (2)$$

where R is the number of iterations, and we set $R = 3$ in our implementation.

3.1.2 Unsupervised Warping Loss

The mask RAFT network is trained in an unsupervised setting by constructing a shape-consistency loss, and a smoothness regularization.

Shape-consistency loss. Based on the warping filed estimation ω in Eq. (2), we get the warped source masks $\{\omega_r(M_s)\}_{r=1}^R$ by the differential bilinear sampling mentioned in spatial transformer [23]. Given the target mask M_t, this ℓ_1 loss is defined as

$$L_{shape} = \sum_{r=1}^R \alpha_r \|\omega_r(M_s) - M_t\|_1, \quad (3)$$

where α_r is used to balance the deformation degree.

Smoothness regularization. To avoid the chaotic deformation, it needs to further restrict the sampling direction of the warping field for maximally holding the content detail of

the source object. Specially, we design a smoothness mask, shown in Fig. 4, and its generated formula is expressed as

$$M_{smooth} = M_{compress}|M_{expand}$$
$$= (M_{edge}\&M_s)|(M_s \oplus M_t\&M_t), \quad (4)$$

where $|$, $\&$ and $\oplus$ denote logical disjunction, conjunction, and XOR, M_{edge} represents edges of the target object product. M_{edge} is computed by convolution operation with all one kernel, $M_{edge} = Cov(M_t, ker)$, where $ker = [1]^{k \times k \times 3}$, k is a predefined kernel size, and we set $k = 9$. (More details are provided in supplementary materials.) Since $M_{smooth} \in \{0,1\}^{H \times W \times 3}$ has same mask maps in three channels, one channel is denoted by $\mathbf{M} \in \{0,1\}^{H \times W}$. Given the warp field estimations $\{\omega_r(M_s)\}_{r=1}^R$, the ℓ_2 regularization on $\mathbf{M}$ is defined as

$$L_{smooth} = \sum_{r=1}^R \beta_r L_{smooth}(\omega_r, \mathbf{M}), \quad (5)$$

where β_r denotes the degree of content retention of different warp fields, and $L_{smooth}(\omega_r, \mathbf{M}) =$

$$\frac{1}{\sum_{i,j} \mathbf{M}_{ij}} \sum_{i,j} \mathbf{M}_{ij} \left(\|\omega_r^{i+1,j} - \omega_r^{i,j}\|_2 + \|\omega_r^{i,j+1} - \omega_r^{i,j}\|_2 \right.$$
$$\left. + \|\omega_r^{i+1,j+1} - \omega_r^{i,j}\|_2 + \|\omega_r^{i+1,j-1} - \omega_r^{i,j}\|_2\right). \quad (6)$$

The above term is a first-order smoothness on the warp field ω by constraining the displacement of horizontal, vertical, and diagonal neighborhoods around coordinate (i,j). It drives the texture content of the source object to be close to its neighborhoods after the deformation. By combining L_{shape} with L_{smooth}, the warping loss is described as

$$L_{overall} = L_{shape} + \gamma L_{smooth}, \quad (7)$$

where $\gamma = 1$ controls the importance of each term.

3.2. Interest-Consistency Texture Transfer

After generating the new product N by LGW, the goal of ICTT is to create a stylized product appearance O with important content details of N by transferring the art style of
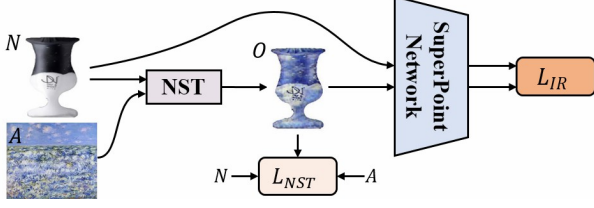
Figure 5. Interest-consistency texture transfer. It consists of a NST method for artistic style transformation and a SuperPoint network for content preservation by interest point constraints.

reference image A to N using neural style transfer (NST) methods. To achieve this goal, we introduce an interest regularization (IR) term in Fig. 5 to maintain similarity between interesting contents of O and N based on the SuperPoint network [15] as it can effectively compute interest point locations and their associated descriptors.

