
Yunfei Liu, Feng Lu\textsuperscript{1,2*}

\textsuperscript{1}State Key Laboratory of VR Technology and Systems, School of CSE, Beihang University, Beijing, China
\textsuperscript{2}Peng Cheng Laboratory, Shenzhen, China
\{lyunfei, lufeng\}@buaa.edu.cn

Abstract

Many real world vision tasks, such as reflection removal from a transparent surface and intrinsic image decomposition, can be modeled as single image layer separation. However, this problem is highly ill-posed, requiring accurately aligned and hard to collect triplet data to train the CNN models. To address this problem, this paper proposes an unsupervised method that requires no ground truth data triplet in training. At the core of the method are two assumptions about data distributions in the latent spaces of different layers, based on which a novel unsupervised layer separation pipeline can be derived. Then the method can be constructed based on the GANs framework with self-supervision and cycle consistency constraints, etc. Experimental results demonstrate its successfulness in outperforming existing unsupervised methods in both synthetic and real world tasks. The method also shows its ability to solve a more challenging multi-layer separation task.

1 Introduction

Real imaging process is influenced by different factors, e.g., shape, illumination, reflection and refraction, making the captured images highly complex. Such images, when used as input data to an AI system, may heavily affect the system performance on various vision-based tasks, such as object detection (Girshick 2015), makeup removal (Cao et al. 2019), rain drop removal (Hao et al. 2019) and 3D reconstruction (Tatarchenko et al. 2019). As a result, separating different visual attributes from a single image has been considered an important research goal for decades.

This paper considers visual attributes separation via Single Image Layer Separation (SILS). It aims at decomposing an image into several independent layers with specific physical meanings. In general, this problem can be formulated as:

\[ I = L_1 \oplus L_2 \oplus \cdots, \]

where \( L_i \) is the \( i \)-th layer and \( I \) is the input image that can be reconstructed by a pixel-wise blend of all the layers. Plenty of real world problems can be represented in this form. As shown in Fig. 1, for reflection separation, the input image \( I \) can be decomposed into a reflection layer \( L_R \) and a background layer \( L_B \) in the form of \( I = L_R + L_B \). While for intrinsic decomposition, the input image \( I \) can be represented by the pixel-wise product of an albedo image \( L_A \) and a shading image \( L_S \), or alternatively formulated as \( \log(I) = \log(L_A) + \log(L_S) \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example.png}
\caption{Examples of single image layer separation: reflection separation and intrinsic image decomposition.}
\end{figure}

Despite its wide applicability, single image layer separation has been found fundamentally ill-posed, due to the existence of countless feasible solutions. To constrain the solution space, conventional methods use different priors such as low rank (Han and Sim 2017), sparsity (Li and Brown 2013) and relative smoothness (Li and Brown 2014) in optimization. Such hand-crafted constraints may work well in specific cases but fail in more complex cases. Recently, the deep convolutional neural network (CNN) has been widely used to handle similar problems (Fan et al. 2018; Yang et al. 2018; Zhang, Ng, and Chen 2018). However, CNN-based methods face major difficulties in high-quality data acquisition: 1) it is difficult or even impossible to obtain the ground truth of each layer for real images, and 2) different datasets vary greatly and cannot be used jointly. Since the performance of CNNs relies heavily on the training data, existing methods have to either use synthetic data with limited reality or train their networks with imperfect data.

In order to overcome such difficulties, this paper suggests \textit{not requiring the ground truth data triplet}, i.e., one input image and its exact two layers as training data, but simply collecting images from the three domains independently. In this manner, sufficient training data can be easily obtained without the acquisition problem. The absence of ground truth triplet, however, makes it an \textit{unsupervised learning problem}. Therefore, we propose an unsupervised single image layer separation (unsupervised SILS) method in this paper.
The proposed unsupervised SILS method does not rely on training data triplets from the three domains. Instead, we introduce two key assumptions, namely the shared information consistency and the layer independence, to help link the three different domains and construct the latent spaces. Based on the assumptions, we design the method by using techniques such as self-supervision, cycle consistency and GANs. The full method is able to perform unsupervised SILS without requiring ground truth triplets in training.

We test our method on two classical tasks, i.e., single image reflection separation and intrinsic image decomposition, and it outperforms state-of-the-art unsupervised methods. As our method does not need ground truth triplets, it can be quickly applied to other layer separation tasks with little data collection burden. Finally, our method can be extended to solve the more challenging multi-layer separation task.

