Learning Super-Resolution of Environment Matting of Transparent Objects from a Single Image

by

Zicun Hang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

University of Alberta

Abstract

This thesis 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 thesis 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. Meanwhile, to facilitate future research, we create a high-resolution synthetic dataset called ETOM-Synthetic with 60,000 samples. The ETOM-Net effectively recovers lost features in the low-resolution input images and produces visually plausible high-resolution environment mattes and the corresponding reconstructed images, demonstrating our method's effectiveness.

Preface

This thesis is an original work by Zicun Hang. No part of this thesis has been previously published.

To my supervisor Herb Yang

For helping me in doing my research.

Acknowledgements

I would like to express my sincere gratitude to Dr. Yang for his help in my research. He gave me some valuable suggestions when I was overwhelmed with the direction of my research and guided me to get out of difficult situations whenever my research hit a bottleneck. Without him, I could not have completed this work.

I would also like to thank my parents for their financial help and spiritual support for my research.

Contents

List of Tables

List of Figures

Glossary

Average Endpoint Error (AEE)

The mean Euclidean distance between two optical flows.

Batch Normalization (BN)

A technique that normalizes each small batch of inputs to a single layer.

Enhanced Transparent Object Matting Network (ETOM-Net)

The method we proposed.

Mean Square Error (MSE)

The average squared difference between the estimated values and the actual value.

Multi Image Super Resolution (MISR)

Super resolution using multiple input images.

Rectified Linear Unit (ReLU)

A linear function that will output the input directly if it is positive, otherwise, it will output zero.

Residual Channel-wise Attention Block (RCAB)

A set of layers using channel-wise attention mechanism and residual learning.

Residual Block (RB)

A stack of layers that the output of a layer is taken and added to another layer more profound in the block.

Residual In Residual Block (RIRB)

A stack of residual blocks.

Restoration Unit (RU)

A set of layers used in our method to restore lost details.

Single Image Super Resolution (SISR)

Super resolution using a single input image.

Chapter 1

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 [2] is defined as follows:

$$C = \alpha F + (1 - \alpha)B, \tag{1.1}$$

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

As Eq. 1.1 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.[30] introduce the environment matting to capture how light in the environment is refracted and reflected by foreground objects. The new model has the following form:

$$C = F + (1 - \alpha)B + \Phi, \tag{1.2}$$

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

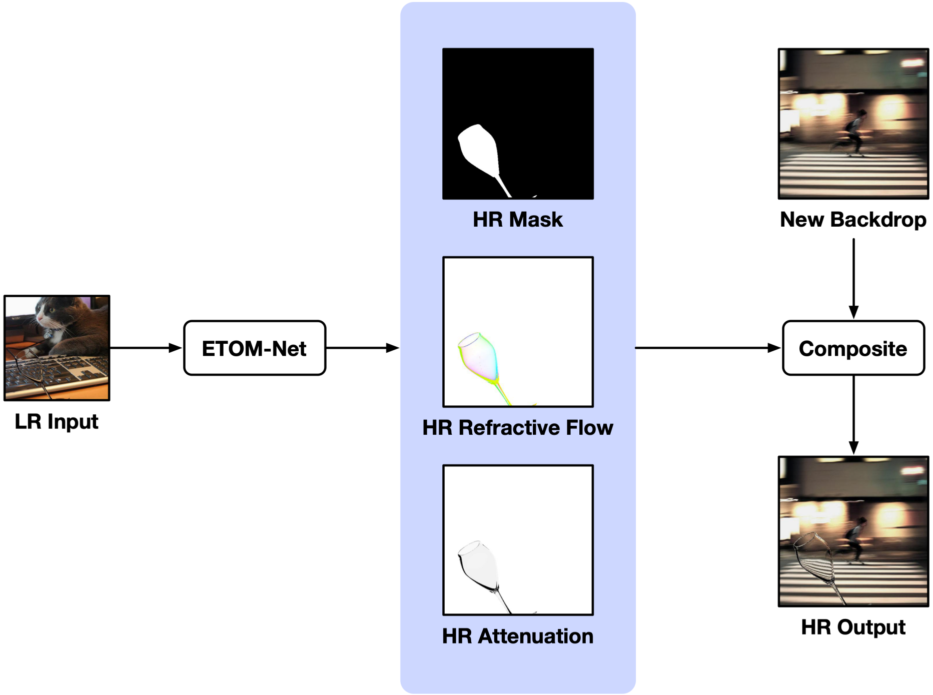

Figure 1.1: High resolution environment matte estimation and image composition.