NST is often to train an image transformation network $\mathcal{F}$ by minimizing a NST loss, denoted as L_{NST}, including both content and texture style losses. In this work, we consider two popular algorithms, AdaIN [21] and LinearWCT [32], and one most relevant method, ArtFlow [3].

IR is to control perceptual differences between N and O by the SuperPoint network, denoted as $\mathcal{S}(\cdot)$, outputting an interest point head with $H \times W$ size and 65 channels, $\mathcal{P} \in \mathbb{R}^{H \times W \times 65}$, and descriptor head with $H \times W$ size and 256 channels, $\mathcal{D} \in R^{H \times W \times 256}$. Then we have $(\mathcal{P}_N, \mathcal{D}_N) = \mathcal{S}(N)$, and $(\mathcal{P}_O, \mathcal{D}_O) = \mathcal{S}(O)$. IR is defined as follows:

$$L_{\text{IR}} = L_{\mathcal{P}}(\mathcal{P}_N, \mathcal{P}_O) + \lambda L_{\mathcal{D}}(\mathcal{D}_N, \mathcal{D}_O), \qquad (8)$$

where $\lambda = 0.00005$. $L_{\mathcal{P}}$ is a square of ℓ_2 norm, that is,

$$L_{\mathcal{P}}(\mathcal{P}_N, \mathcal{P}_O) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \|\mathbf{p}_{hw}^N - \mathbf{p}_{hw}^O\|_2^2, \quad (9)$$

where $\mathbf{p}_{hw}^N$ and $\mathbf{p}_{hw}^O$ are 65-dimensional vectors belonged to $\mathcal{P}_N$ and $\mathcal{P}_O$, respectively. $L_{\mathcal{D}}$ is a hinge loss [15] with positive margin $m_p = 1$ and negative margin $m_n = 0.2$, that is, $L_{\mathcal{D}}(\mathcal{D}_N, \mathcal{D}_O) =$

$$\frac{1}{(HW)^2} \sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{i=1}^{H} \sum_{j=1}^{W} l_d(\mathbf{d}_{hw}^N, \mathbf{d}_{ij}^O; g_{hwij}), \qquad (10)$$

where $l_d(\mathbf{d}^N, \mathbf{d}^O; g) = \beta g \max(0, m_p - (\mathbf{d}^N)^T \mathbf{d}^O) + (1-g) \max(0, (\mathbf{d}^N)^T \mathbf{d}^O - m_n)$, g_{hwij} is a homography-induced correspondence between (h, w) and (i, j) cells,

$$g_{hwij} = \begin{cases} 1, & \text{if } \|\widehat{\mathcal{H}\mathbf{h}_{hw}^N} - \mathbf{h}_{ij}^O\| \le 8, \\ 0, & \text{otherwise}. \end{cases}, \ \mathbf{h}_{hw}^N \text{ denotes the}$$

location of the center pixel in the (h, w) cell, and $\widehat{\mathcal{H}\mathbf{h}_{hw}^N}$ denotes multiplying the cell location $\mathbf{h}_{hw}^N$ by the homography $\mathcal{H}$ and dividing by the last coordinate. By combining L_{NST} with L_{IR}, the ICTT loss is described as

$$L_{\text{ICTT}} = L_{\text{NST}} + \mu L_{\text{IR}}, \qquad (11)$$

where $\mu = 1$ controls the balance between NST and IR.

4. Experiments

In this section, we conduct extensive experiments to evaluate the visual product design ability of our InST approach, *e.g.*, company logos, bottles, porcelain fashions, and flying cars. More comparisons of product design are available in supplementary materials.

4.1. Experimental Settings

Dataset. consists of source and target products (or objects) and art style images. Following [56], the source products are selected from the Metropolitan Museum of Art Collection via the open-access API [1], and their segmentation masks are obtained by using the PointRend [30]. We use the clothes collected from the Zalando dataset [24] as the target products and get their segmentation masks by using the VI-TON [19]. The art style images are the WikiArt dataset [8]. In addition, the MS-COCO dataset [36] is also considered as the content images for training the network in the ICTT module. The input images are resized to 512×512. Each image is randomly cropped to 256×256 for training.

