IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2020, accepted December 27, 2020, date of publication January 1, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048879

Perspective Preserving Style Transfer for Interior Portraits

WEN-YIN CHEN[1], JOSE JAENA MARI OPLE[2], MAYNARD JOHN SI[2], DANIEL STANLEY TAN[2], AND KAI-LUNG HUA[2], (Member, IEEE)

[1]Department of Arts and Design, National Taipei University of Education, Taipei 106320, Taiwan
[2]Department of Computer Science and Information Technology, National Taiwan University of Science and Technology, Taipei 106335, Taiwan

Corresponding author: Kai-Lung Hua (hua@mail.ntust.edu.tw)

This work was supported by the Center for Cyber-Physical System Innovation and the Center of Intelligent Robots from The Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan and the Ministry of Science and Technology of Taiwan under Grant MOST109-2218-E-011-010, Grant MOST109-2221-E-011-125-MY3, and Grant MOST109-2221-E-001-016.

ABSTRACT One of the jarring limitations of existing style transfer techniques is their failure to capture the illusion of depth through perspective. These often result in flat-looking images that have their style elements simply distributed across the image. Though recent methods attempt to alleviate this by considering depth information for a distinct stylization between foreground and background, they still fail to capture an image's perspective. When used on interior portraits where perspective is instinctively observed through its surfaces (walls, ceiling, floor), previous methods cause unwanted styling such as style elements distorting boundaries of surfaces and style elements not receding according to the perspective of the surfaces. In this paper, we developed a novel approach to effectively preserve interior portraits' perspective during style transfer, yielding stylized images that distribute and warps style elements according to the interior surfaces' perspective. Our method involves removing the perspective information from an interior portrait image, such that when performing style transfer the image can be considered as a flat perspective-neutral canvas. After that, we restore the perspective to the image leading to its style elements recede towards the vanishing point of its respective surface. We also observe that our approach was able to preserve depth information for some styles despite not extracting depth maps from the content.

INDEX TERMS Neural style transfer, image processing, image filters.

I. INTRODUCTION

The expression of one's self through artistic means is one of the fascinating aspects of human culture. Imagine having the creative ability to capture the beauty of Vincent Van Gogh's Starry Night and translating this into your envisioned scenery. In the past, you would have to be an expert artist having a lot of time to mimic the style of another artwork. This task of rendering a style from one image to another is known as style transfer. The need for technology that tackles this task is driven by the rise in popularity of photographic filters in platforms such as social media (e.g. Instagram [1], Snapchat [2]), Extended Reality (XR) environments (e.g Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) [3]), and, recently, real-time in-game [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqi Wang.

Gatys *et al.* [5] were the first to use Convolutional Neural Networks (CNN) in replicating painting styles of famous artists and applying them to normal day-to-day images. Since then, there has been a focus on improving the Neural Style Transfer (NST) in terms of stylization quality [6]–[14].

However, these approaches fail to capture perspective, an important characteristic of artworks that gives off the illusion of depth and contribute to the viewer's immersion. We can observe that style elements tend to be simply distributed evenly across the whole image, resulting in flat-looking stylized images. With perspective, objects appear smaller as their distance from the observer increases [15]. In the context of style transfer, this would mean warping (stretching or shrinking) the style elements depending on the perceived distance. For interior portraits, perspective can easily be inferred due to the presence of multiple parallel lines that converge into one or two vanishing points [16] completed by the different surfaces of the room (Fig. 1).

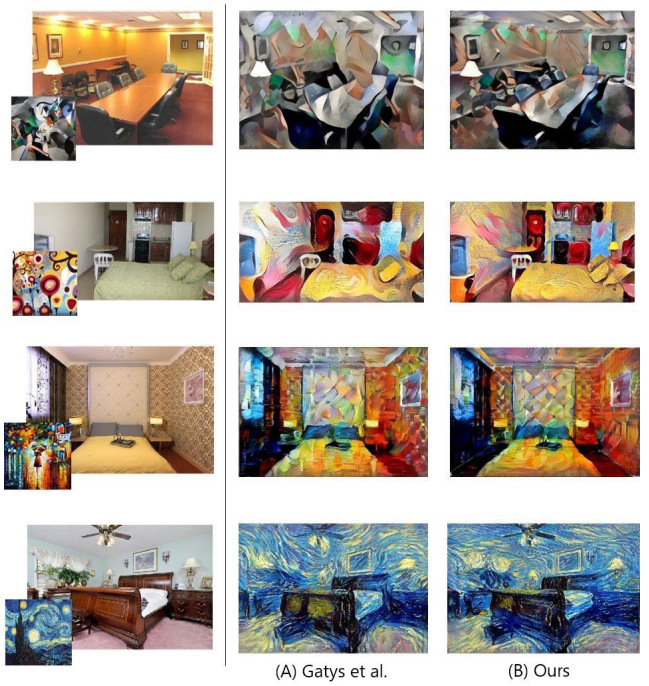

FIGURE 1. Examples of interior portrait style transfer results. We compare our method with the Neural Style Transfer introduced by Gatys *et al.* [5] (A). We can see that our method (B) was able to stylize interior surfaces according to their perspective, resulting in images that does not distort boundaries between surfaces and objects compared to Gatys *et al.* [5].

There are no style transfer works that deal directly with perspective. Alternatively, there are works for depth preserving style transfer, which indirectly preserve perspective since perspective implies depth. The style transfer methods by [14], [17] simply focus on distinguishing the foreground from the background. They merely adjust the saturation and density of style elements in relation to the foreground and background, causing foreground style elements to appear weaker than what is seen in the background. Another work by [9] uses depth maps ground-truths to train a feed-forward style transfer network [6].

Our goal is to capture perspective when performing style transfer. Using depth maps to capture perspective is a viable strategy, however, we could deal directly with perspective. We propose a framework for style transfer that effectively preserves an interior portrait image's perspective. We summarized our method into three major steps: (1) Perspective Removal, (2) Style Transfer, and (3) Perspective Restoration. Our main idea is to prevent the style transfer process from affecting the perspective information by temporarily removing it then restoring it after the style transfer. In Perspective Removal (step 1), we identify the keypoints/corners of major surfaces of a room interior (i.e. floor, ceiling, walls) using a keypoint detector network [18]. Then, we perform four-point transforms to generate a front-view rendering of each major surface and stitch them together to form a perspective-neutral image. In step 2, we perform style transfer to the image using a slow image optimization algorithm [5] to produce a stylized image with a neutral perspective. Finally, we restore the perspective to the stylized image by reversing the transformation performed in step 1. These procedures ensure that perspective is agnostic to the style transfer algorithm, and generated style elements will follow the perspective.