After the work of Zongker et al.[30], many approaches [6], [19], [20], [25], [29] 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. [4] 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 a high-resolution image from a single low-resolution 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 this thesis, we focus on super-resolution of environment matting of transparent objects from a single image.

Following [4], we define the environment matte in our work as a triple, which consists of a pixelwise segmentation mask, a refractive flow field and an attenuation map. The segmentation mask is used to locate the boundary of the foreground transparent object, the attenuation map denotes how much the light is attenuated by the object, and the refractive flow field represents the pixel offset between the image of the object and its corresponding background image. To estimate a high-resolution environment matte, we propose a CNN-based method called ETOM-Net. As shown in Fig. 1.1, given a low-resolution image containing a transparent object as input, ETOM-Net estimates the corresponding high-resolution environment matte, which is then used to synthesize the transparent objects onto a new background, resulting in a high-resolution output image. The contributions of this thesis can be summarized as follows:

1. We propose an encoder-decoder network in the main phase with three restoration units for super-resolution environment matting. The network effectively recovers lost features in low-resolution input images and produces visually plausible high-resolution environment matte and synthesized images.

2. In addition to the main phase, we incorporate a refinement phase with residual learning to improve the quality of the high-resolution environment matte and the reconstructed image.

3. The authors of [4] created a synthetic dataset because there was no readily available dataset for learning transparent environment matting. Although we use this dataset in our work, we also create a higher resolution synthetic dataset for our work and for future use by other researchers.

Chapter 2

Related Work

In this chapter, we introduce representative works on single image super-resolution, environment matting, and recently proposed deep learning-based methods.

2.1 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 [1], [7], [11], [18], [22], [29].

SISR can be categorized into four types: interpolation-based methods, reconstruction-based methods, example-based methods and learning-based methods.

- Interpolation-based methods (e.g., bicubic interpolation) are fast and straightforward, but they cannot generate high accuracy results with sharp edges.

- Reconstruction-based approaches (e.g., [10], [18], [22], [23]) utilize prior to achieve relatively good results. However, these methods are not very efficient and their performance will degrade due to inaccurate prior knowledge.

- Example-based methods (e.g., [1], [3], [9], [11], [12], [26], [27]) use a small training database with LR and HR image pairs to predict high-frequency details. The limitation of these methods is that the quality of the results depends heavily on the training images.

- Learning-based approaches (e.g., [7], [14], [16], [24]) use deep learning techniques to learn the statistical relationship between LR and HR image pairs from a large training dataset. Deep learning neural networks that have many layers each with different width can theoretically approximate arbitrary functions. Thus they can solve the very complex mapping problem of the mapping between LR and HR. Also, the quality of the results is highly dependent on the amount of training data. The more the data, the better is the performance. From the experimental results published in recent years, deep learning has an excellent learning ability, and methods based on it have achieved state-of-the-art results.

Dong et al. first introduce a neural network model into SISR, called super-resolution convolutional neural network (SRCNN) [7]. The authors preprocess the input image using bicubic interpolation to scale it to the desired resolution. After the preprocessing, they use a three-layer CNN to learn an end-to-end mapping $\mathcal{F}$ between the low-resolution and the high-resolution images.

The proposed network SRCNN consists of three steps: The first step is a patch extraction and representation operation. It extracts patches by convolving the input low-resolution image with a set of filters and adding a bias to each filter, and the output patches are represented as vectors of high-dimensional feature maps. The second step is a nonlinear mapping operation that maps each high-dimensional feature map from the first step to another high-dimensional feature map that is conceptually a representative of the high-resolution patch that will be used in the next step. The last step is a reconstruction operation, which uses high-resolution representations to generate a super-resolved image. The proposed network is defined as follows:

$$F(\mathbf{Y}) = W_3 * \max(0, W_2 * \max(0, W_1 * \mathbf{Y} + B_1) + B_2) + B_3, \tag{2.1}$$

where Y denotes the low-resolution input image, W_i and B_i represent the filters and biases in each step, respectively. They use the Mean Squared Error (MSE) loss as the loss function and minimize it with stochastic gradient descent.