Training. Since our model includes the LGW and ICTT modules, our training schedule is decomposed into three steps. First, the source and target products are used to train the warping network of LGW. The hyper-parameters are set to $\{\alpha_r\}_{r=1}^3 = \{0.1, 0.2, 1\}$ in Eq. (3), $\{\beta_r\}_{r=1}^3 = \{0.1, 0.05, 0.01\}$ in Eq. (5), and $\gamma = 1$ in Eq. (7). Second, the art style images and MS-COCO as the content images are used to train the art transfer network of ICTT. The hyper-parameters are set to $\lambda = 0.00005$ in Eq. (8), and $\mu = 1$ in Eq. (11). Third, we jointly optimize the both the warping and art transfer networks using the collected dataset. In our experiment, We train these three steps for 50k/60k/10k iterations with a batch size 16/2/2 and the Adam [29] optimizer with a learning rate of 0.001/initial 0.0001 with decay of 0.00001/0.0001. Training takes roughly 10/12/8 hours on a single GTX 2080Ti GPU.

4.2. Main Results

To demonstrate that the proposed InST has a geometric and texture transfer ability to create a new product with wonderful visual appearances, we compare it with two recent geometric transfer methods, *e.g.*, DST [28] and GTST [38], and three texture transfer methods, *e.g.*, AdaIN [21], LinearWCT [32], and ArtFlow (content preservation) [3].

Visual comparisons. We showcase new visual products qualitatively from three aspects: (i) geometric warping, (ii) texture transfer, (iii) their combination.

Geometric warping. Fig. S7 shows the new product design results by the geometric style transfer algorithms. For example, the round earth and Rubik's cube are transferred into the logos of Twitter, Apple, Meta, McDonald's, and Jordan, respectively. Compared to the geometric methods,

Figure 6. Visual product design results using the geometric style transfer methods, *e.g.*, DST [28], GTST [38] and our InST. Compared to DST and GTST, our intermediate results between cars and aircraft have more reference values for the product designers as they are similar to the top view of the products (*e.g.*, Terrafugia and AeroMobil-4.0[1])
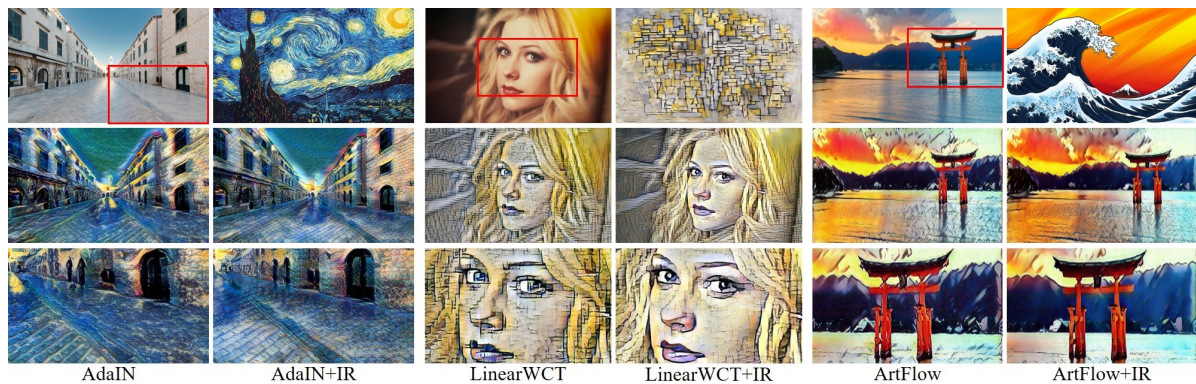

Figure 7. Content preservation results using the texture style transfer methods, *e.g.*, AdaIN [21], LinearWCT [32] and ArtFlow [3].

e.g., DST and GTST, our LGW module can better match the geometric shape of the target and better maintain the texture content of the source. The reasons for their failures are that DST and GTST only have little semantic relationships between two objects by using the corresponding key points [28] and learning a small-scale warping field [38], resulting in a worse result when facing large-scale geomet-

ric shapes. In contrast, we design a smooth mask warping field to adapt for large-scale warping in the visual product design.

