ABSTRACT This paper addresses the problem of super-resolution of environment matting of transparent objects. In contrast to traditional methods of environment matting of transparent objects, which often require a large number of input images or complex camera setups, recent approaches using convolutional neural networks are more practical. In particular, after training, they can generate the environment mattes using a single image. However, they still do not have super-resolution capabilities. This paper first proposes an encoder-decoder network with restoration units for super-resolution environment matting, called Enhanced Transparent Object Matting Network (ETOM-Net). Then, we introduce a refinement phase to improve the details of the output further. The ETOM-Net effectively recovers lost features in the LR input images and produces visually plausible HR environment mattes and the corresponding reconstructed images, demonstrating our method’s effectiveness.

INDEX TERMS Super-resolution, environment matting, deep learning, convolutional neural network.

I. INTRODUCTION

Image matting has been used in many real-life applications, such as in image and video editing or in showing the weather map superimposed with the meteorologist commonly seen in our daily TV news. The matting process estimates an alpha matte that separates the foreground object from the background, so that the object can be placed on a new background, which is how the film industry creates special effects. The image matting model [1] is defined as follows:

$$C = \alpha F + (1-\alpha)B,$$

where $C$ denotes the composited pixel value, $F$ and $B$ denote, respectively, the foreground pixel value and the background pixel value. $\alpha$ denotes the opacity, indicating the degree of blending between the foreground and the background.

As Eq. 1 shows, it is only able to handle non-transparent objects because it does not take optical properties into account, such as refraction and reflection of transparent objects.

To address this limitation, Zongker et al. [23] introduce the environment matting to capture how light in the environment is refracted and reflected by the surface of the foreground object.

$$C = F + (1-\alpha)B + \Phi,$$

where $\Phi$ represents the contribution of light from the environment that is reflected or refracted by the surface of the foreground object.

After the work of Zongker et al. [23], many approaches [24–28] have been proposed to improve their method, but the proposed methods are still limited by the large number of input images or complex image capture settings. Inspired by the performance of convolutional neural networks in high-level computer vision tasks, Chen et al. [29] propose a CNN-based approach, called Transparent Object Matting Network (TOM-Net), which learns the environment matte from a single input image and is effective and efficient compared to earlier works.

Considering that the input images used are usually of low quality, combining super-resolution capabilities with environment matting would be a good combination. The process of reconstructing an HR image from a single LR image is called Single Image Super-Resolution (SISR). While there are a large number of off-the-shelf methods available, simply using an SISR method to super resolve an environment matte will not produce plausible results. In particular, for refractive flows, super-resolution cannot be performed with existing methods. Thus, in our work, we focus on super-resolution of environment matting of transparent objects from a single image.
In this paper, we propose a new network called ETOM-Net with three restoration units for super-resolution environment matting. The network effectively recovers lost features in LR input images and produces visually plausible HR environment matte and synthesized images as shown in Fig. 1. In addition to the main phase, we incorporate a refinement phase with residual learning to improve the quality of the HR environment matte and the reconstructed image.

II. RELATED WORK

A. SINGLE IMAGE SUPER-RESOLUTION

Super-resolution is the process of generating a high resolution image from a low resolution or degraded image. Super-resolution can be divided into two categories depending on how many images are used as input: SISR and Multi Image Super Resolution (MISR). SISR is challenging but is more practical in real-world applications, which is the focus of many recent researchers.

SISR can be categorized into four types: interpolation-based methods (e.g., bicubic interpolation), reconstruction-based methods [3], [4], [6], [9], example-based methods [2], [5], [7], [8], [10]–[12] and learning-based methods [13], [15], [17], [19], [21]. The focus of this paper is on deep learning solutions.

Dong et al. first introduce a neural network model into SISR, called SRCNN [13]. This method uses a three-layer CNN to perform patch extraction and representation, non-linear mapping and reconstruction. As a milestone, it has a lightweight structure that achieves not only speed for practical use but also better performance than the state-of-the-art conventional methods. FSRCNN [14] is an improvement of SRCNN. Instead of scaling up the LR input at the beginning as SRCNN does, it processes the LR image directly and applies a deconvolution layer at the end to scale the results to the correct size. It has a faster training speed but still maintains a good performance.

Later, ResNet [15] proposed by He et al. incorporates a residual learning framework to improve the training for very deep networks. Since the introduction of ResNet, researchers have explored the possibility of using residual learning in SISR. In VDSR [16], Kim et al. present a very deep convolutional network with global residual learning to achieve a significant improvement in accuracy. DRCN [18] proposed by the same group, first applies a deeply recursive neural network to SISR. Inspired by DRCN, DRRN [17] extends VDSR by introducing recursive blocks with residual units. Unlike VDSR, which uses only global residual learning, DRRN combines local and global residual learning, and VDSR can be considered a particular case of DRRN when residual units are not used.