Experiments on images of various interior portrait layouts demonstrate our system's effectiveness to preserve perspective and depth in image style transfer. A preview of our results can be seen in Figure 1. In summary, our contributions are as follows:

- We developed a generalized method that allows style transfer algorithms to have perspective preservation for interior portraits.
- We implemented an algorithm that removed perspective from major room surfaces (i.e. floor, ceiling, walls) using keypoint estimation and four-point transforms.
- Our experiments show that our method can inherently preserve depth even without using depth-maps as guidance.

II. RELATED LITERATURE

Style transfer is the task of applying the style of one image to another image. It has been present for more than two decades but it wasn't until recently that rapid progress has been made. Older approaches were limited in terms of flexibility and stylization quality. The papers [19]–[24] can only process low-level features (e.g. textures, color ambiance). Other papers [24], [25] require expert knowledge and cannot be applied to arbitrary images. Recently, in a seminal work by Gatys *et al.* [5], we saw an artificial system able to replicate the painting styles of famous artists and apply them to normal day-to-day images. The work leverages on using a pre-trained Convolutional Neural Network (CNN) to extract feature responses to represent the content of a photo as well as using the summary statistics of the extracted features to represent the style. This process of using CNNs for this domain is referred to as Neural Style Transfer (NST).

Since the emergence of NST algorithms, there had been works devoted to improving the stylization quality of NST algorithms by controlling perceptual factors (e.g. stroke size, spatial style, and color control) [7]–[9], [13] and address specific tasks (e.g. head portraits [26], [27]). Johnson *et al.* [6] and Huang and Belongie [11] were able to speed up the process by training a feed-forward network and produce a stylized image in a single forward pass. Despite the advancements, these approaches still suffer from the same issue of producing images that don't capture the perspective or, alternatively, depth.

The work of [17] used the network from [6] but adjusted the content and style elements by determining whether image regions are foreground (stronger content) or background (stronger style). Similar to [14], [17] used depth information to modify the stylization strength, but unlike [17], their stylization is much more granular. For [17] and [14], although depth information captures the scene layout (i.e. boundary

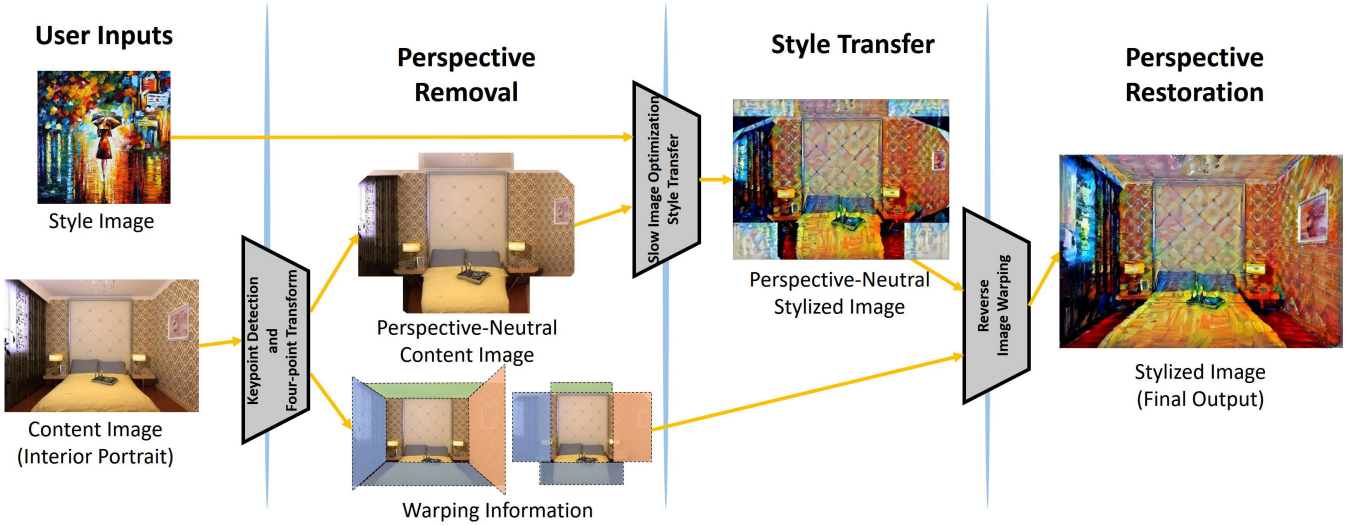

FIGURE 2. **Framework overview. Our method has three major steps. (1) Perspective Removal: Generates a perspective-neutral rendering of the content image and also outputs the necessary warping information to reverse the rendering. (2) Style Transfer: Can be any style transfer algorithm that can work with flat (perspective-neutral) images to generate a stylized image. (3) Perspective Restoration: Reverses the warping of the image such that perspective is applied to the style elements (e.g. brush stroke).**

between the foreground and background), the synthesized image still fails to capture the perspective of the content image and simply distributes style elements based on the foreground and background. The work [9] can capture depth by directly using depth maps as ground-truths for training a feed-forward style transfer network [6].

III. METHODOLOGY

The objective of our method is to generate a stylized image X' given an interior portrait image X such that perspective is preserved even after stylization using the style image I_s. Our method is composed of three major steps: (1) Perspective Removal, (2) Style Transfer, and (3) Perspective Restoration. Our main idea is to initially dissociate perspective from the input image so that perspective information will not be mutated by the style transfer algorithm. After performing style transfer, we restore the untouched perspective information to the stylized perspective-neutral image. The overview of our method can be seen in Fig. 2.