Texture transfer. Fig. 7 shows the content preservation of the texture style transfer algorithms, *e.g.*, AdaIN, Linear-

---

[1]https://www.beautifullife.info/automotive-design/10-real-flying-cars/
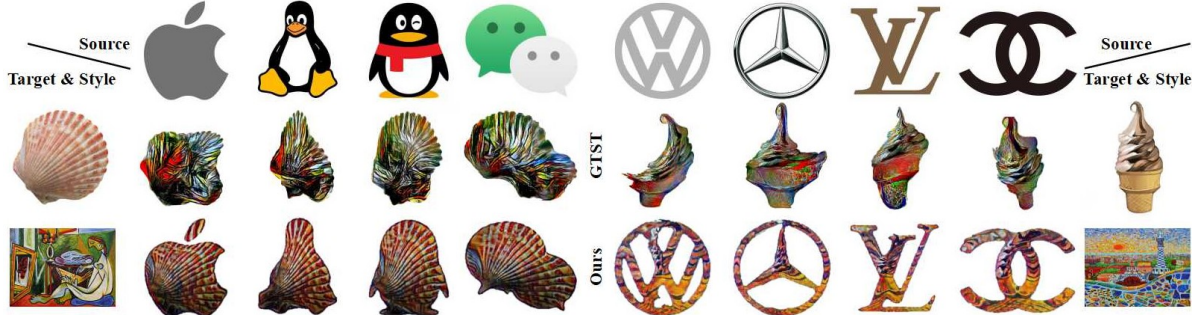
Figure 8. Visual logo design results using the geometric and texture style transfer methods, *e.g.*, GTST [38] and our InST.

Figure 9. Visual product design results using the geometric and texture style transfer methods, *e.g.*, GTST [38] and our InST.

Method	DST [28]	GTST [38]	our LGW
mIoU ↑	0.6000	0.7285	**0.9284**

Table 1. Quantitative evaluations of geometric warping methods.

Method	AdaIN [21]	AdaIN+IR	LinWCT [32]	LinWCT+IR	ArtFlow [3]	ArtFlow+IR
SSIM ↑	0.3424	**0.3886**	0.4612	**0.4932**	0.5042	**0.5643**
Time(s) ↓	0.054	0.054	0.419	**0.416**	**0.138**	0.140

Table 2. Quantitative evaluations of stylization methods.

WCT and ArtFlow. We can observe that our IR regularization can improve all the algorithms to preserve more content details since it perceives interest points to be similar. This is very different from ArtFlow as it considers reversible neural flows and unbiased feature transfer.

Geometric&texture transfer. We evaluate the overall product design with beautiful appearances based on the combination of geometric and texture style transfer against the state-of-the-art GTST [38]. Fig. 1 shows that our InST method is to create wonderful product appearances, such as the snail logo of Apple and Twitter. In addition, Figs. 8 and 9 also show more product design results. Compared to GTST, our method can provide larger-scale warping and retain more details of the source object (or product).

Quantitative comparisons. In addition to the above visual comparisons, we provide two quantitative comparisons for the LGW and IR modules. First, we evaluate the geometric warping performance with mean intersection-over intersection (mIoU), which is the popular metric for semantic segmentation [39]. In Table 1, we see that LGW has higher mIoU scores than DST and GTST. This means that the warping product better matches the target's geometry. Second, similar to [3], the Structural Similarity Index (SSIM) between the content and stylized images is considered as a metric to measure the performance of the detail preserva-

Method	Geometry			Texture		Combination	
	DST [28]	GTST [38]	Ours	TextMethod	Ours	GTST [38]	Ours
Votes↑	37	60	**1043**	337	**763**	134	**1006**

Table 3. Quantitative evaluations of user study. TextMethod denotes a set of AdaIN [21], LinWCT [32] and ArtFlow [3].

tion. Table 2 reports that these methods using our IR term have higher SSIM scores and can preserve more detailed information without additional test times.