Ledig et al. present SRResNet [19] with 16 blocks deep ResNet as part of the SRGAN, which is the first time that the Generative Adversarial Network (GAN) is applied to image super-resolution. Soon after, Lim et al. remove the Batch Normalization (BN) from SRResNet and propose EDSR [20], which can stack more residual blocks under the same condition. RCAN [21] uses a residual in residual structure to form a very deep network and a channel attention mechanism to adaptively rescale channel-wise features, which performs highly accurate SISR.

Later on, Cheng et al. combine an encoder-decoder network with a residual-in-residual structure, which includes several residual channel-wise attention blocks inspired by RCAN, named EDRN [22]. In EDRN, it adopts a coarse-to-fine structure, which can gradually recover the lost information and reduce the noise impact. They also use batch normalization in real SISR, which has been shown to be inefficient for SISR with synthetic datasets. The results show that applying BN to downsampling or upsampling convolutional layers yields a performance improvement without a significant increase in execution time. EDRN can effectively restore HR images from real-world LR images and is one of the best methods of NTIRE 2019 Real SR Challenge.

B. ENVIRONMENT MATTING

Environment matting first introduced by Zongker et al. [23] captures not only a foreground object and how the light is attenuated passing through it but also how the object refracts and reflects light from the scene. The foreground object can then be composited into a new environment with physically correct reflection and refraction effects from the environment. They use three monitors to display a series of magenta and green stripes and a digital camera to capture the scene so they obtain the environment matte by identifying background areas corresponding to each foreground pixel.

Chuang et al. [24] further extend the original environment matting in two distinct directions. The first is to utilize more backdrops to capture complex and subtle object refraction and reflection. The second is to obtain a simplified matte using only one image, which allows them to achieve a real-
time environment matting of colourless objects in motion.

Both methods assume that some region in the background maps to a foreground pixel in the image. However, Wexler et al. [25] believe that a probabilistic model-based approach, which assumes that each background pixel has a probability of contributing to the colour of some foreground pixel and does not require complex calibration setup, is a better choice. Their method has a limitation that diffuse scattering affects the estimation of probability densities.

Peers et al. [26] use a series of wavelet patterns to obtain the environment matte of a scene while capturing the effect of diffuse reflections, which is not possible in previous methods and reduce post-processing time by linearly combining a large number of basis images.

Zhu and Yang [27] introduce a frequency-based environment matting method. Instead of analyzing images in the time domain, their method uses Fourier analysis to analyze the data in the frequency domain, which can obtain a more physically correct result at the expense of requiring many images. Later, Qian et al. [28] incorporate compressive sensing theory to the frequency-based environment matting, resulting in higher performance but with much fewer number of images.

TOM-Net [29] is a CNN-based environment matting approach proposed by Chen et al. They design a deep learning framework to learn the mapping between a single input image and the corresponding environment matte, including an object segmentation mask, an attenuation map and a refractive flow field by assuming that the foreground object is transparent, has no colour, and has only one mapping at each point. They can then composite a new image using the output matte and a new backdrop. They also created a large-scale synthetic dataset and a real dataset for training and testing. Their approach is effective and efficient, requiring no cumbersome capture procedures and lengthy processing times, and it still yields visually pleasing results. Although Chen et al. have explored the potential of CNN-based environment matting, their method TOM-Net does not have the super-resolution capability that is practical in real-life situations where the input images are usually of low quality. Such a limitation motivates this research.

III. FORMULATION

A. SUPER-RESOLUTION

Low-resolution images can be seen as a degradation of high-resolution images. In general, HR images and LR images are linked by this model:

\[ I_{LR} = (I_{HR} \otimes k) \downarrow s + n, \]  

where \( \otimes \) represents the convolution operation with blur kernel \( k \), \( \downarrow s \) denotes the downsampling operation of the scale factor \( s \), and \( n \) denotes the additive noise. Since our main focus is on synthetic data, we assume that bicubic downsampling and Gaussian blur are used in our work to generate LR images from the ground truth.

B. ENVIRONMENT MATTING

Following the work of [24] and [29], we first assume that the foreground object is colourless and transparent because too many optical properties would make the model too complex to obtain good results from it.

For refraction, Wexler et al. [25] assume that each background pixel has a probability to contribute to some foreground pixel, and Zongker et al. [23] assume that each foreground pixel is a linear combination of pixel values of a region in the background. In our work, we assume that there is no reflection of the foreground object, and in the single background setting, each foreground pixel comes from a single pixel in the background.