1.1 Related Works

The problem of image layer separation can be considered as a special case of image domain translation. Previous domain translation can be formulated as a mapping function between a source and a target domain (Radford, Metz, and Chintala 2016; Yi et al. 2017; Zhu et al. 2017; Liu, Breuel, and Kautz 2017), while single image separation is a mapping from the source domain to two other domains physically related to the source domain. There are various image separation problems in the field of computer vision where different physical models are applied as priors for separation tasks. For instance, relative smoothness (Li and Brown 2014), ghost cues (Shih et al. 2015) and layer independence priors (Li and Brown 2013) are introduced for separating and removing the reflection on the glass surface from the background. Bi et al (Bi, Han, and Yu 2015) applied sparse prior to tackle image intrinsic decomposition. However, the above mentioned methods can only handle simple cases with image gradient or color changes, and can not be adopted to more complex situations that may require other techniques such as photometric stereo (Lu et al. 2015; 2017).

In recent years, supervised deep learning methods are applied to many image separation tasks. For example, fully convolutional networks with various guidance, e.g., image gradient information (Fan et al. 2017), face structure priors (Wan et al. 2019), feature discriminative prior (Wang and Lu 2019) and perceptual losses (Zhang, Ng, and Chen 2018), are used for single image reflection removal. Besides, Ronneberger et al. used U-net-like CNNs for intrinsic image decomposition (Ronneberger, Fischer, and Brox 2015).

Since real shot images with layer separation results are hard to collect, the development of unsupervised algorithms is in great need. Michael et al (UH 2017) proposed a self-supervised intrinsic image decomposition method. It can be trained with only a few images with ground-truth, and then transferred to other unpaired images. However, the albedo layer should be the same among different training images. Li et al (Li and Snavely 2018) and Lettry et al (Lettry, Vanhoey, and Van Gool 2018) proposed unsupervised intrinsic image decomposition methods, but these methods need multiple inputs with the same albedo layer for training. More recently, Hoshen et al (Hoshen) proposed an unsupervised single-channel blind source separation method, while the method requires a pair of input images which limits its applicability. Overall, unsupervised single image layer separation remains an open problem to solve.

2 Problem Formulation and Analysis

Single Image Layer Separation can be considered as a one-to-two image domain translation problem. In particular, let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ denote three image domains, then we try to find out a proper separation $x = y \oplus z$. In the supervised case, we have training samples $(x, y, z)$ drawn from a joint distribution $P_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}}(x, y, z)$. While in the unsupervised case, we only have samples drawn from the marginal distributions $P_{\mathcal{X}}(x)$, $P_{\mathcal{Y}}(y)$ and $P_{\mathcal{Z}}(z)$. Since we can infer nothing about the joint distribution only from the marginal samples without additional assumptions, the unsupervised problem is fundamentally ill-posed, as mentioned in Sec. 1. To address this problem, we introduce the following assumptions:

Assumption A: Shared information consistency. Given that $x = y \oplus z$, we assume that there exists a shared latent space $\mathcal{F}_y$ for $x$ and $y$, i.e., $\forall x, y, E_y(x) \in \mathcal{F}_y$ and $E_y(y) \in \mathcal{F}_y$, where $E_y$ is a function that maps data from the color image space to the latent space $\mathcal{F}_y$. Similarly, we have $E_z(x)$ and $E_z(z)$ both belong to the same latent space $\mathcal{F}_z$.

Following this assumption (also see Fig. 2 (a)), the original image layer separation pipeline can be reformed as:

$$ x \rightarrow \{y, z\} \Rightarrow x \rightarrow \underbrace{f_y \xrightarrow{G_y} y}_{f_z \xrightarrow{G_z} z} , \quad (2) $$

where $f_y = E_y(x) \in \mathcal{F}_y$ and $f_z = E_z(x) \in \mathcal{F}_z$. Such an $f_y$ can be understood as the extracted information, which only belongs to the domain $\mathcal{Y}$, from the blended input $x$. Then, $G_y$ is another mapping function to project $f_y$ back to the color image space and generate the layer image $y$.

In conclusion, this assumption allows producing $y/z$ from a single $x$ if we find out correct $E_y/E_z$ and $G_y/G_z$.