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. Moreover, the authors hypothesize that higher performance could be obtained by adding more hidden layers/filters to the network and by using different training strategies, leading the way for subsequent research.

Fast super-resolution convolutional neural networks (FSRCNN) [8] is an improvement of SRCNN, as the high computational cost of SRCNN still hinders its real-time performance requirements in practical applications. Instead of scaling up the low-resolution input at the beginning as SRCNN does, it processes the low-resolution image directly and applies a deconvolution layer at the end to scale the results to the correct size. In SRCNN, the nonlinear mapping step follows the feature extraction step, and then the high-dimensional low-resolution features are directly mapped to the high-resolution feature space. However, the computational complexity of the nonlinear mapping step is quite high because the dimensionality of low-resolution features is usually very large. To solve this problem, they add a shrinking layer after the feature extraction layer to reduce the dimensionality of the low-resolution feature. The filters in this layer behave like linear combinations in low-resolution features. This strategy greatly reduces the number of parameters in their model. An expanding layer is added after the nonlinear mapping process as an inverse process of the shrinking layer to expand the high-resolution feature dimension and improve the final restoration quality. Compared to SRCNN, FSRCNN runs more than 40 times faster, but still maintains good performance.

Later, residual neural network (ResNet) [13] proposed by He et al. incorporates a residual learning framework to improve the training of very deep networks, since deeper neural networks are more difficult to train. Unlike normal convolutional layers that learn the underlying mapping directly, the authors let these layers adapt to a residual mapping. The residual learning can be formulated as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}) + \mathbf{x}, \tag{2.2}$$

where $\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x}$, $\mathbf{x}$ and $\mathbf{y}$ denote the input and the output, respectively, and $\mathcal{H}(\mathbf{x})$ denotes the desired underlying mapping.

The authors assume that it is easier to optimize the residual mapping than the original one. For example, if the optimum is an identity mapping, then it is easier to learn a zero residual than to fit the identity mapping with nonlinear layers. A residual learning with two layers is shown in Fig. 2.1. In this case, $\mathcal{F}(\mathbf{x}) = W_2\sigma(W_1\mathbf{x})$ in which σ denotes Rectified Linear Unit (ReLU) and W_i denotes the weight of each layer. A shortcut connection and element-wise addition are used to perform $\mathcal{F}(\mathbf{x}) + \mathbf{x}$.

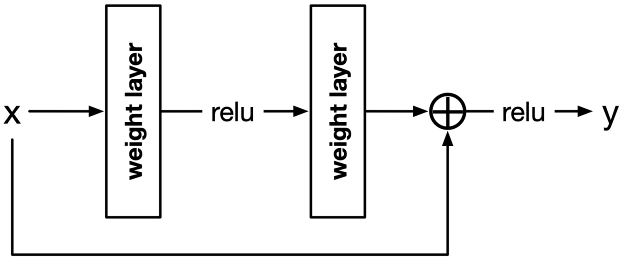

Figure 2.1: Residual learning.

In the experiments, a general network with 34 parameter layers inspired by VGG nets is used as the baseline model, and on top of this general network, the authors insert shortcut connections to turn the general network into its corresponding residual version. Compared to the baseline, the residual version model has much lower training errors and is easier to optimize. Moreover, by increasing the depth of the network substantially, higher accuracy can be obtained.

Since the introduction of ResNet, researchers have explored the possibility of using residual learning in SISR. In very deep super-resolution (VDSR) [14], Kim et al. present a very deep convolutional network for highly accurate SISR. The proposed network takes as input the interpolated low-resolution image (scale up to the desired resolution) and predicts lost details (residual information). d layers are used in this network, all of which are of the same type: 64 filters of size 3 × 3 × 64, except for the first and last layer. The first layer operates on the input image and the last layer is used for image reconstruction which consists of a single filter of size 3 × 3 × 64. Since the size of the feature map decreases each time a convolution operation is applied, the authors pad zeroes before each convolution to keep the size of all feature maps, including the output image, the same. Once the residual information is predicted, it is added back to the input low-resolution image with global residual learning to generate the high-resolution output image. The authors also demonstrate that very deep networks can achieve high performance in super-resolution, with performance increasing rapidly as depth grows. Using a network with a depth of 20 and trained at a very high learning rate, their proposed method VDSR outperforms existing methods by a large margin and has a very fast convergence rate.