A. PERSPECTIVE REMOVAL

We want to separate the perspective information from the interior portrait X to generate a perspective-neutral image X_m. We assume that X subscribes to the Manhattan world structure [28], which dictates that surfaces are aligned with three dominant directions, typically corresponding to the x, y, and z axes. Under this assumption [28], an interior portrait can be represented as a cuboid with at most five surfaces visible in an image, namely P_{left}, P_{front}, P_{right}, $P_{ceiling}$, and P_{floor}. The assumption simplifies our perspective removal but makes our method unable to handle non-Manhattan structures. Each surface is assumed to have a rectangular structure but due to perspective, their shape is warped in the interior portrait. We generate X_m by rendering each surface in their rectangular front-facing view. Our Perspective Removal procedure consists of two steps: (1) Find the four keypoints (corners) of each surface, (2) Perform a four-point transform to each surface to render their rectangular front-facing view. The illustration of our procedure can be seen in Fig. 4.

1) KEYPOINT DETECTION NETWORK

The goal is to identify the four corner points of each surface to be used in the perspective transformation. To infer the boundaries of each surface, we utilized an encoder-decoder network tasked for the end-to-end inference of keypoints of an interior portrait [18]. The end-to-end network takes the image of an interior portrait and directly outputs a set of 2D room layout keypoint heatmap. A ground truth keypoint heatmap was generated by centering a 2D Gaussian at every keypoint. An encoder-decoder network is then tasked to regress to the ground truth heatmap for each of the six-room types depicted in Fig. 3. An auxiliary network that branches from the bottleneck layer of the encoder-decoder is then tasked to predict which of the six-room types the input interior portrait belongs to. Once the room type class label of the input image has been identified, the associated keypoint heatmap is selected. An illustration of this architecture is shown in Figure 3.

A training example is denoted as (X, y, t), where X is the input image, and y stands for the ground-truth coordinates of the keypoints k with room type t. During training, the cost functions are Euclidean loss and cross-entropy loss for the keypoint heatmap regression and room type prediction respectively. Given the keypoint heatmap regressor φ and room type classifier ψ, the following loss function is optimized during training:

$$\sum_k \mathbb{1}_{k,t}^{kpt} ||G_k(y) - \varphi_k(X)||^2 - \lambda \sum_k \mathbb{1}_{c,t}^{room} \log(\psi_c(I)) \quad (1)$$

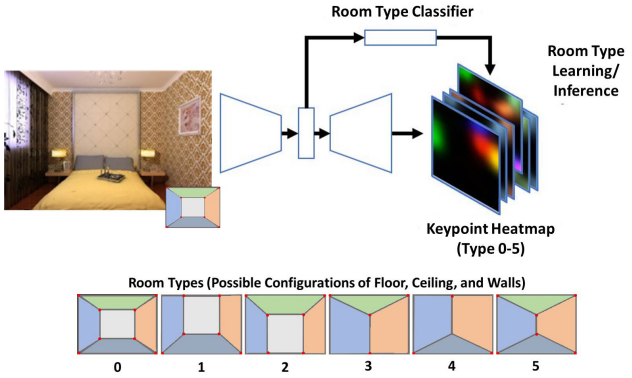

FIGURE 3. Simplified illustration of the keypoint detection architecture [18]. An encoder-decoder network outputs keypoint heatmaps for each room type. A room type classifier side network predicts the room type class label which is used to select the associated keypoint heatmap. At the bottom are the possible room layouts containing perspective.

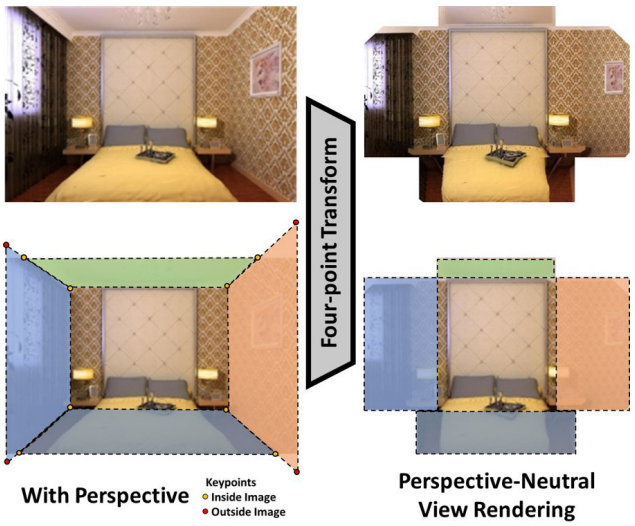

FIGURE 4. Removing perspective from an interior portrait by finding keypoints and performing four-point transform. Keypoints of each major surface are estimated using [18]. Some keypoints are found inside the image (yellow dots) while some keypoints are found outside the image (red dots) so we extend the boundary lines of the surfaces.

where we have an indicator function $\mathbb{1}_{k,t}^{kpt}$ which denotes if keypoint k appears in the ground truth room type t, and indicator function $\mathbb{1}_{c,t}^{room}$ which denotes if room type index c is equal to the ground truth room type t, G is the Gaussian centered at y, and λ being the weight term set to 5 by cross validation. The first term compares the predicted heatmaps to ground truth heatmaps for each keypoint while the second term encourages the auxiliary network to produce a high confidence value with respect to the associated room type.

2) SELECTING NEW SURFACE KEYPOINTS

As seen in Fig. 4 and Fig. 5, some keypoints are outside the image. We are unable to obtain all four corner points of a surface through the keypoint detection network alone as some may lie beyond the edge of the image. If we were to use these

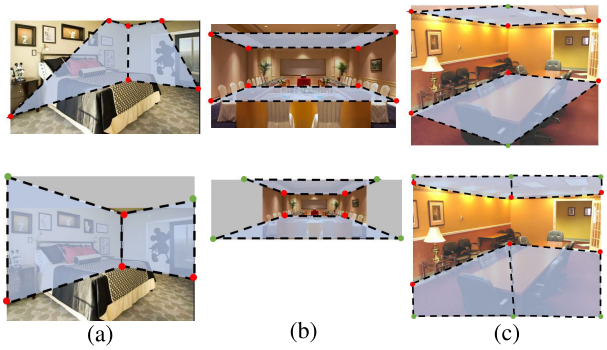

FIGURE 5. Examples of surfaces formed by extracting 4 keypoints. (a) depicts a case when forming the wall surfaces, (b) & (c) depict cases for the ceiling and floor surfaces. The top row images show the surfaces formed from the initial extracted points from the network. The bottom row images show the surfaces formed after selecting new keypoints.