User Study. We conduct a user study to evaluate the effect of the proposed InST algorithm against the existing methods. We divide the evaluation into three groups from the perspectives of geometric warping, content maintenance, and their combination, and each group includes ten options. In total, we collected 3420 votes from 114 users, and each group gets 1140 votes. Table 3 reports the results of the specific votes. Given the source and target products, 91.5% of users reported that our LGW network better matches the target's geometry, compared to only 5.3% for GTST [38] and 3.2% for DST [28]. In the content maintenance evaluation, 66.9% of users thought our ICTT module maintains more content details than the corresponding texture style transfer methods [3, 21, 32]. Finally, when evaluating the overall effect from the above two aspects, our proposed algorithm accounted for 88.2% of the 1140 votes compared to 11.8% for GTST [38]. Overall, our results were favorite among all aspects and evaluated methods.

4.3. Ablation Study

Since the comparative experiments on LWG and IR of ICTT have been available in the above subsection, we perform an ablation experiment on position embedding of the mask RAFT network in LWG. We test the importance of the

position embedding by training an LGW module without this component. Fig. 10 shows the comparison results with three recurrent updates. The position embedding achieves better performance because such an operation enhances the correlation of adjacent positions.
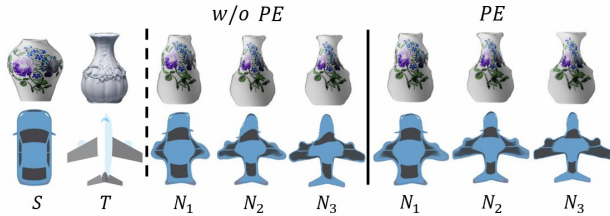

Figure 10. Ablation study on position embedding.

5. Discussion

In this section, we discuss three problems to better understand our mask RAFT and the limitations of our InST method. In addition, the potential applications are available in supplementary materials.

Why are RAFT [51] suitable for the geometric warping task? There are three reasons for explanations. 1) Optical-flow estimation is widely applied to estimate a warping between two moving geometries of objects in consecutive video frames by learning a warp field [22, 39, 44, 50, 51]. 2) Similar to the optical-flow estimation, a semantic transformer method [27] has been used to train a geometric warping filed between similar objects, named GTST [38], which is better than DST [28]. 3) RAFT [51] is state-of-the-art as it got the best paper award of ECCV 2020.

Why do we design a mask warping field? One reason is that it is difficult or impossible to directly warp RGB pixels of one object to match another when they are semantically irrelevant or have a big difference between their shapes, such as snail and Twitter logo. Another reason is that the difference between the two masks is lower than textural RGB images, leading to easier optimization. We train our LGW module with RGB images and their masks inputs and show the loss curves in Fig. 11. It is clear that using masks inputs has much lower loss and faster convergence than RGB. For further comparison, we also show their visualization results separately in Fig. 11, and obviously, mask RAFT has better deformation than RGB-based RAFT.

What differences between RAFT and mask RAFT? Compared to RAFT [51], our mask RAFT has the following four differences. Firstly, we design an unsupervised loss and a mask smoothness to learn a large-scale warping field, while RAFT explores a small-scale light flow field in a supervised setting. Secondly, before RAFT, we introduce a mask extraction stage to obtain object (or product) mask from its RGB image. Thirdly, we present a position embedding for the feature extraction behind enhancing the correlation of adjacent positions. Fourthly, we use the feature $\hat{F}_t$ of the
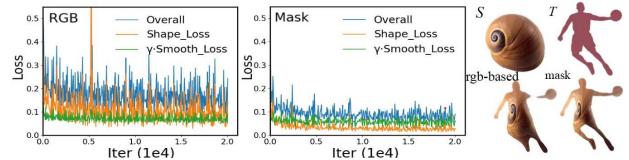

Figure 11. Losses of our LGW module with RGB and mask inputs.

target instead of the feature extraction using another network. Overall, our mask RAFT can better warp large-scale geometric shapes.

Limitations. Here, we discuss the limitations of geometric warping. Because our purpose is to realize a large-scale warping field between products (or objects), which has a little semantic correspondence, we do not rely on semantic information to guide the warping field. When input pairs share semantic attributes, our method may generate counter-intuitive results. For example, in Fig. 12, our LGW method tries to match the shapes regardless of internal semantic alignment, such as aligning eyes with eyes.