With these assumptions, similar to [29], the transparent environment matting problem can be modelled as follows:

\[ O = (1 - I_{mask})I_{ref} + I_{mask}I_{rho} \cdot S(I_{ref}, G(I_{flow})), \]  

In this model, \( O \) denotes the composited image, \( I_{mask} \), \( I_{rho} \), \( I_{ref} \) denote the pixelwise mask of the foreground object, the attenuation map of the foreground object, and the background image, respectively. The mask \( I_{mask} \in \{0,1\} \) has two values, and \( I_{mask}(i,j) = 0 \) denotes that the pixel at \((i,j)\) is a background pixel and vice versa. The amount of attenuation map \( I_{rho} \in [0,1] \) indicates how much the object attenuates the light.

\( S() \) is a function that re-samples the image using the background image values and pixel locations from a flow-field grid, and the computation is done by bilinear interpolation. The grid specifies the normalized sampled pixel positions, with most values in the range of \([-1,1]\), and it is generated by the function \( G() \) using a two-channel refractive flow \( I_{flow} \) that represents an offset \((V_x, V_y)\) between the composited image and its corresponding background image.

The function \( G() \) is a flow-field grid generator that first generates a two-dimensional base grid that has values from the left to the right and from the top to the bottom from 0 to width and height, respectively. It then scales this base grid to \([-1,1]\) and adds the input refractive flow element-wise to this scaled base grid to form a flow-field grid as the input to \( S() \).

From Eq. 4, the environment matting problem can now be solved by estimating an environment matte which includes a pixelwise mask \( I_{mask} \), an attenuation map \( I_{rho} \), and a refractive flow field \( I_{flow} \) from a single input image, as shown in Fig. 1. Note that \( I_{rho} \) and \( I_{flow} \) only apply to the region where \( I_{mask} = 1 \), and outside of this region, we use the corresponding pixels in the background as the composited pixels. So the quality of \( I_{mask} \) has a significant influence on the reconstructed image.

IV. PROPOSED METHOD

A. ARCHITECTURE

The main phase of our proposed method ETOM-Net is shown in Fig. 2. Similar to [29] and [30], it contains an encoder-decoder structure with a shared encoding process and three
et Θ denotes the element-wise addition, ⊗ denotes the element-wise product, and Pooling denotes global average pooling.

independent decoding processes corresponding to the three output environment mattes.

In this structure, we use six encoders and eighteen decoders inspired by the work of [29]. Every three decoders form a combination that shares the same input, which allows the three decoding processes to learn features from each other, and so the three output environments mattes are more correlated.

Each encoder contains two convolutional layers with steps equal to 1 and 2, and two ReLU activation layers, forming a factor of 64 for downsampling. Each decoder has one convolutional layer, one ReLU activation layer and one upsampling layer that recovers the resolution downsampled by the encoder. Skip connections are also used to connect feature maps of the same size during the encoding and decoding processes. The encoding process can be represented as:

\[ O_{E,i} = E_i(O_{E,i-1}), \]  

and the output of each decoder can be formulated as:

\[ O_{D,i,j} = D_{i,j}(\sum_{j=0}^{2} O_{D,i-1,j} + O_E), \]

where \( O_{E,i} \) denotes the output of the i-th encoder, \( O_{D,i,j} \) denotes the output of the decoder \((i, j)\), i.e., the j-th decoding process of scale i. \( E_i \) denotes the i-th encoder, \( D_{i,j} \) denotes the decoder \((i, j)\). \( O_E \) denotes the output of the encoding process of the same feature dimension as \( O_{D,i-1,j} \).

We add three new Restoration Units (RU) after the encoding process and before the decoding processes, which allows the main phase of the network to focus on more informative parts of the LR input and also to enhance the discriminative power of the network. Fig. 3 shows the structure of the restoration unit. Each RU consists of four Residual In Residual Blocks (RIRB), which is inspired by the work of [22], and a convolutional layer, with each RIRB stacked with ten Residual Blocks (RB) and one convolutional layer. The output of the RU can be formulated as:

\[ O_{RU} = Conv(RIRB_3(\cdots(RIRB_0(I_{RU})))) + I_{RU}, \]

where \( O_{RU} \) denotes the output of RU, and \( I_{RU} \) denotes the input of RU. \( Conv \) and \( RIRB_i \) denote a convolutional layer and the i-th RIRB block, respectively. And the output of \( RIRB_i \) can be obtained by:

\[ O_{RIRB,i} = Conv(RB_3(\cdots(RB_0(I_{RIRB,i})))) + I_{RIRB,i}, \]

where \( O_{RIRB,i} \) denotes the output of the i-th RIRB, \( I_{RIRB,i} \) denotes the input of the i-th RIRB and \( RB_i \) denotes the i-th RB block.