points, we would be unable to cover the whole surface during the succeeding steps of the framework (as seen in the top row of Figure 5). Thus, we select our surface's new corner points by projecting its boundary lines beyond the image and selecting the points where these lines intersect (as seen in Figure 5 (a) & (b)). There are some surfaces where only 3 keypoints could be extracted. This is a challenge because simply adding another point placed at the bottom-middle area of the image would not account for other areas of the surface (as seen in the top row of Figure 5 (c)). For these cases, we divide the surface into two sub-surfaces. We do this by introducing 3 new points with one placed between two points at either the top (ceiling) or bottom (floor) of the image and the remaining points placed at the respective corners (as seen in the bottom row of Figure 5 (c)). Our interior surface plane can then be defined as:

$$P_s = (\rho_1, \rho_2, \rho_3, \rho_4) \tag{2}$$

where P_s stands for the perspective projection plane of interior surface P_{left}, P_{front}, P_{right}, $P_{ceiling}$, and P_{floor} with points ρ_1, ρ_2, ρ_3, ρ_4 being the top-left, top-right, bottom-right, bottom-left vertices respectively. Depending on the room layout, P_s may be rectangular, trapezoidal, or rhomboid. We crop our surface planes from the rest of the image to avoid outer distortions during transformation.

3) INTERIOR SURFACE MAP CONSTRUCTION

We obtain $\hat{\boldsymbol{\mathcal{X}}}$ by capturing front-facing views P_s' for each surface P_s(front view of walls, bottom view of ceiling, top view of floor). To do this, we subject a surface P_s to perspective transformation.

First, we find the 3 x 3 homography matrix M that maps the source points of the surface ρ_1, ρ_2, ρ_3, ρ_4 to points ρ_1', ρ_2', ρ_3', ρ_4' which we simply defined as a rectangular box based on the dimensions of the surface. This mapping can be defined as:

$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} = \begin{bmatrix} M_{11}M_{12}M_{13} \\ M_{21}M_{22}M_{23} \\ M_{31}M_{32}M_{33} \end{bmatrix} \times \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{3}$$

where x_i and y_i represent the x and y coordinates of source point p_i and x_i' and y_i' represent the x and y coordinates of destination point p_i'. For each destination pixel, we assign the pixel value from the fractional coordinates of the original surface image given by the product of the destination pixel coordinates and homography matrix M. P_s' at pixel x_i' and y_i' is given by:

$$P_s'(x_i', y_i') = P\Big(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\Big) \quad (4)$$

Since we are unable to use fractional coordinates in looking up pixel values from the input image, we interpolate these to its nearest neighbor.

We then construct our interior surface map X_m by concatenating each surface P_s' according to their relative position. We are also able to manipulate the intensity of which style elements recede by manipulating the destination points during perspective transformation such that the outer edge points (e.g. top left and bottom left point of the left wall) are drawn to the middle of their boundary line. This would result in a warped surface in the shape of a trapezoid with the edge points serving as the vertices of its short base. This would stretch out style elements upon perspective image reconstruction.

B. STYLE TRANSFER

The goal of our style transfer is to synthesize a stylized interior surface map X_m', given a style image I_s and the constructed interior surface map as the content image X_m. Our style transfer pipeline is then given as:

$$X_m' = \mathbb{F}(X_m, I_s) \quad (5)$$

where our $\mathbb{F}$ is our style transfer technique. For our baseline, $\mathbb{F}$ is inspired by the approach of [5]. By using intermediate layers of the VGG network, we are able to extract the content and styles representations of an interior portrait as well as a given artwork respectively. The content component of the newly stylized interior surface map X_m' is captured by penalizing the difference of the representations derived from content X_m and the stylized interior surface map. For the style component, it has been observed that Gram matrices represent style as it encodes the second order statistics of the set of filter responses (correlations between filter responses in different layers). Thus, the style component by matching Gram-based summary statistics of style I_s and stylised images X_m'. The details for capturing each are as follows.

Given the interior surface map X_m and style image I_s, we synthesize a stylized interior surface map X_m' which minimizes the loss function:

$$\mathcal{L}_{total}(X_m, I_s, X_m') = \alpha\mathcal{L}_{content}(X_m, X_m') + \beta\mathcal{L}_{style}(I_s, X_m') \quad (6)$$

where $\mathcal{L}_{total}$ is the sum of the content loss $\mathcal{L}_{content}$ and style loss $\mathcal{L}_{style}$ with α and β being the weighting factors that balances these.

The content loss $\mathcal{L}_{content}$ compares the content representation of a given content image to that of the stylized image. This defined by the squared Euclidean distance between the feature representations $\mathcal{F}^l$ of the interior surface map X_m and stylized interior surface map X_m' in layer l:

$$\mathcal{L}_{content}(X_m, X_m') = \sum_{l\in\{l_s\}} ||\mathcal{F}^l(X_m) - \mathcal{F}^l(X_m')||^2 \quad (7)$$

where $\{l_s\}$ denotes the set of chosen VGG layers that represents the content of the image.

The style loss compares the Gram-based style representation of a style image to that of the stylised image. This is defined by the squared Euclidean distance between the Gram-matrices of the feature representations $\mathcal{F}^l$ of style image I_s and stylized interior surface map X_m' in layer l:

$$\mathcal{L}_{style}(I_s, X_m') = \sum_{l\in\{l_s\}} ||\mathcal{G}(\mathcal{F}^l(I_s)) - \mathcal{G}(\mathcal{F}^l(X_m'))||^2 \quad (8)$$

where $\mathcal{G}$ is the aforementioned Gram matrix to encode the second order statistics given by the set of chosen VGG layers $\{l_s\}$ that represents the style of the image.