Proposed by the same group, the deeply-recursive convolutional network (DRCN) [15] first applies a deeply recursive neural network to SISR. The authors first introduce a base model, which consists of three parts: an embedding network, an inference network and a reconstruction network. The embedding network at the beginning accepts a low-resolution input image and represents it as a set of feature maps. The inference network in the middle consists of a single recursive layer that handles the task of super-resolution. The recursive layer has D recursions, each of which applies the same convolution followed by a ReLU to widen the receptive field. The reconstruction network at the end converts the multi-channel high-resolution feature maps into a high-resolution final image. Although the base model is powerful, it is difficult to train due to vanishing and exploding gradients. The authors then propose an improved model that addresses these issues by using recursive supervision and skip-connections. They demonstrate that DRCN outperforms existing methods by a significant amount on benchmark images.

Inspired by DRCN, the deep recursive residual network (DRRN) [24] extends VDSR by introducing recursive blocks with residual units. The recursive block is formulated as:

$$H^u = \mathcal{F}\left(H^{u-1}, W\right) + H^0, \tag{2.3}$$

where $u = 1, 2, \cdots, U$, U is the number of residual units in each recursive block, H^0 is the output of the first convolutional layer in the recursive block, H^{u-1} and H^u are the input and output of the u-th residual unit, $\mathcal{F}$ denotes the residual function, and W is a shared weight set within the recursive block. The structure of the recursive block is shown as Fig. 2.2, it starts with one convolutional layer and then stacks U residual units. The final network is created by stacking B recursive blocks and one convolutional layer, as shown in Fig. 2.3, which reconstructs the residual between the input low-resolution and output high-resolution images. This residual is then added element-wise to the global identity mapping of the input low-resolution image.

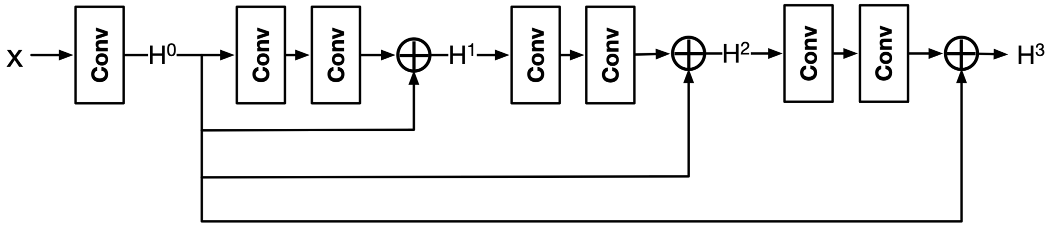

Figure 2.2: Structure of the recursive block with $U = 3$.

In contrast to 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.

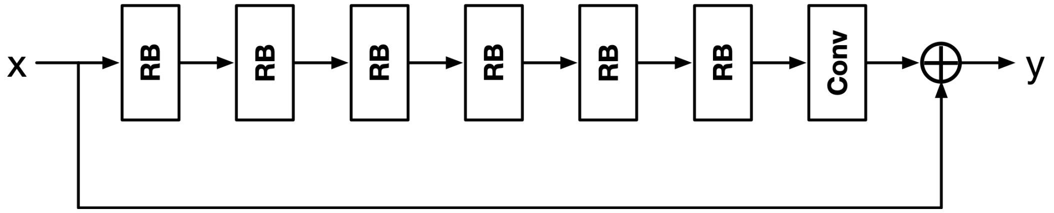

Figure 2.3: Network structure of DRRN with $B = 6$.

Ledig et al. present super-resolution residual network (SRResNet) [16] with 16 blocks deep ResNet as part of the super-resolution generative adversarial network (SRGAN), which is the first time that the Generative Adversarial Network (GAN) is applied to image super-resolution. SRGAN consists of two networks: a generator network and a discriminator network. The generator network trains a generating function G that estimates the corresponding high-resolution counterpart given a low-resolution input image, which can be formulated as:

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{n=1}^{N} l^{SR} \left(G_{\theta_G} \left(I_n^{LR} \right), I_n^{HR} \right), \tag{2.4}$$

where θ_G are the parameters of the generator network G_{θ_G}, $n = 1, \cdots, N$, N is the number of training images, and l^{SR} is a perceptual loss function. The discriminator network is trained to distinguish between super-resolution images and real images, which allows us to train the generative model G with the aim of fooling the discriminator. The adversarial min-max problem is defined as follows:

$$\begin{aligned} \min_{\theta_G} \max_{\theta_D} & \mathbb{E}_{I^{HR} \sim p_{\text{train}} (I^{HR})} \left[\log D_{\theta_D} \left(I^{HR} \right) \right] + \\ & \mathbb{E}_{I^{LR} \sim p_G(I^{LR})} \left[\log \left(1 - D_{\theta_D} \left(G_{\theta_G} \left(I^{LR} \right) \right) \right] . \end{aligned} \tag{2.5}$$

The authors design the perceptual loss as a combination of content loss and adversarial loss, which are pixel-level MSE loss and discriminator probability on all training samples, respectively. SRResNet creates new state-of-the-art results on benchmark datasets, and by training with adversarial loss, SRGAN produces more photo-realistic results in terms of mean-opinion-score (MOS) testing.

Soon after, Lim et al. remove the Batch Normalization (BN) from the residual blocks of SRResNet and propose enhanced deep super-resolution network (EDSR) [17], which can stack more residual blocks under the same condition with residual scaling techniques. The authors also extend their single-scale model EDSR to multiple scales by proposing a multi-scale deep super-resolution system (MDSR) with scale-dependent modules and a shared master network. Both their proposed single-scale and multi-scale models achieve the highest rankings in standard benchmark datasets and the DIV2K dataset.

The residual channel attention network (RCAN) [28] uses a residual structure to form a very deep network, and since the low-resolution inputs and features containing rich low-frequency information are treated equally on different channels, hindering the representational power of the network, the authors employ a channel attention mechanism that adaptively rescale the features on the channels, allowing the network to focus on more informative features. The channel attention mechanism is shown in Fig. 2.4. It has a global average pooling layer, two convolutional layers with a ReLU in between, and a sigmoid function. After the sigmoid function, the final channel statistics is obtained, which is then used to rescale the input x by element-wise production. RCAN proves to be effective in super-resolution of bicubic (BI) and blur-downscale (BD) degradation models, and it also shows good results in object recognition.

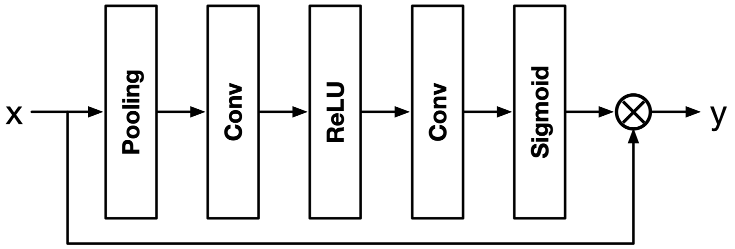

Figure 2.4: Channel attention mechanism, $\otimes$ denotes element-wise product.

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 the encoder-decoder residual network (EDRN) [5]. 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 high-resolution images from real-world low-resolution images and is one of the best methods of NTIRE 2019 Real SR Challenge.

2.2 Environment Matting

Environment matting first introduced by Zongker et al. [30] 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 as defined in Eq. 1.2. 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. [6] 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 by simplifying the matting equation under the assumption that the object is colorless and has no roughness or translucency, which allows them to achieve a real-time environment matting of 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 requires at least two input images: one containing the transparent object and the other containing only the background to compute the receptive field of pixel p (a set of pixels that contribute to a particular output pixel p). The authors demonstrate that their proposed method works well given sufficiently rich backgrounds or enough images, but has the limitation that diffuse scattering affects the estimation of the probability density.

Peers et al. [19] 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. As a result, their method can handle any kind of material properties. Large areas of highly specular materials may still be captured, but this is problematic due to the slow convergence rate. Their method also exploits the idea of linearly combining basis images, requiring only minimal post-processing, whereas previous methods for environment matting require per-pixel optimization procedures.

Inspired by the fact that a time-domain signal has a unique decomposition in the frequency domain, Zhu and Yang [29] transfer the environment matting problem from the time domain to the frequency domain and introduce a frequency-based environment matting method. Instead of analyzing the image in the time domain as in previous methods, their method uses Fourier analysis to analyze the data in the frequency domain, which simplifies the experimental setup because matching two signals by phase requires very precise time synchronization, which is not needed when identifying frequencies. Their method can obtain a more physically correct result and is robust to noise, at the expense of requiring many images.