Assumption B: Layer independence. We assume that the separated $y$ and $z$ should be independent to each other in the latent space, i.e., samples belonging to the same domain have more similar features in the latent space.

Computationally, for any two samples $f^a_y, f^b_y$ from $\mathcal{F}_y$, and any other two samples $f^a_z, f^b_z$ from $\mathcal{F}_z$, we assume:

$$ d_\phi \gg d_\phi, \forall d_\phi \in \{|f^a_y - f^a_z|, |f^a_y - f^b_z|, |f^b_y - f^a_z|, |f^b_y - f^b_z|\}, \quad |f^b_y - f^b_z| \text{ and } \forall d_\phi \in \{|f^b_y - f^b_z|, |f^b_y - f^b_z|\}, \quad (3) $$

where $d_\phi$ indicates an inter latent space distance and $d_\phi$ indicates an inner latent space distance. This assumption can be satisfied by minimizing $d_\phi$ while maximizing $d_\phi$.

3 Learning for Unsupervised SILS

Based on the two assumptions in Sec. 2, we propose an unsupervised single image layer separation method. In particular, our method implements self-supervision, cycle consistency,
GANs, etc. to achieve efficient unsupervised learning. The overall framework of our method can be seen in Fig. 2 (b).

Self Supervision (SS). According to assumption A, $x$ contains all the information to produce $f_y$ and $f_z$, respectively. Meanwhile, assumption B requires that $f_y$ and $f_z$ should be as different as possible. Therefore, we need to carefully design the encoders $E_Y(x)$ and $E_Z(x)$ so that their outputs have maximized distances.

To this end, we propose a novel self-supervision framework to optimize both encoders and constrain the feature distributions in both latent spaces. In particular, we maximize the distance between output features from $E_Y$ and $E_Z$; for $E_Y$, we minimize the $L_1$ distance between $E_Y(x_i)$ and $E_Y(y_i)$, where $x_i$ and $y_i$ are unpaired samples randomly selected from $X$ and $Y$; for $E_Z$, we minimize the $L_1$ distance between $E_Z(x_i)$ and $E_Z(z_i)$, where $x_i$ and $z_i$ are unpaired samples randomly selected from $X$ and $Z$.

Cycle Consistency (CC). Assumption A describes how to produce $y$ and $z$ from the input $x$, while there should be one more step to close the loop: that $y$ and $z$ should re-produce $x$. This is physically obvious and leads to the cycle consistency.

As shown in Fig. 2 (b), let $y'$ and $z'$ be the output of $G_Y$ and $G_Z$, then we can simply fuse them again to produce $x' = y' \oplus z'$. Here we introduce the first cycle consistency constraint that enforces $x = x'$. Moreover, the new $x'$ can be further separated into new $y''$ and $z''$, resulting the second cycle consistency constraint of $y'' = y'$ and $z'' = z'$. In this manner, the physical correctness of the image layer separation & fusion processes is guaranteed.

GANs. Assumption A and B mainly focus on the separability of different layers, while there is no guarantee that the separated images look realistic. Therefore, we adopt the widely successful Generative Adversarial Networks (GANs) in our framework. The aim is to ensure the output $y$ and $z$ show realistic visual appearances as real images in their respective domains $Y$ and $Z$.

The GANs structure comprises generators and discriminators. For the former, we have $E_Y \circ G_Y$ and $E_Z \circ G_Z$ work as the generator to produce layer images $y$ and $z$, as shown in Fig. 2 (b). As for the latter, we add two discriminators $D_Y$ and $D_Z$. They are used to distinguish the outputs of the generators from the real images in domains $Y$ and $Z$.

Loss Function Design. According to the techniques introduced above, we propose the detailed loss functions as following. The overall loss function is designed as:

$$
\min_{E,G} \max_D \mathcal{L}_{SS}(E_Y, E_Z) + \mathcal{L}_{GAN_y}(E_Y, G_Y, D_Y) + \mathcal{L}_{GAN_z}(E_Z, G_Z, D_Z) + \mathcal{L}_{CC_y}(E_Y, G_Y, E_Z, G_Z) + \mathcal{L}_{CC_z}(E_Y, G_Y, E_Z, G_Z).
$$