Figure 12. Limitation: The in-principle limit is semantic corresponding between similar objects.

6. Conclusion

In this paper, we proposed an industrial style transfer method for visual product design tasks. Our method constructed a geometric transformation field to create a new product and further learned a style transformation network to transfer an art style of the reference image to the new product. It is worth mentioning that our method warped a source product to imitate the geometric shape of the target product even if they are not semantically relevant. Extensive experiments demonstrated that our method outperforms the state-of-the-art style transfer algorithms, particularly the challenging large-scale geometric shapes. We also applied the style transfer pipeline into some product design tasks, *e.g.*, amazing logos, beautiful bottles, flying cars and porcelain fashions. Hopefully, our work can open an avenue to assist or inspire designers to design new industrial products by using style transfer techniques.

Acknowledgements

References

[1] The metropolitan museum of art open access. https:// met-museum.github.io. 2020. 5

[2] Kfir Aberman, Jing Liao, Mingyi Shi, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Neural best-buddies: Sparse cross-domain correspondence. *ACM Trans. Graph. (TOG)*, 37(4):1–14, 2018. 3

[3] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *CVPR*, pages 862–871, 2021. 2, 3, 5, 6, 7, 14

[4] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002. 3

[5] P.H. Bloch. Seeking the ideal form: Product design and consumer response. *Journal of Marketing*, 59(July):16–29, 1995. 2

[6] P.H. Bloch, F.F. Brunel, and T.J. Arnold. Individual differences in the centrality of visual product aesthetics: Concept and measurement. *Journal of Consumer Research*, 29(March):551C565, 2003. 2

[7] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:567–585, 1989. 3

[8] K Nichol. Painter by numbers. https://www.kaggle.com/c/painter-by-numbers. 2016. 5

[9] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: an explicit representation for neural image style transfer. In *CVPR*, pages 2770–2779, 2017. 2

[10] Ming-Ming Cheng, Xiao-Chang Liu, Jie Wang, Shao-Ping Lu, Yu-Kun Lai, and Paul L Rosin. Structure-preserving neural style transfer. *IEEE Trans. on Image Process.*, 29:909–920, 2019. 3

[11] M.E.H. Creusen and Jan P. L. Schoormans. The different roles of product appearance in consumer choice. *Journal of Product Innovation Management*, 22(1):63C81, 2005. 1, 2

[12] N. Crilly, J. Moultrie, and P.J. Clarkson. Seeing things: Consumer response to the visual domain in product design. *Design Studies*, 25(6):547–577, 2004. 1, 2

[13] Navneet Dalal and Bill Triggs. Histogram of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 3

[14] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. In *AAAI*, pages 1210–1217, 2021. 1

[15] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshop*, pages 337–349, 2018. 2, 3, 5

[16] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. 1, 2

[17] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *CVPR*, pages 3985–3993, 2017. 2

[18] Bumsub Ham, Minsu Cho, C. Schmid, and J. Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40:1711–1725, 2018. 3

[19] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and L. Davis. Viton: An image-based virtual try-on network. In *CVPR*, pages 7543–7552, 2018. 5

[20] Kibeom Hong, Seogkyu Jeon, Huan Yang, Jianlong Fu, and Hyeran Byun. Domain-aware universal style transfer. In *ICCV*, pages 14609–14617, 2021. 3

[21] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017. 1, 2, 3, 5, 6, 7, 14

[22] Junhwa Hur and S. Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, pages 5754–5763, 2019. 8

[23] Max Jaderberg, K. Simonyan, Andrew Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 4

[24] Nikolay Jetchev and Urs Bergmann. The conditional analogy gan: Swapping fashion articles on people images. In *ICCVW*, 2017. 5

[25] Shuhui Jiang, Jun Li, and Yun Fu. Deep learning for fashion style generation. *IEEE Trans. Neural Netw. Learn. Syst.*, doi:10.1109/TNNLS.2021.3057892:1–13, 2021. 1

[26] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016. 2

[27] Seungryong Kim, S. Lin, Sangryul Jeon, Dongbo Min, and K. Sohn. Recurrent transformer networks for semantic correspondence. In *NIPS*, pages 6129–C6139, 2018. 3, 8