Within each RB, we utilize two convolutional layers, with a ReLU activation layer between them and a Residual Channel-wise Attention Block (RCAB) [21] at the end. The RCAB has a global average pooling at the beginning and a Tanh activation layer at the end. For residual learning,
the inputs of RU, RIRB and RB are added to their outputs, the input of RCAB is multiplied to its output as well. The formulation of $O_{RB,i}$ can be represented as:

$$O_{RB,i} = RCAB(Conv(ReLU(Conv(I_{RB,i})))) + I_{RB,i},$$

(9)

and the output of RCAB can be formulated as:

$$O_{RCAB} = Tanh(Conv(Conv(Pooling(I_{RCAB})))) * I_{RCAB},$$

(10)

where $O_{RB,i}$ denotes the output of the i-th RB, $I_{RB,i}$ denotes the input of the i-th RB, $O_{RCAB}$ denotes the output of RCAB, $I_{RCAB}$ denotes the input of RCAB. $ReLU$, $Pooling$ and $Tanh$ denote the Rectified Linear Unit, the average pooling and the hyperbolic tangent function, respectively.

Inspired by [29], we train the main phase of ETOM-Net with four different loss scales. This multi-scale loss starts with a feature map size of $64 \times 64 \times 64$ and ends with a size of $8 \times 512 \times 512$ (the same size as the output mattes), named scale 0 to scale 3. In addition, we apply different weights to different loss scales to make the network more focused on large-scale features. The scale of the super-resolution in our proposed method ETOM-Net is set to $\times 2$, and can be extended to $\times 3$ and $\times 4$ by training the model using different scales.

Along with the main phase, we add a refinement phase using residual learning to produce more detail to the output mattes of the main phase. As shown in Fig. 4, the refinement phase takes the LR input, and three output environment mattes from the main phase as input, and then the input tensor is passed through several downsampling blocks, five RBs and several upsampling blocks to form the output mattes. Each RB consists of two convolutional layers and a ReLU activation layer, the input of RB is then subjected to an average pooling operation and added to the output of RB as the final output. The output of the refinement phase can be represented as:

$$O_{\text{refine}} = Conv(Conv(ReLU(Conv(Up(Conv(Conv(I_{mask} + I_{flow} + I_{rho} + I_{ir}))))))),$$

(11)

and the output of $RB_i$ can be formulated as:

$$O_{RB,i} = Conv(Conv(ReLU(Conv(I_{RB,i})))) + I_{RB,i},$$

(12)

where $O_{\text{refine}}$ denotes the output of the refinement phase, $Up$ denotes the upsampling process, and $Down$ denotes the downsampling process. $I_{mask}$, $I_{flow}$, $I_{rho}$, and $I_{ir}$ denote the inputs of the refinement phase, including a pixelwise mask, a refractive flow, an attenuation map, and an LR input, respectively.

**B. LOSS FUNCTION**

1) Main Phase

The loss function $\mathcal{L}_{\text{main}}$ of the main phase is divided into four parts similar to [29]: a pixelwise mask loss $\mathcal{L}_{\text{mask}}$, a refractive flow field loss $\mathcal{L}_{\text{flow}}$, an attenuation loss $\mathcal{L}_{\text{rho}}$ and a reconstruction loss $\mathcal{L}_{\text{rec}}$. The loss function of the main phase can then be denoted as

$$\mathcal{L}_{\text{main}} = \lambda_1 \mathcal{L}_{\text{mask}} + \lambda_2 \mathcal{L}_{\text{flow}} + \lambda_3 \mathcal{L}_{\text{rho}} + \lambda_4 \mathcal{L}_{\text{rec}},$$

(13)

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weights of each component of the loss.

**Segmentation mask loss** We define pixelwise mask segmentation as a typical classification problem. The output mask has two channels, representing the probabilities of the foreground and the background, respectively. Simply put, a pixel is part of the transparent object if the value of its first channel is more significant and vice versa. In this case, we compute the mask loss $\mathcal{L}_{\text{mask}}$ using the cross-entropy loss

$$\mathcal{L}_{\text{mask}} = \text{Mean} \left[ -\log \left( \frac{\exp(P_{ij}[C_{ij}])}{\sum_{k=0}^{1} \exp(P_{ij}[k])} \right) \right]$$

(14)

where $P_{ij} = (P_{ij}^f, P_{ij}^b)$ denotes the probability of the pixel at $(i, j)$ belongs to the foreground and the background, respectively, and $C_{ij} \in \{0, 1\}$ denotes the ground truth of the pixel at $(i, j)$ ($C_{ij} = 0$ means the pixel at $(i, j)$ is a foreground pixel). Mean denotes the average of all pixels.