The first part is related to the self-supervised training. Based on Equ. (3), the objective function can be designed as:

$$
\mathcal{L}_{SS}(E_Y, E_Z) = \lambda_1 d_{\phi}(E_Y(y), E_Y(x)) + \lambda_2 d_{\psi}(E_Z(z), E_Z(x)) + \lambda_3 (1 - d_{\psi}(E_Y(x), E_Z(x)));
$$

where the hyper-parameters $\lambda_1$, $\lambda_2$ and $\lambda_3$ control the weights of different objective terms. We adopt $d_{\phi}(a, b) = |a - b|$ in practice, i.e., we compute the $L_1$ distance of two inputs $a, b$. Besides, $d_{\psi}$ measures the distance between the latent codes.
which are the outputs of $E_Y$ and $E_Z$, and we use a modified sigmoid function to compute the distance:

$$d_{ψ}(a, b) = \frac{1}{1+e^{g(a,b)}}, \quad \text{where } g(a, b) = -\frac{|a - b| - αe^a}{α^2}.$$  (6)

Here, $α$ controls the shape of the distance curve. Its effect is shown by the experimental results in the top of Fig. 3 (b).

The GANs-related objective functions are as below:

$$L_{GAN_Y}(E_Y, G_Y, D_Y) = λ_0\mathbb{E}_{y\sim P_Y}[\log D_Y(y)] + λ_0\mathbb{E}_{f_x\sim E_Y(f_u|x)}[\log(1 - D_Y(G_Y(f_y)))] + \lambda_0\mathbb{E}_{f_z\sim E_Z(f_u|x)}[\log(1 - D_Z(G_Z(f_z)))]$$

$$L_{GAN_z}(E_Z, G_Z, D_Z) = λ_0\mathbb{E}_{z\sim P_Z}[\log D_Z(z)] + \lambda_0\mathbb{E}_{f_z\sim E_Z(f_u|x)}[\log(1 - D_Z(G_Z(f_z)))]$$  (7)

The above functions are in the form of conditional GANs objective functions. The only hyper-parameter $λ_0$ controls the impact of the entire GAN objective functions.

Finally, for the cycle-consistency constraint, the loss functions are based on the $L_1$ differences:

$$L_{CC_Y} = λ_4|y' + z' - x|,$$

$$L_{CC_z} = λ_5|G_Y(E_Y(y' + z')) - y'|,$$

$$L_{CC_z} = λ_6|G_Z(E_Z(z' + z')) - z'|.$$  (8)

where $y' = G_Y(E_Y(x))$ and $z' = G_Z(E_Z(x))$. The hyper-parameters $λ_4, λ_5$ and $λ_6$ control the weights of these terms.

## 4 Implementation via CNNs

Our unsupervised SILS network consists of a two-stream generator and two discriminators, as shown in Fig. 2 (b). The generator is composed of two encoder/decoder pairs, i.e., $E_Y \circ G_Y$ and $E_Z \circ G_Z$, to generate $y$ and $z$ from a single $x$. The discriminators determine whether the generated $y$ and $z$ look real enough. Below are the implementation details.

Due to the different tasks in our experiments, we design two types of generators in practice.

**Generator for synthetic data.** The synthetic data is rendered by using simple square and circle shapes, as shown in Fig. 3 (a). Therefore, the generator is designed to be relatively simple. The structures of $E_Y \circ G_Y$ and $E_Z \circ G_Z$ are the same, but they do not share parameters. In detail, each of the encoders $E_Y$ and $E_Z$ has five convolutional layers, which have $\{16, 32, 64, 128, 256\}$ filters of the size $4 \times 4$ and a stride of 2. Batch normalization (Ioffe and Szegedy 2015) and leaky ReLU activation are applied after every convolutional layer. Each of the decoders $G_Y$ and $G_Z$ has the same layers as the encoder but in a reverse order plus a final layer with 3 channels. The corresponding layers of the encoder and decoder have mirror-link connections as described in (Ronneberger, Fischer, and Brox 2015), yielding sharper results.

**Generator for real image separation tasks.** In order to handle real world tasks, we propose a more complex generator as an alternative. Overall, $E_Y \circ G_Y$ and $E_Z \circ G_Z$ still share the same structure, but not the parameters. In detail, the encoders $E_Y$ and $E_Z$ are implemented by using VGG-19 (Simonyan and Zisserman 2015). The decoders $G_Y$ and $G_Z$ have 4 convolution blocks in each, and each block comprises a convolution layer, a leaky ReLU activation layer and an up-sample layer. The last block is followed by a fully convolutional network with 64 filters of size $3 \times 3$, a stride of 1 and dilation rations of $\{2, 4, 8, 16, 32, 1\}$. Instance normalization (Dmitry Ulyanov and Lempitsky 2016) and leaky ReLU activation are applied after each convolution layer. The output layer is a convolution layer with three $1 \times 1$ filters.