C. PERSPECTIVE RESTORATION

Given our stylized interior surface map X_m' and the bounding boxes for each surface P_s', we map the projection of surface P_s' back to its original perspective projection P_s through the inverse homography matrix that maps the destination points ρ_1', ρ_2', ρ_3', ρ_4' back to the source points ρ_1, ρ_2, ρ_3, ρ_4 (Eq. 3 & 4). Each of the surface represents an area of the original image; therefore, we reconstruct our stylized perspective image by returning the surfaces to their original location and blending them together.

IV. RESULTS AND DISCUSSION

The goal of our method is to allow perspective preservation to arbitrary style transfer algorithms. To evaluate our performance, we examine our algorithm in terms of perspective preservation, depth map fidelity, and stylization quality. Examples of the outputs of our method can be seen in Fig. 6.

A. EXPERIMENTAL DETAILS

We resize the RGB image to a resolution of 320 × 320. We scale the resulting interior surface map down to have the larger dimension as 512 if ever it exceeds the resolution of 512 × 512. For keypoint detection, we used the architecture from [18] and trained the network using the LSUN [29] dataset which has 4000 training, 394 validation, and 1000 test images. Though our framework can utilize any style transfer technique, we only used Gatys *et al.* [5] and Huang and Belongie [11] as our implementation. For [5], it uses a pretrained VGG-19 network [30], which was trained on ImageNet [31] with 1.3M training and 50K validation images. For [11], we used a pretrained network, which was trained on MS-COCO [32] dataset with around 80K training images. Using the VGG network, we use the *relu*4_2 layer for the content loss and layers *relu*1_1, *relu*2_1, *relu*3_1,

FIGURE 6. Examples of perspective preserving style transfer on 6 different room layouts and styles.

*relu*4_1, *relu*5_1 for the style loss. We opt to initialize our style transfer's output image as the interior surface map. We set the number of iterations to 1000. We simply preserve the content and style weighting factors as 1.0. For testing our stylization, we use 100 bedroom images from LSUN [29] test dataset and 100 indoor images from NYUv2 [33] test dataset.

B. PERSPECTIVE PRESERVATION

Our method preserves perspective by performing transformation such that perspective information is removed from it. Then, we perform style transfer on that perspective-neutral image. The underlying style transfer we used is from Gatys et al [5], as such, we compare our perspective-preserving method to the baseline [5]. To examine the effectiveness of our method, we refer to Fig. 7 and Fig. 8.

We can observe whether style elements can preserve perspective when it recedes with respect to a vanishing point. Style transfer can result in irregular-looking style elements which would make it difficult to objectively observe. To address this, we chose a style image rich in thick visible horizontal lines that are repeating in a predictable manner.

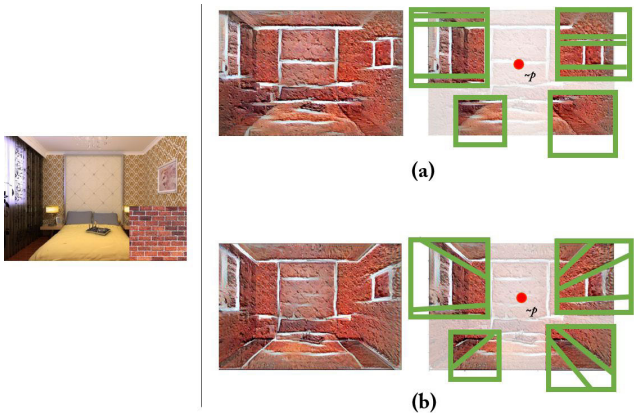

FIGURE 7. Perspective preservation comparison with Gatys *et al.* [5]. When projecting lines across visible style elements, the original approach produces lines that do not intersect or approach the vanishing point *p*. In contrast, our approach produces lines that better retain the perspective with projected lines approximately intersecting with the vanishing point *p*.

This would yield a result that is reminiscent of the characteristics of the style image. Lines or object boundaries in the

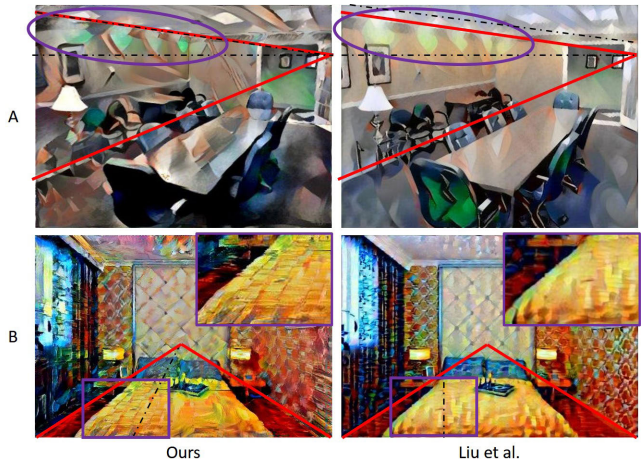

FIGURE 8. Comparing our method to another depth-preserving style transfer [9]. Despite having similar depth maps, the style elements of our method follows the perspective of the image. (A) The purple encircled style elements in our image are shrinking the further it is from the viewpoint while Liu *et al.* [9] have haphazard distribution. (B) The strokes in our image follow perspective while [9] has vertical strokes.

resulting image may inherit the lines from the style image as seen in Fig. 7.

In Fig. 8, we qualitatively compare our method to a depth-preserving style transfer, Liu *et al.* [9]. Both the outputs of our method and [9] have depth-preservation. However, only in our output does the style elements generally obey the perspective. In Row A, we could see that the style elements of our method shrink towards the vanishing points, which is a characteristic of perspective. But for Liu *et al.* [9], even if the depth is preserved, the style elements do not obey perspective. In Row B, the strokes of our method similarly approach the vanishing points while the other method does not. If we look at the zoomed-in patches enclosed by purple boxes the sense of perspective is much more prominent.

C. DEPTH MAP COMPARISON

Qualitative analysis is not enough so we perform quantitative experiments to prove that our method enables perspective preservation. We use depth maps of the stylized image as a substitute for detecting perspective. Depth maps contain 3D feature information of its objects, and a preserved depth map would indicate that the spatial characteristics (e.g. perspective) of the content image are retained. Though it is impossible to completely preserve the content image's depth map during style transfer, we want the depth map of our results to resemble that of the content image as much as possible, especially for interior portraits where depth is critical. We can recover depth information through the use of a network tasked to recover depth information from single images [34].