**Refractive flow field loss** The output refractive flow of the main phase has two channels, representing the horizontal and vertical displacements, respectively. The output value is in the range of $[-1, 1]$ because of the $Tanh$ activation function. We multiply them by different ratios at different scales, so the output flow has the same range as the width. For this one, we use the average endpoint error (AEE) loss, which is defined as the mean of the Euclidean distance (Frobenius norm) between the estimated flow and the ground truth flow

$$\mathcal{L}_{\text{flow}} = \text{Mean} \left[ \left\| F - \tilde{F} \right\|_F^2 \right],$$

(15)

where $(\tilde{F}_{ij}^x, \tilde{F}_{ij}^y)$ denotes the output refractive flow at $(i, j)$, and $(\tilde{F}^x_{ij}, \tilde{F}^y_{ij})$ denotes the ground truth refractive flow at $(i, j)$.

**Attenuation map loss** The value of the output attenuation map is in the range $[0, 1]$, showing how much light can pass through the object. We use the mean square error (MSE) loss

$$\mathcal{L}_{\text{rho}} = \text{Mean} \left[ (A_{ij} - \hat{A}_{ij})^2 \right],$$

(16)

where $A_{ij}$ is the output attenuation map at $(i, j)$, and $\hat{A}_{ij}$ is the ground truth attenuation map at $(i, j)$.

**Reconstruction loss** To evaluate the quality of composited images, we reconstruct the image using the output environment mattes and the corresponding HR ground-truth background and compare it to the ground truth HR input image. As with the attenuation map loss, we use the MSE loss

$$\mathcal{L}_{\text{rec}} = \text{Mean} \left[ (V_{ij} - \hat{V}_{ij})^2 \right],$$

(17)
where $V_{ij}$ is the pixel value of $(i,j)$ in the reconstructed image, and $\hat{V}_{ij}$ is the pixel value of $(i,j)$ in the ground truth image.

2) Refinement phase

Similar to the main phase, the loss function $L_{\text{refine}}$ of the refinement phase has two parts: a pixelwise mask segmentation loss $L_{\text{mask}}$ and a refractive flow field loss $L_{\text{flow}}$. The loss function of the refinement phase can then be denoted as

$$L_{\text{refine}} = \lambda_1 L_{\text{mask}} + \lambda_2 L_{\text{flow}},$$

(18)

where $\lambda_1, \lambda_2$ are the weights of each component of the loss, and the $L_{\text{mask}}$ and the $L_{\text{flow}}$ are the same as in the main phase.

C. COMPARISON

Here, we compare the similarities and differences between TOM-Net [29] and our proposed method ETOM-Net. Fig. 5 illustrates a brief comparison which does not include implementation differences within each block.

In the main phase, similar to TOM-Net, we use an encoder-decoder structure with three independent decoding processes to generate three environment mattes. In addition, both methods use skip connections to connect feature maps of the same size and multi-scale losses with four different scales.

Unlike TOM-Net, our approach takes LR images as input and predicts HR environment mattes by adding three RUs between the encoding and decoding processes. Each RU consists of four RIRBs, which are stacks of residual blocks using channel-wise attention mechanism and residual learning, as shown in Fig. 3.

In the refinement phase, both methods use residual learning to refine the mattes predicted by the main phase. However, our refinement phase takes upsampled LR image and the output mattes of the main phase as input and predicts only refined segmentation mask and refractive flow field with the same attenuation map as the input.

As we mentioned before, the quality of the mask does have a significant impact on the reconstructed image. Thus, compared to the refinement loss in the TOM-Net, we add a mask loss to improve the quality of the output mask further as its edges are not smooth enough after being super-resolved in the main phase. Moreover, we remove the attenuation map loss because the output attenuation map of the main phase is good enough that the refinement phase cannot improve it in any way, and training with it will slow down the convergence of the mask and refractive flow. As with TOM-Net, we do not include reconstruction loss in the loss function during the refinement phase because it does not help to preserve the sharp edges of the refractive flow field.

V. EXPERIMENTAL RESULTS

A. DATASET

Chen et al. [29] created a large-scale synthetic dataset because there was no off-the-shelf dataset for transparent object matting. This dataset consists of background images, input images, ground truth segmentation masks, attenuation and refractive flows, with a total of 178,000 samples for training. They also created a validation dataset with 900 samples for testing.
In our work, we also use this dataset, and since we are mainly concerned with super-resolution environment matting, and TOM-Net has demonstrated good generality of multi-scale encoder-decoder structures from basic to complex shapes, we use only part of their dataset (glass and glass with water) to reduce the training and testing time. Therefore, we used a dataset with 60,000 training samples and 400 validation samples, which saved us much time, and we could use them for the ablation study of several variants of the proposed method.