For the skip connection between the encoder and decoder, we select ‘conv1\_2’, ‘conv2\_2’ and ‘conv3\_2’ layers in the encoder and make skip connections to the first block of the decoder. This strategy has been shown efficient for image synthesis and enhancement (Zhang, Ng, and Chen 2018).

**Discriminator.** We implement $D_Y$ and $D_Z$ as the discriminators. They share the same multi-branches structures as in (Liu, Breuel, and Kautz 2017), but do not share parameters. For the $i$-th branch in $D_Y$ or $D_Z$, the input image is down-sampled by $i - 1$ times via average pooling and then fed to the network. Each branch has four convolution layers with $\{32, 64, 128, 32\}$ filters of size $4 \times 4$ and a stride of 2. In the end, the output features from different branches are fused together followed by a sigmoid activation. For the synthetic data case, the number of branches is simply 1, while for the real image separation tasks, the number of branches is set to 3.

**Training.** We use ADAM (Kingma and Ba 2015) optimizer with a learning rate of 0.0001 and momentums of 0.9 and 0.9. The default values of hyper-parameters in Equ. (5)-(8) are set to $λ_0 = 5.0, λ_1 = 0.5, λ_2 = 0.5, λ_3 = 1.0, λ_4 = 1.0, λ_5 = 1.0, λ_6 = 1.0$ and $α = 1.4$.

Each mini-batch in training contains one input image from the domain $X$, one layer image from the domain $Y$ and another one from domain $Z$. Note that these three images do not form an image separation triplet. For the real world tasks, data argumentation operations are applied, including random scalings by factors within $0.8 \sim 1.2$, random croppings and random horizontal flippings.

Inheriting from GANs, the training solves a min-max problem to find a saddle point in the solution space. To make it stable, we use a gradient update strategy similar to (Liu, Breuel, and Kautz 2017) with gradient penalization to solve Equ. (4). Specifically, a gradient descent step first updates $E_Y, E_Z, G_Y$ and $G_Z$ with fixed $D_Y$ and $D_Z$, then another gradient descent step updates $D_Y$ and $D_Z$ with fixed $E_Y, E_Z, G_Y$ and $G_Z$.

## 5 Experiments

Experiments are conducted on the synthetic data and also real world single image separation tasks. The former aims at providing with intuitive and quantitative results, and the latter demonstrates the real world applicability of our method. Moreover, our method is extended to solve a more challenging task that involves multiple layers.

### 5.1 Datasets

We prepare three types of dataset: 1) synthetic dataset, 2) real world intrinsic image decomposition dataset and 3) real world reflection removal dataset. Note that in any case, we do not have the data triplet from the three domains, i.e., input image with its two layers, in the training set.
We conduct experiments with synthetic data to analyse our method in detail. Since the self-supervision (SS) is a major technical component of our method, we study its performance and optimal parameter settings.

First, we analyze the distance loss in Equ. (6). Results are shown in Fig. 3 (b), where loss = 1 − \(d_0\) and distance indicates \(|a−b|\). The curves show that the loss values, ranging from 0 to 1, indeed decline with increasing distance values but with different shapes. According to the curve shapes, we set \(\alpha = 1.4\) empirically in the follow experiments.

We then study the balance between the intra-domain distance and inter-domain distance in SS. In Equ. (5), \(\lambda_1, \lambda_2\) control the intra-domain distance loss while \(\lambda_3\) controls the inter-domain distance loss. Therefore, we tune these hyper parameters and re-train the model without touching other settings. The resulting mean square errors (MSEs) are shown in the bottom of Fig. 3 (b), where \((\lambda_1 + \lambda_2)/\lambda_3 = 0.1\) produces the worst model while \((\lambda_1 + \lambda_2)/\lambda_3 = 1\) gives the best one. This proves that a good balance should consider both intra-domain and inter-domain distances of the feature, which confirms our Assumption B. In practice, we set \(\lambda_1 = \lambda_2 = 0.5\) and \(\lambda_3 = 1\) in the following experiment.