Later, Qian et al. [20] incorporate compressive sensing theory, which provides a framework for reconstructing sparse signals with much fewer measurements than the signal dimension, to the frequency-based environment matting. Compared to existing methods for environment matting, their method achieves higher performance on both synthetic and real data, but requires a much smaller number of images.

Transparent object matting network (TOM-Net) [4] 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 thesis research.

Chapter 3

Formulation

In this chapter, we formulate the super-resolution and the environment matting that we use in our work.

3.1 Super-Resolution

Low-resolution images can be seen as a degradation of high-resolution images as shown in Fig. 3.1. In general, HR images and LR images are linked by this model:

$$I_{LR} = (I_{HR} \otimes k) \downarrow_s + n, \tag{3.1}$$

where $\otimes k$ 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 low-resolution images from the ground truth.

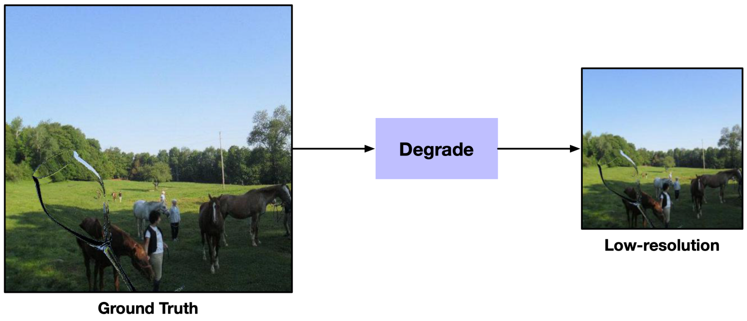

Figure 3.1: Degradation model.

3.2 Environment Matting

Following the work of [6] and [4], 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.[30] 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 [4] , the transparent environment matting problem can be modelled as follows:

$$\mathcal{O} = (1 - I_{mask})I_{ref} + I_{mask}I_{rho} \cdot \mathcal{S}(I_{ref}, \mathcal{G}(I_{flow})). \tag{3.2}$$

In this model, $\mathcal{O}$ denotes the composited image, $I_{mask}, I_{rho}, I_{ref}$ denotes 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.

$\mathcal{S}()$ 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 $\mathcal{G}()$ 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 $\mathcal{G}()$ is a flow-field grid generator that first generates a two-dimensional base grid that has values from the left to the right and 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 $\mathcal{S}()$.

From Eq. 3.2, 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. 1.1. 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.

Chapter 4

Proposed Method

In this chapter, we propose a new deep learning method called Enhanced Transparent Object Matting Network (ETOM-Net), which consists of two parts. The first part, called the main phase, adopts an encoder-decoder structure with multiple scales that takes a low-resolution image of a transparent object as input and extracts a high-resolution environment matte with a pixelwise mask, a refraction flow, and an attenuation map as output. The second part is called the refinement phase using residual learning, which takes the same low-resolution image and the main phase's output as input to predict a sharper and more accurate environment matte.

4.1 Architecture

The main phase of our proposed method ETOM-Net is shown in Fig. 4.1. Similar to [4] and [21], it contains an encoder-decoder structure with a shared encoding process and three independent decoding processes corresponding to the three output environment mattes.

In this structure, we use six encoders and eighteen decoders. 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, two batch normalization layers and two ReLU activation layers, forming a factor of 64 for downsampling. Each decoder has one convolutional layer,