### B. IMPLEMENTATION DETAILS

For training settings, we used a batch size of eight in the main phase and four in the refinement phase, with learning rates starting at 0.0005 and 0.0002 for the main and refinement phases, respectively. We also decayed the learning rate by half every five epochs. For the optimizer, we used the Adam algorithm ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-08}$) and applied L2 penalty to prevent overfitting.

For the loss function Eq. 13 in the main phase, we set the segmentation mask weight $\lambda_1 = 1$, the refractive flow weight $\lambda_2 = 0.1$, the attenuation weight $\lambda_3 = 10$, and the reconstructed image weight $\lambda_4 = 10$. For the refinement phase loss in Eq. 18, we set the segmentation mask weight $\lambda_1 = 10$, and the refractive flow weight $\lambda_2 = 0.1$. Since we used four different loss scales in the main phase of training, we weighted them by 1/8, 1/4, 1/2 and 1 from scale 0 to scale 3, respectively, to make the network focus on the larger scale.

During the training process, the input images in the dataset are downsampled by a factor of two and used as the input to the ETOM-Net main phase, and subsequently, its output is concatenated with the input of the main phase and sent to the refinement phase. The refinement phase outputs the refined segmentation mask and refraction flow field but does not do anything to the attenuation map. Once the training is complete, we can use the trained models main phase and refine phase to predict an HR environment matte using a single LR input image in a single pass.

Using the Adam optimizer, the processing time on an Nvidia 2080Ti GPU is shown in Table 1. Note that the refinement phase takes longer to test than the main phase, partly because the refinement phase requires the results of the main phase, so it runs two models, while the main phase only needs to run one model. We also train and test TOM-Net with the same dataset, as shown in Table 1. Although our network requires longer training time, the model runs almost as fast as TOM-Net at testing time. The best model for each phase is selected based on the validation results of each epoch using early stopping. The code for ETOM-Net can be found on GitHub: https://github.com/1asso/ETOM-Net.

### C. RESULTS

In the experiments, we compare the refinement phase of the ETOM-Net against the main phase, the TOM-Net (use bicubic upsampled images as input), and a super-resolution method with the TOM-Net, as shown in Fig. 6.

Here, we use the EDRN [22] to achieve super-resolution because it employs an encoder-decoder residual network and
a channel-wise attention mechanism, which is similar to the structure of our ETOM-Net. A single input image is fed to the EDRN to produce a super-resolved output image, and then the output is used as the input to the TOM-Net to predict the high-resolution environment matte. We introduce a baseline model to produce the worst results by using the corresponding background image as the output reconstructed image, two one-filled tensors as the output attenuation and segmentation mask to simulate null attenuation and mask, and use a zero-fill tensor as the output refractive flow field to simulate a null flow without offsets.

Quantitative results of ETOM-Net are shown in Table 2. We can see that all methods are much better than the baseline model, which indicates that they can estimate the environment matte successfully. Compared to the original TOM-Net, SR + TOM-Net produces poorer results in all metrics because the super-resolution approach introduced many unrealistic synthetic details, which affect the performance of TOM-Net. TOM-Net itself can already predict visually good results by taking bicubic upsampled images as input. However, the main phase of our ETOM-Net has beaten TOM-Net in terms of reconstructed images and attenuation. In particular, the
main phase produces much better results than TOM-Net in terms of attenuation, which demonstrates the effectiveness of using the RU with an encoder-decoder network. The ETOM-Net refinement phase further improves the output refractive flow and segmentation mask of the main phase, giving better results for both metrics than the main phase. It also produces the best overall results among all tested methods, showing the effectiveness of the refinement phase.

Fig. 7 presents some qualitative results of our ETOM-Net compared to that of the TOM-Net and SR + TOM-Net. Our method produces smoother borders in terms of the output segmentation mask, and the stems of the wine glasses look more natural than that of the other methods. The ETOM-Net output attenuation has more detail, especially in the feet and stems, and the reconstructed images have more detail and more realistic refraction.

VI. ABLEST STUDY

In order to understand the effectiveness of each component in our ETOM-Net, we create several variants of ETOM-Net and analyzed them quantitatively. Here, we focus on the RU, the object mask loss ($L_{\text{mask}}$) and the attenuation loss ($L_{\text{rho}}$) in the refinement phase. We remove the RU and half of the RU to create model main - RU and model main - RU (part), respectively, and remove $L_{\text{mask}}$ in the refinement phase to create model refine - ($L_{\text{mask}}$), and include $L_{\text{rho}}$ to create model refine + ($L_{\text{rho}}$). We also add a baseline model by using the same tactics as in the previous section.