[28] Sunnie S. Y. Kim, Nicholas I. Kolkin, Jason Salavon, and Gregory Shakhnarovich. Deformable style transfer. In *ECCV*, pages 246–261, 2020. 1, 2, 3, 5, 6, 7, 8, 13

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

[30] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, pages 9799–9808, 2020. 3, 5

[31] Dmytro Kotovenko, Matthias Wright, Arthur Heimbrecht, and Bjorn Ommer. Rethinking style transfer: From pixels to parameterized brushstrokes. In *CVPR*, pages 12196–12205, 2021. 3

[32] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. In *CVPR*, pages 3809–3817, 2019. 1, 3, 5, 6, 7, 14

[33] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NIPS*, page 385C395, 2017. 2, 3

[34] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *IJCAI*, pages 2230–2236, 2017. 2

[35] Tianwei Lin, Zhuoqi Ma, Fu Li, Dongliang He, Xin Li, Errui Ding, Nannan Wang, Jie Li, and Xinbo Gao. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In *CVPR*, pages 5141–5150, 2021. 3

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 5

[37] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *ICCV*, pages 6649–6658, 2021. 3

[38] Xiao-Chang Liu, Yong-Liang Yang, and Peter Hall. Learning to warp for style transfer. In *CVPR*, pages 3702–3711, 2021. 1, 2, 3, 5, 6, 7, 8, 13, 14, 15

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 7, 8

[40] G. LoweDavid. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[41] Ruth Mugge, Darren W. Dahl, and Jan P.L. Schoormans. what you see, is what you get? guidelines for influencing consumers perceptions of consumer durables through product appearance. *Journal of Product Innovation Management*, 35(3):309C329, 2018. 2

[42] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, pages 5880–5888, 2019. 3

[43] Scott K. Radford and Peter H. Bloch. Linking innovation to design: Consumer responses to visualproduct newness. *Journal of Product Innovation Management*, 28(s1):208C220, 2011. 2

[44] A. Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, pages 2720–2729, 2017. 8

[45] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. 2

[46] Ignacio Rocco, R. Arandjelović, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, pages 39–48, 2017. 3

[47] Ignacio Rocco, R. Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, pages 6917–6925, 2018. 3

[48] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: multi-scale zero-shot style transfer by feature decoration. In *CVPR*, pages 8242–8250, 2018. 3

[49] Yichun Shi, Debayan Deb, and Anil K Jain. Warpgan: Automatic caricature generation. In *CVPR*, pages 10762–10771, 2019. 3

[50] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 8

[51] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 3, 4, 8

[52] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: feed-forward synthesis of textures and stylized images. In *ICML*, volume 1, page 4, 2016. 2

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3

[54] Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. Collaborative distillation for ultra-resolution universal style transfer. In *CVPR*, pages 1860–1869, 2020. 3

[55] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. Diversified arbitrary style transfer via deep feature perturbation. In *CVPR*, pages 7789–7798, 2020. 3

[56] Shangzhe Wu, Ameesh Makadia, Jiajun Wu, Noah Snavely, Richard Tucker, and Angjoo Kanazawa. De-rendering the world's revolutionary artefacts. In *CVPR*, pages 6338–6347, 2021. 5

[57] Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *ICCV*, pages 14618–14627, 2021. 3

[58] Xide Xia, Meng Zhang, Tianfan Xue, Zheng Sun, Hui Fang, Brian Kulis, and Jiawen Chen. Joint bilateral learning for real-time universal photorealistic style transfer. In *ECCV*, pages 327–342. Springer, 2020. 3

[59] Jia xin Cheng, Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Prem Natarajan. Style-aware normalized loss for improving arbitrary style transfer. In *CVPR*, pages 134–143, 2021. 1

[60] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *ICCV*, pages 4442–4451, 2019. 1, 3

[61] Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: landmark detection and geometric style in portraits. *ACM Trans. Graph. (TOG)*, 38(4):1–15, 2019. 3

[62] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. In *ECCV workshop*, pages 0–0, 2018. 2

[63] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. Multimodal style transfer via graph cuts. In *ICCV*, pages 5943–5951, 2019. 3