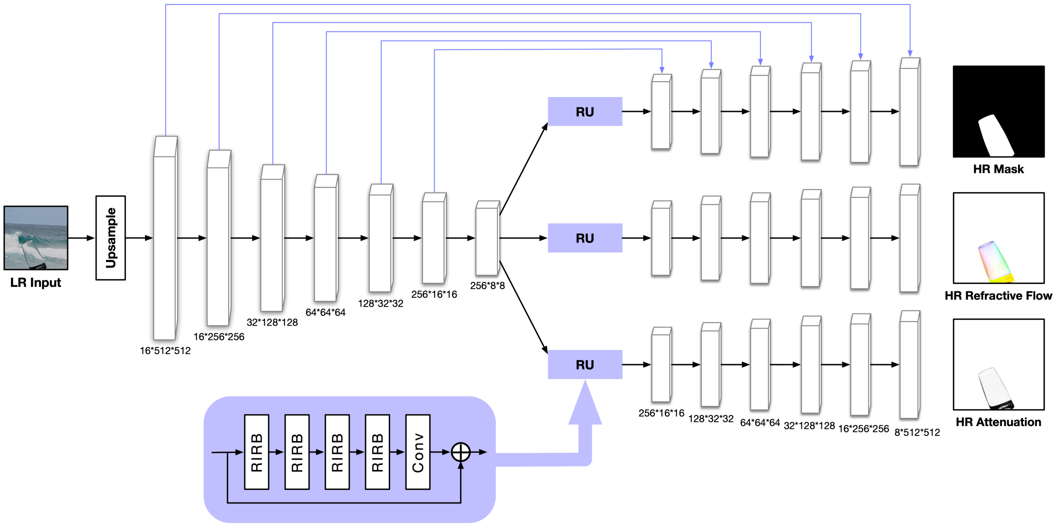

Figure 4.1: ETOM-Net main phase.

one batch normalization 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:

$$\mathcal{O}_{E,i} = E_i(O_{E,i-1}), \tag{4.1}$$

and the output of each decoder can be formulated as:

$$\mathcal{O}_{D,i,j} = D_{i,j}(\sum_{j=0}^{2} O_{D,i-1,j} + O_E), \tag{4.2}$$

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 Restoration Unit (RU)s 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. 4.2 shows the structure of the restoration unit. Each RU consists of four Residual In Residual Block (RIRB), which is inspired by the work of [5], and a convolutional layer, with each RIRB stacked

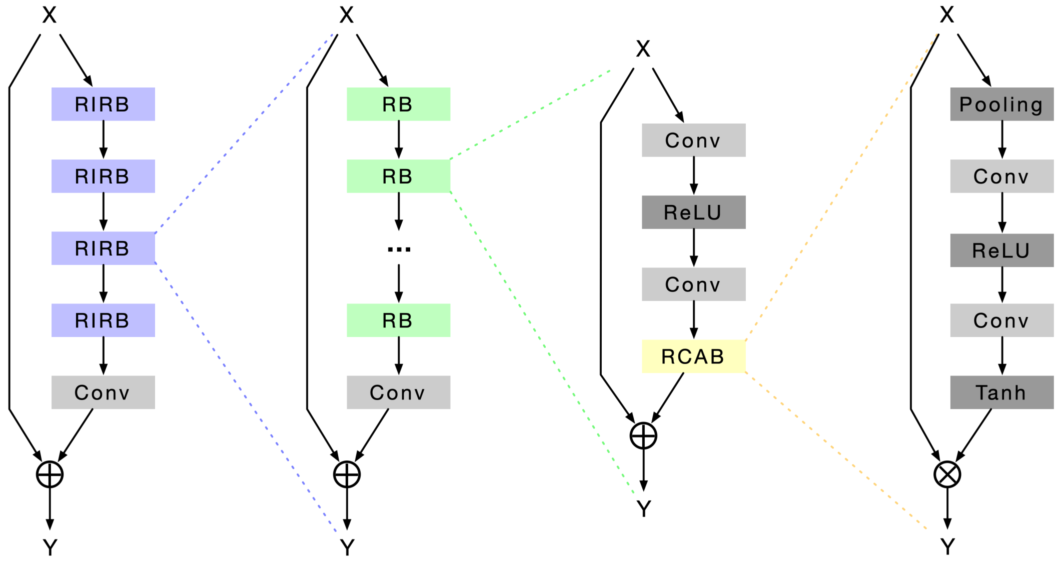

Figure 4.2: Structure of RU. ⊕ denotes the element-wise addition, ⊗ denotes the element-wise product, and Pooling denotes global average pooling.

with ten Residual Block (RB) and one convolutional layer. The output of the RU can be formulated as:

$$\mathcal{O}_{RU} = Conv(RIRB_3(\cdots(RIRB_0(I_{RU})))) + I_{RU}, \tag{4.3}$$

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:

$$\mathcal{O}_{RIRB,i} = Conv(RB_9(\cdots(RB_0(I_{RIRB,i})))) + I_{RIRB,i}, \tag{4.4}$$