Similar to that in the experimental results, we use the IoU for evaluating object masks, EPE for refractive flow, and MSE for attenuation and reconstructed image, respectively. The quantitative results are presented in Table 3.

Table 3: Ablation study results

| Variants          | Rec↓ | Attenuation↓ | Flow↓ | Mask↑ |
|-------------------|------|--------------|-------|-------|
| Baseline          | 0.696| 1.426        | 2.569 | 0.102 |
| main - RU (part)  | 0.215| 0.209        | 1.583 | 0.957 |
| main - RU         | 0.221| 0.225        | 1.602 | 0.954 |
| main              | 0.201| 0.203        | 1.579 | 0.958 |
| refine - ($L_{\text{mask}}$) | 0.204| 0.206        | 1.553 | 0.957 |
| refine + ($L_{\text{rho}}$) | 0.211| 0.202        | 1.605 | 0.957 |
| main + refine     | 0.192| 0.203        | 1.516 | 0.968 |

respectively, and remove $L_{\text{mask}}$ in the refinement phase to create model refine - ($L_{\text{mask}}$), and include $L_{\text{rho}}$ to create model refine + ($L_{\text{rho}}$). We also add a baseline model by using the same tactics as in the previous section.

Similar to that in the experimental results, we use the IoU for evaluating object masks, EPE for refractive flow, and MSE for attenuation and reconstructed image, respectively. The quantitative results are presented in Table 3.

First of all, all variants, including the main and the refine phases of the ETOM-Net, unquestionably exceed the baseline by a large margin in all evaluation metrics. Removing the RU from the main phase degrades the overall performance, and removing half of the RU gives better results than removing all of them, which indicates that the number of RIRBs inside the RU does have an impact on performance.

For the refinement phase, removing $L_{\text{mask}}$ or adding $L_{\text{rho}}$ to the loss function leads to a decrease in performance, and although the presence of $L_{\text{rho}}$ further improves the attenuation metric, the other three metrics decrease as a result. In general, the main phase with the refinement phase produces the best results. Fig. 8 shows the effectiveness of the refinement phase.

We also evaluate how the number of RIRBs within each RU affects the results and training. As shown in Fig. 9, figure...
(a) shows the relationship between the mean square error of the output reconstructed image and the number of RIRBs, and figure (b) shows the effect of the number of RIRBs on the GPU memory footprint (megabytes). From Fig. 9 (b), we can see that the GPU cost is positively correlated with the number of RIRBs. We used a batch size of eight in our tests, and for eight and sixteen RIRBs, we had to halve the batch size to four in order to enable the model to be trained, and in the absence of gradient clipping, the training process was volatile because of the presence of gradient explosion. From Fig. 9 (a), the results are getting better from no RIRBs to eight RIRBs, especially from two to four, achieving a leap in performance. For sixteen RIRBs, the performance drops, probably because of the overly complex network. By weighing the performance gain against the GPU footprint of training, we choose four as the number of RIRBs within each RU in our model.

VII. CONCLUSION

In this paper, we combine SISR and environment matting to propose an efficient CNN-based method for super-resolution of environment matting of transparent objects called ETOM-Net.

The proposed network uses an encoder-decoder architecture with skip connections and multi-scale losses that takes a single low-resolution image as input and then estimates the corresponding high-resolution environment matting, including a refractive flow field, a pixelwise segmentation mask, and an attenuation map.

Three restoration units are added between the encoding and the decoding processes to allow the network to focus on more informative parts of the low-resolution input and restore more details to the high-resolution output environment matting. Furthermore, a refinement network using residual learning is introduced to improve the details of the output segmentation mask and the output refractive flow of the main phase further.

ETOM-Net produces visually plausible results and outperforms the baseline model by a large extent. Compared to the TOM-Net and SR+TOM-Net models, the main phase of our method already outperforms them in terms of the reconstructed image and the attenuation map. With the refinement phase, ETOM-Net produces the best overall results among all models, which demonstrates the effectiveness of our proposed method.

Although the proposed method ETOM-Net is very effective, it is limited by the fixed-scale super-resolution and can only be applied to transparent objects with a single mapping (one mapping at each point), which will we explore in future work.