In Fig. 9, we compare the depth maps of the content image and the depth maps from the resulting style transfer methods. We can observe that we were able to yield a depth map that resembles the original structure of the interior portrait. This is in contrast to [5] wherein the boundaries of surfaces and objects in the scene are disfigured — likely caused by the stylization. Note that we do not extract or use any form of depth information for our framework, this was simply a result of preserving the perspective of the interior surfaces. In Fig 9, Liu *et al.* [9] and our method have similar depth map output. But note that our method does not use depth

TABLE 1. Comparison of the mean squared error between the predicted depth maps of the content image and the stylized image. The content images are the bedroom images in LSUN test dataset [29].

	Style 1	Style 2	Style 3	Style 4	Style 5
Gatys et al. [5]	171.15	222.85	284.36	270.42	274.56
Liu et al. [11]	**89.49**	126.67	**126.73**	**131.17**	173.30
Ours	93.48	**125.35**	128.62	141.06	**173.17**

TABLE 2. Comparison of the mean squared error between the predicted depth maps of the content image and the stylized image. The content images are selected images in NYUv2 test dataset [33] that satisfies Manhattan-world assumption [28].

	Style 1	Style 2	Style 3	Style 4	Style 5
Gatys et al. [5]	274.44	158.91	149.74	175.70	127.18
Liu et al. [9]	214.59	**152.77**	**96.29**	**71.90**	**33.59**
Ours	**210.19**	154.98	102.60	73.15	34.22

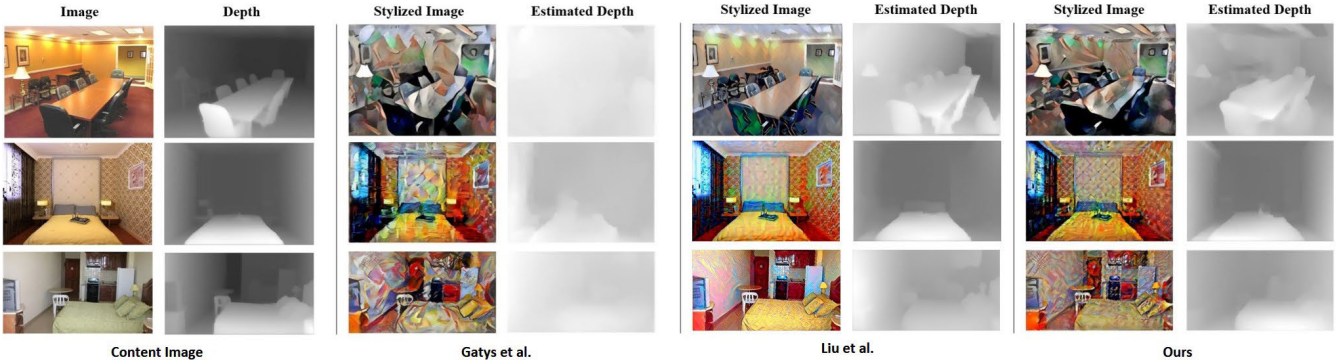

FIGURE 9. Depth map comparison with Gatys *et al.* [5] and Liu *et al.* [9]. Despite not using any depth information aside from the perspective, our approach was able to produce a depth map that resembles that of the original interior portrait. In contrast, the resulting depth map of Gatys *et al.* [5] fails to capture and disfigures the structure of the interior portrait.

TABLE 3. Comparison between baseline performance (i.e. content, style, depth) of the underlying style transfer method [5], [11] and the performance with perspective preservation (PP). Bold values mean it has the best performance for *baseline vs with PP*.

Style Transfer Method	Content		Style		Depth	
	baseline	with PP	baseline	with PP	baseline	with PP
Gatys et al. [5]	407.47	**301.79**	102.41	**88.64**	335.81	**163.73**
Huang et al. [11]	**254.45**	281.51	92.92	**82.27**	120.67	**31.52**

FIGURE 10. Output comparison between the baseline style transfer method: (B) Gatys *et al.* [5] and (C) Huang *et al.* [11]. For B and C, the left image is the baseline output while the right image is with perspective preservation.

map ground-truths and only uses the prior knowledge of the room's Manhattan structure [28].

For the quantitative results, refer to Table 1 and Table 2, where we compute the Mean Squared Error (MSE) between the predicted depth maps of the content image and the stylized image. We compared our method to Gatys *et al.* [5] since it is the backbone style transfer method we used, and it is a good benchmark on how the improvement we made to the algorithm. We also compared our method to a style transfer method [9] that uses depth information. We used two datasets, LSUN [29] and NYUv2 [33], as the source of the content images. For [29] we used the 456 bedroom test images, while for [33] we selected 200 images, which satisfies the Manhattan assumption [28], as our content images.

The general trend for the results in Table 1 and Table 2 is that our method significantly improves upon [5] in terms of depth map fidelity. Compared with [9], our method has similar results, but our approach does not need depth map ground-truths. For more stylized images, refer to Fig. 11.

D. STYLIZATION QUALITY COMPARISON

A good perspective-preservation ability does not indicate that the method is a good performing style transfer. To measure our style transfer performance, we compute the content loss and style loss of the stylized image. Content loss tells how much the stylized image is similar to the content image. We compute content loss by using MSE between the *relu4_2* VGG [30] features of the two images. For the style loss, the computation is much more complicated. First, we get the features of the two images (content and stylized) using the VGG [30] layers: *relu1_1*, *relu2_1*, *relu3_1*, *relu4_1*, and *relu5_1*. Second, we compute the gram matrices of each layer's features. Finally, we compute the MSE between the corresponding layers and perform a weighted summation of the MSE of each layer. We also compute the depth loss which is the MSE between depth map of the stylized image and content image.

TABLE 4. Runtime and parameter count of the additional procedures introduced by our perspective preserving style transfer method. The processes are run in the CPU.