Next, we study the impacts of our SS and CC constraints by conducting an ablation study. Results are reported in Fig. 3 (c), showing that removing any of these constraints will increase the final error drastically, i.e., twice as large or even more. This persuasively proves that both the SS and CC are essential to our method.

For more information, we also show the results of CycleGAN (Zhu et al. 2017) and UNIT (Liu, Breuel, and Kautz 2017) on our synthetic dataset. Their MSEs are 0.67/0.61 and 0.46/0.37, which are obviously larger than ours (0.289/0.274). This demonstrates the advantage of our method.

Finally, we consider the overall loss function in Equ. (4). We introduce two temporary weights \(w_1\) and \(w_2\) to balance each term, and the resulting loss function becomes

\[
\min_{E,G} \max_D w_1L_{SS} + L_{GAN} + w_2L_{CC}. \tag{9}
\]

We conduct experiments with different \(w_1\) and \(w_2\) and the results are in Fig. 4, where we find that \(w_1 = 1, w_2 = 10\) gives the best result.
5.3 Evaluation on Real Image Separation Tasks

Intrinsic image decomposition. We compare the proposed method with the most representative unsupervised methods, i.e., LM14 (Li and Brown 2014), CycleGAN (Zhu et al. 2017), UNIT (Liu, Breuel, and Kautz 2017) and LK18 (Lettry, Vanhoey, and Van Gool 2018). They are trained on the same dataset as ours. Besides, we also show results of fully-supervised methods MSCR (Narihira, Maire, and Yu 2015) and FY18 (Fan et al. 2018) using their pre-trained models. Note that these fully-supervised methods are trained on ground truth data triplets and thus their results can only serve as a reference.

Visual comparison among all the methods is illustrated in Fig. 5. CycleGAN produces the worst images with bad textures and wrong colors. Its training may fail with a high probability. LK18 and LM14 perform better than CycleGAN while their visual qualities are still unsatisfactory. UNIT produces relatively good albedo layers. However, it cannot output the shading layer due to its image-to-image translation framework. On the other hand, our method generates both the albedo and shading layers with a good quality and they appear highly identical to the ground truth images. The visual quality of our results is even better than the two fully supervised methods, i.e., MSCR and FY18, in some cases.

We also provide a qualitative comparison in Table 1, where LMSE is an error metric specifically designed for intrinsic image data introduced by (Grosse et al. 2009). Numerical results show that our method achieves state-of-the-art performance among these unsupervised methods. Its accuracy is even comparable to the fully supervised methods that rely on exact ground truth data triplets in training.

In this experiment, we also study our feature extraction results. Fig. 7 visualizes the distribution of the encoded features by using their 2D PCA vectors. It is clear that the encoder $E_\gamma$ projects $x$ and $y$ into the same cluster (indicating the same latent space for albedo layer), and the encoder $E_\zeta$ projects $x$ and $z$ into another cluster (latent space for shading layer). The two clusters stay apart from each other with a large distance. This verifies the correctness of our Assumption A and B and thus confirms the design of our method.

Reflection removal. Similar to the intrinsic image decomposition case, we compare our method with the unsupervised methods LM14, CycleGAN and UNIT. They are trained on the same dataset. For the fully-supervised methods, we use the pre-trained models of ZN18 (Zhang, Ng, and Chen 2018) and BDN (Yang et al. 2018) on the same dataset but with ground truth triplet data. Fig. 6 gives the visual comparison among unsupervised methods and ours. LM14 can separate both background and reflection layers from the input image, but the separation is clearly not ideal. Both the CycleGAN and UNIT only recover the background layer, where there still exist artifacts or remaining reflections in many image

![Figure 5: Qualitative comparisons on MIT dataset. From left to right: MSCR and FY18 are fully supervised method shown for reference. LM14, CycleGAN and UINT are unsupervised methods than can be directly compared with ours. Our results have a good visual quality and appear almost identical to the ground truth.](image-url)