REFERENCES

[1] R. Brinkmann, “Introduction to Digital Compositing,” The Art and Science of Digital Compositing, pp. 1–14, 2008.
[2] W. Fan and D.-Y. Yeung, “Image Hallucination Using Neighbor Embedding over Visual Primitive Manifolds,” 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[3] J. Sun, Z. Xu, and H.-Y. Shum, “Image super-resolution using gradient profile prior,” 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[4] W.-T. Sai, S. Liu, M. S. Brown, and S. Lin, “Super resolution using edge prior and single image detail synthesis,” 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010.
[5] M. Bevilacqua, A. Roumy, C. Guillelmet, and M.-L. A. Morel, “Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding,” Proceedings of the British Machine Vision Conference 2012, 2012.
[6] R. Fattal, “Image upsampling via imposed edge statistics,” ACM SIGGRAPH 2007 papers on - SIGGRAPH 07, 2007.
[7] R. Zeyde, M. Elad, and M. Protter, “On Single Image Scale-Up Using Sparse-Representations,” Curves and Surfaces Lecture Notes in Computer Science, pp. 711–730, 2012.
[8] W. Freeman and E. Pasztor, “Learning low-level vision,” Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.
[9] Z. Lin and H.-Y. Shum, “Fundamental limits of reconstruction-based super-resolution algorithms using local translation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 1, pp. 83–97, 2004.
[10] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.
[11] W. Freeman, T. Jones, and E. Pasztor, “Example-based super-resolution,” IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56–65, 2002.
[12] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[13] C. Dong, C. C. Loy, K. He, and X. Tang, “Image Super-Resolution Using Deep Convolutional Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295–307, 2016.
[14] C. Dong, C. C. Loy, and X. Tang, “Accelerating the Super-Resolution Convolutional Neural Network,” Computer Vision – ECCV 2016 Lecture Notes in Computer Science, pp. 391–407, 2016.
[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[16] J. Kim, J. K. Lee, and K. M. Lee, “Accurate Image Super-Resolution Using Very Deep Convolutional Networks,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[17] Y. Tai, J. Yang, and X. Liu, “Image Super-Resolution via Deep Recursive Residual Network,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[18] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-Recursive Convolutional Network for Image Super-Resolution,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[19] G. LeCun, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[20] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution,” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.
[21] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image Super-Resolution Using Very Deep Residual Channel Attention Networks,” Computer Vision – ECCV 2018 Lecture Notes in Computer Science, pp. 294–310, 2018.
[22] G. Cheng, A. Matsune, Q. Li, L. Zhu, H. Zhang, and S. Zhan, “Encoder-Decoder Residual Network for Real Super-Resolution,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
[23] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin, “Environment matting and compositing,” Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH 99, 1999.
[24] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski, “Environment matting extensions,” Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH 00, 2000.
[25] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman, “Image-based environment matting,” ACM SIGGRAPH 2002 conference abstracts and applications on - SIGGRAPH 02, 2002.
[26] P. Peers and P. Dutré, “Wavelet environment matting”, Eurographics workshop on Rendering, 2003.
[27] J. Zhu and Y.-H. Yang, “Frequency-based environment matting,” 12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.
[28] Y. Qian, M. Gong, and Y.-H. Yang, “Frequency-Based Environment Matting by Compressive Sensing,” 2015 IEEE International Conference on Computer Vision (ICCV), 2015.
[29] G. Chen, K. Han, and K.-Y. K. Wong, “TOM-Net: Learning Transparent Object Matting from a Single Image,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[30] J. Shi, Y. Dong, H. Su, and S. X. Yu, “Learning Non-Lambertian Object Intrinsics Across ShapeNet Categories,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
ZICUN HANG received the B.Sc. degree from Northeastern University, Shenyang, China, in 2019, and the M.Sc. degree from University of Alberta, Edmonton, Canada, in 2021. His research interests include computer vision and pattern recognition.

YEE-HONG YANG received the B.Sc. degree (Hons.) from The University of Hong Kong, the M.Sc. degree from Simon Fraser University, and the M.S.E.E. and Ph.D. degrees from the University of Pittsburgh. He was a Faculty Member with the Department of Computer Science, University of Saskatchewan, from 1983 to 2001, and was the Graduate Chair, from 1999 to 2001. In addition to the department-level committees, he was also with many college and university level committees. From 2003 to 2005, he was the Associate Chair (Graduate Studies) with the Department of Computing Science, University of Alberta, where he has been a Professor, since 2001. He has authored or co-authored over 140 papers in international journals and conference proceedings in the areas of computer vision and graphics. His research interest includes a wide range of topics from computer graphics to computer vision. He has served on the program committees of many national and international conferences. In 2007, he was invited to serve on the Expert Review Panel to evaluate computer science research in Finland. He serves on the Editorial Board of the journal Pattern Recognition. He served as a Reviewer for numerous international journals, conferences, and funding agencies.

***