	Runtime (sec)	Parameters
Perspective Removal	1.70	85408411
Perspective Restoration	2.27	N/A
Total	3.97	85408411

Our method can utilize any style transfer method as its backbone. In this section, other than Gatys *et al.* [5], we used Huang *et al.* [11] to compute our metrics, as seen in Table 10. We compared the effects of our method in terms of content, style, and depth. In the Table, for each criterion (e.g. content), there are two sub-columns: baseline and with PP. For *baseline*, it means that the output is the result of using just the style transfer method. For *with PP*, it means that our perspective-preserving (PP) method is applied to the corresponding style transfer method. Based on the trends found in the Table, in general, our method improves the content loss, the style loss, and the depth loss of the baseline style transfer technique. In Fig 10, we could see the sample outputs of Gatys *et al.* [5], Huang and Belongie [11], and Liu *et al.* [9]. For (B) and (C) the left image is the baseline output, while the right image is their output with our perspective preservation method.

E. COMPUTATIONAL COMPLEXITY

Our method could improve the perspective preservation of other style transfer methods but it introduces additional procedures: Perspective Removal and Perspective Restoration. The overhead runtime and parameters can be seen in Table 4. The runtime profiling done using only the CPU with Intel i7-8700 processor. In the Perspective Removal module, we used a pretrained network [18] to find the keypoints of each major surfaces of a room. The keypoint detection module has a lot of parameters and it greatly increases the memory consumption of our method. An alternative network could be used by future research. For Perspective Restoration, it has longer runtime than Perspective Removal even if it has no parameters. The bottleneck of this process is the copying of values from the perspective-neutral stylized image to their respective mapping in the original content image. We also

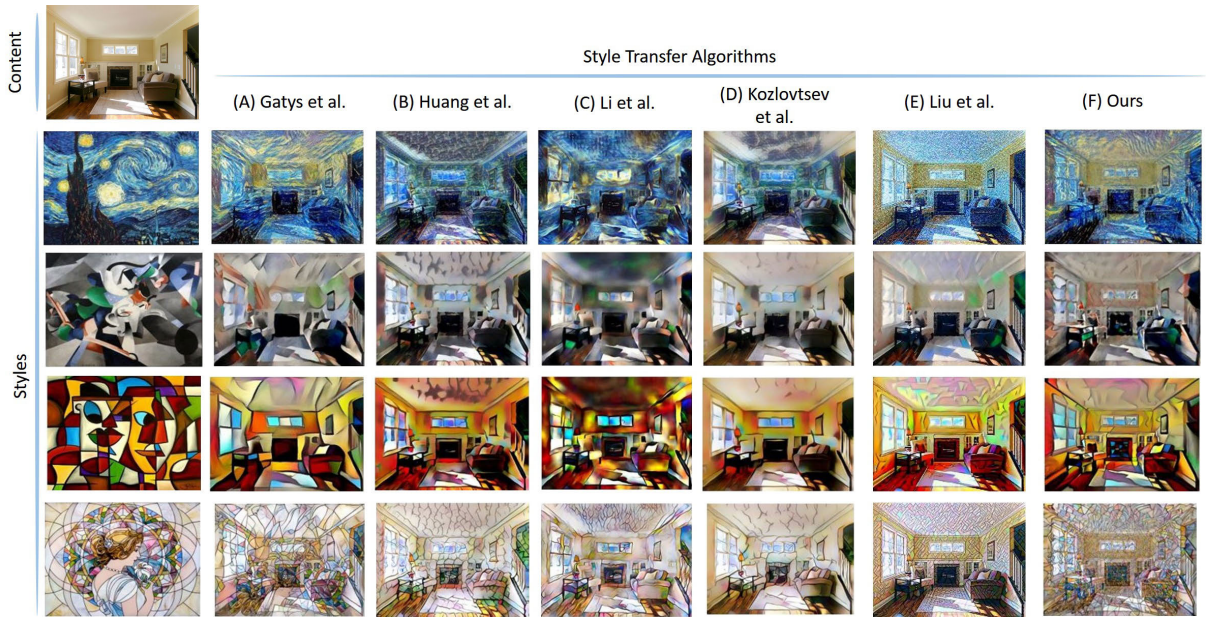

FIGURE 11. **Comparison of different style transfer methods on an interior portrait. From left to right stylized images: (A) Gatys *et al.* [5], (B) Huang *et al.* [11], (C) Li *et al.* [35], (D) Kozlovtsev et al [14], (E) Liu *et al.* [9], and (F) Ours.**

measured the runtime of the baseline style transfer method we used. For Gatys *et al.* [5] it is 66.21 seconds while for Huang and Belongie [11] it is 0.87 seconds.

V. CONCLUSION

Most style transfer algorithms are not able to capture perspective in the stylization of images; hence, we created a framework that preserves the perspective of interior portraits in style transfer. Our method works by, first, removing perspective from the image to obtain a perspective-neutral view rendering of the image. We perform style transfer to the neutral image then restoring the perspective to the image. This ensures that the perspective is agnostic to the style transfer algorithm. Our perspective removal method works by finding the major surfaces of a room interior, which are walls, ceilings, and floor. We warp these surfaces to obtain rectangular front-view renderings using four-point transforms. The next steps are style transfer, then reverse warping. Our approach yields stylized images with style elements warping along with the perspective of an interior surface. We performed evaluations considering the artistic elements that give an artwork its illusion of depth. Our method improves its baseline style transfer algorithm such that the depth information from the stylized image can be retrieved with an average MSE of 142.05, while the baseline has MSE of 263.05. Despite not using any depth preserving technique, we were able to yield a depth map that resembles that of the original interior portrait when compared to other methods.

ACKNOWLEDGMENT

(Wen-Yin Chen, Jose Jaena Mari Ople, and Maynard John Si are co-first authors.)

REFERENCES

[1] Facebook. *Instagram: A Simple, Fun & Creative Way to Capture, Edit & Share Photos, Videos & Messages With Friends & Family*. [Online]. Available: https://www.instagram.com/

[2] Snap. *Snapchat: Snapchat Lets You Easily Talk With Friends, View Live Stories From Around the World, and Explore News in Discover. Life's More Fun When You Live in the Moment!* Accessed: Jan. 3, 2020. [Online]. Available: https://www.snapchat.com/

[3] J. A. Paradiso and J. A. Landay, "Guest Editors' introduction: Cross-reality environments," *IEEE Pervas. Comput.*, vol. 8, no. 3, pp. 14–15, Jul. 2009.