<table>
<thead>
<tr>
<th>Method</th>
<th>Albedo MSE</th>
<th>Shading MSE</th>
<th>Avg. MSE</th>
<th>Total MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM14</td>
<td>0.0286</td>
<td>0.0227</td>
<td>0.0255</td>
<td>0.0366</td>
</tr>
<tr>
<td>CycleGAN</td>
<td>0.0272</td>
<td>N/A</td>
<td>0.0272</td>
<td>0.407</td>
</tr>
<tr>
<td>UNIT</td>
<td>0.0207</td>
<td>N/A</td>
<td>0.0207</td>
<td>0.0310</td>
</tr>
<tr>
<td>LK18</td>
<td>0.0232</td>
<td>0.0166</td>
<td>0.0197</td>
<td>0.0379</td>
</tr>
<tr>
<td>Ours</td>
<td>0.0167</td>
<td>0.0140</td>
<td>0.0154</td>
<td>0.025</td>
</tr>
<tr>
<td>MSCR</td>
<td>0.0207</td>
<td>0.0124</td>
<td>0.0165</td>
<td>0.024</td>
</tr>
<tr>
<td>FY18</td>
<td>0.0127</td>
<td>0.0085</td>
<td>0.0106</td>
<td>0.020</td>
</tr>
</tbody>
</table>

Table 1: Numerical comparison on MIT intrinsic dataset. Note that MSCR and FY18 are fully-supervised methods, which are trained on ground truth data triplets, and thus their results can only serve as a reference.
regions. Overall, our method achieves the best background cleanliness and the visual quality of the reflection layer.

To make a qualitative comparison with state-of-the-art methods, we use peak-signal-to-noise-ratio (PSNR) (Huynh-Thu and Ghanbari 2008) and structural similarity index (SSIM) (Wang et al. 2004) as evaluation metrics. Table 2 shows that our method achieves the state-of-the-art performance against other unsupervised methods, and is also superior to the fully-supervised method BDN (Yang et al. 2018).

### 5.4 Evaluation on Multi-layer Separation

In this section, we extend our evaluation to the multi-layer case. We re-render the synthetic dataset with three different shapes, i.e., square, circle and triangle, as shown in Fig. 8 (a). Totally 5000 images are synthesized without containing any triplet data, and other settings remain the same as described in Sec. 5.2. Accordingly, we modify our method by adding another network branch of a generator and a discriminator to handle the third layer. The example results are shown in Fig. 8 (b), which demonstrate the successfulness of our method to handle such a challenging problem.

![Figure 7: Visualization of our feature encoding distribution using the corresponding PCA vectors. We use t-SNE (Maaten and Hinton 2008) to aid the visualization of the latent space. Encoder \( E_y \) projects \( x \) and \( y \) into the same cluster (latent space for albedo layer), and encoder \( E_z \) projects \( x \) and \( z \) into another cluster (latent space for shading layer).](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Background SSIM</th>
<th>Background PSNR</th>
<th>Reflection SSIM</th>
<th>Reflection PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM14</td>
<td>0.715</td>
<td>17.00</td>
<td>0.357</td>
<td>17.58</td>
</tr>
<tr>
<td>CycleGAN</td>
<td>0.622</td>
<td>16.63</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>UNIT</td>
<td>0.738</td>
<td>19.23</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.821</strong></td>
<td><strong>20.47</strong></td>
<td><strong>0.425</strong></td>
<td><strong>19.31</strong></td>
</tr>
<tr>
<td>Sup. (Ref.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BDN</td>
<td>0.820</td>
<td>18.87</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>ZN18</td>
<td>0.858</td>
<td>22.22</td>
<td>0.420</td>
<td>18.94</td>
</tr>
</tbody>
</table>

Table 2: Comparison on reflection removal dataset. Note that BDN and ZN18 are fully-supervised methods, which are trained on ground truth data triplets, and thus their results can only serve as a reference.

![Figure 8: The extend experiment on three layers separation task. (a) Example of the blend image and each layer. (b) Two visual results obtained by the proposed method. From left to right: input image and separated layers.](image)

## 6 Conclusion

In this paper, we propose an unsupervised single image layer separation method. It is unsupervised since it requires no ground truth data triplet, i.e., input images and its layer images for training. To this end, we introduce two assumptions about the feature distributions of different layers, namely the shared information consistency and layer independence assumptions. Based on them, we design our method by using the GANs framework with self-supervision and cycle consistency constraints. Experimental results show that our method outperforms existing unsupervised methods in both synthetic and real world tasks. The method can also be extended to
solve a more challenging three-layer separation task.

References

Bi, S.; Han, X.; and Yu, Y. 2015. An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *TOG*.


Hoshen, Y. Towards unsupervised single-channel blind source separation using adversarial pair unmix-and-remix. In *ICASSP*. IEEE.