[4] R. Poplin and A. Prins. *Behind the Scenes With Stadia's Style Transfer ML*. [Online]. Available: https://stadia.dev/blog/behind-the-scenes-with-stadias-style-transfer-ml/

[5] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[6] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2016, pp. 694–711.

[7] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3985–3993.

[8] X. Wang, G. Oxholm, D. Zhang, and Y.-F. Wang, "Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5239–5247.

[9] X.-C. Liu, M.-M. Cheng, Y.-K. Lai, and P. L. Rosin, "Depth-aware neural style transfer," in *Proc. Symp. Non-Photorealistic Animation Rendering (NPAR)*. New York, NY, USA: ACM, 2017, pp. 4:1–4:10.

[10] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 386–396.

[11] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.

[12] Y. Jing, Y. Liu, Y. Yang, Z. Feng, Y. Yu, D. Tao, and M. Song, "Stroke controllable fast style transfer with adaptive receptive fields," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 238–254.

[13] Y. Yao, J. Ren, X. Xie, W. Liu, Y.-J. Liu, and J. Wang, "Attention-aware multi-stroke style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1467–1475.

[14] V. Kitov, K. Kozlovtsev, and M. Mishustina, "Depth-aware arbitrary style transfer using instance normalization," 2019, *arXiv:1906.01123*. [Online]. Available: http://arxiv.org/abs/1906.01123
[15] J. D'Amelio, *Perspective Drawing Handbook*. Chelmsford, MA, USA: Courier Corporation, May 2004.
[16] K. Andersen, *The Geometry of an Art: The History of the Mathematical Theory of Perspective From Alberti to Monge*. Berlin, Germany: Springer, 2008.
[17] R. Liao, Y. Xia, and X. Zhang, "Depth-preserving style transfer," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 2016.
[18] C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich, "RoomNet: End-to-end room layout estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4865–4874.
[19] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1033–1038.
[20] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*. New York, NY, USA: ACM, 2001, pp. 327–340.
[21] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 795–802, Jul. 2005.
[22] H. Zhao, X. Jin, J. Shen, and F. Wei, "Real-time photo style transfer," in *Proc. 11th IEEE Int. Conf. Comput.-Aided Design Comput. Graph.*, Aug. 2009, pp. 140–145.
[23] W. Zhang, C. Cao, S. Chen, J. Liu, and X. Tang, "Style transfer via image component analysis," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1594–1601, Nov. 2013.
[24] Y. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand, "Style transfer for headshot portraits," *ACM Trans. Graph.*, vol. 33, no. 4, p. 148, 2014.
[25] S. Bruckner and M. E. Gröller, "Style transfer functions for illustrative volume rendering," in *Computer Graphics Forum*, vol. 26, no. 3. Hoboken, NJ, USA: Wiley, 2007, pp. 715–724.
[26] A. Selim, M. Elgharib, and L. Doyle, "Painting style transfer for head portraits using convolutional neural networks," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 129:1–129:18, Jul. 2016.
[27] J. Yaniv, Y. Newman, and A. Shamir, "The face of art: Landmark detection and geometric style in portraits," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 60:1–60:15, Jul. 2019.
[28] J. M. Coughlan and A. L. Yuille, "The manhattan world assumption: Regularities in scene statistics which enable Bayesian inference," in *Advances in Neural Information Processing Systems*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA, USA: MIT Press, 2001, pp. 845–851.
[29] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," 2015, *arXiv:1506.03365*. [Online]. Available: http://arxiv.org/abs/1506.03365
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
[33] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," 2013, *arXiv:1301.3572*. [Online]. Available: http://arxiv.org/abs/1301.3572
[34] W. Chen, Z. Fu, D. Yang, and J. Deng, "Single-image depth perception in the wild," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 730–738.
[35] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," Jan. 2017, *arXiv:1701.01036*. [Online]. Available: https://arxiv.org/abs/1701.01036

WEN-YIN CHEN received the M.A. degree in art and design from University for the Creative Arts, U.K. She is currently pursuing the Ph.D. degree with the Department of Arts and Design, National Taipei University of Education. Her research interests include 3D design, modeling, and digital image processing.

JOSE JAENA MARI OPLE received the B.S. degree in computer science from De La Salle University, Philippines, in 2018, and the M.S. degree from the National Taiwan University of Science and Technology (NTUST), in 2020, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering. His research interests include digital image processing and deep learning applied to computer vision.

MAYNARD JOHN SI received the B.S. degree in computer science from De La Salle University, Philippines, in 2018, and the M.S. degree from the National Taiwan University of Science and Technology, in 2020. His research interests include image processing and style transfer.

DANIEL STANLEY TAN received the M.S. degree in computer science from De La Salle University. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. His research interests include digital image processing and deep learning applied to computer vision.

KAI-LUNG HUA (Member, IEEE) received the B.S. degree in electrical engineering from National Tsinghua University, Hsinchu, Taiwan, in 2000, the M.S. degree in communication engineering from National Chiao Tung University, Hsinchu, in 2002, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA, in 2010. Since 2010, he has been with the National Taiwan University of Science and Technology, where he is currently a Professor with the Department of Computer Science and Information Engineering. Since 2019, he has also been the Vice Dean of the College of Electrical Engineering and Computer Science. He is the Director of the Artificial Intelligence Research Center. He is a member of Eta Kappa Nu and Phi Tau Phi. He was a recipient of the MediaTek Doctoral Fellowship. His current research interests include digital image and video processing, computer vision, and machine learning. He has received several research awards, including the 2019 Outstanding Research Award of Taiwan Tech, 2018 Young Scholar Award of Taiwan Tech, Top Performance Award of 2017 ACM Multimedia Grand Challenges, Top 10% Paper Award of 2015 IEEE International Workshop on Multimedia Signal Processing, the Second Award of the 2014 ACM Multimedia Grand Challenge, the Best Paper Award of the 2013 IEEE International Symposium on Consumer Electronics, and the Best Poster Paper Award of the 2012 International Conference on 3D Systems and Applications.