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) [28] 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 RB_i can be represented as:

$$\mathcal{O}_{RB,i} = RCAB(Conv(ReLU(Conv(I_{RB,i})))) + I_{RB,i}, \tag{4.5}$$

and the output of RCAB can be formulated as:

$$\mathcal{O}_{RCAB} = Tanh(Conv(ReLU(Conv(Pooling(I_{RCAB}))))) * I_{RCAB}, \tag{4.6}$$

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 [4], 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*64*64 and ends with a size of 8*512*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 ×2, and can be extended to ×3 and ×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. 4.3, the refinement phase takes the low-resolution input, and three output environment mattes from the main phase as input, and then the input tensor is passed through several downsampling blocks, five RB and several upsampling blocks to form the output mattes. Each RB consists of two convolutional layers, two batch normalization 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:

$$\mathcal{O}_{refine} = Conv(Up(RB_4(\cdots(RB_0(Down(I_{mask} + I_{flow} + I_{rho} + I_{lr})))))), \tag{4.7}$$

and the output of RB_i can be formulated as:

$$\mathcal{O}_{RB,i} = BN(Conv(ReLU(BN(Conv(I_{RB,i}))))) + I_{RB,i}, \tag{4.8}$$

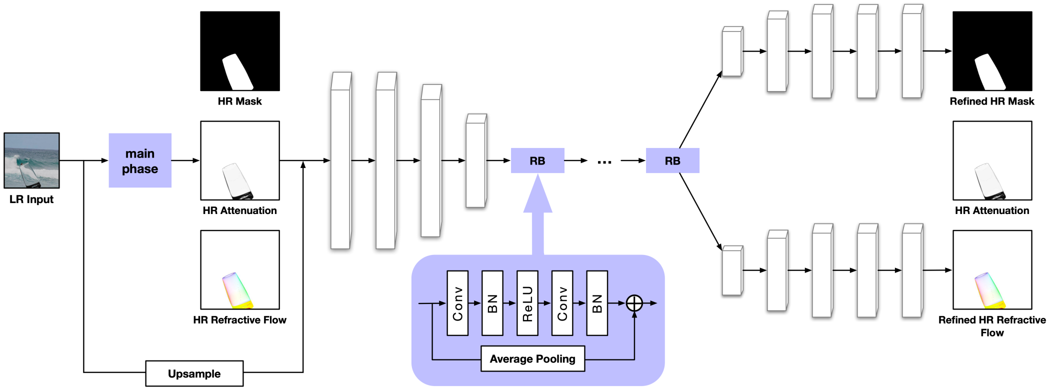

Figure 4.3: ETOM-Net refinement phase.

where O_{refine} denotes the output of the refinement phase, Up denotes the upsampling process, $Down$ denotes the downsampling process, and BN denotes a batch normalization operation. $I_{mask}, I_{flow}, I_{rho}$, and I_{lr} denote the inputs of the refinement phase, including a pixelwise mask, a refractive flow, an attenuation map, and a low-resolution input, respectively.

4.2 Loss Function

In this section, we explain the two loss functions used in the main phase and in the refinement phase.

4.2.1 Main Phase

The loss function $\mathcal{L}_{main}$ of the main phase is divided into four parts similar to [4]: a pixelwise mask loss $\mathcal{L}_{mask}$, a refractive flow field loss $\mathcal{L}_{flow}$, an attenuation loss $\mathcal{L}_{rho}$ and a reconstruction loss $\mathcal{L}_{rec}$. The loss function of the main phase can then be denoted as

$$\mathcal{L}_{main} = \lambda_1 \mathcal{L}_{mask} + \lambda_2 \mathcal{L}_{flow} + \lambda_3 \mathcal{L}_{rho} + \lambda_4 \mathcal{L}_{rec}, \tag{4.9}$$

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 $\mathcal{L}_{mask}$ using the cross-entropy loss

$$\begin{aligned}\mathcal{L}_{mask} &= Mean\left[-\log\left(\frac{\exp(P_{ij}[C_{ij}])}{\sum_{k=0}^{1}\exp(P_{ij}[k])}\right)\right] \\ &= Mean\left[\left(-P_{ij}[C_{ij}] + \log\left(\sum_{k=0}^{1}\exp(P_{ij}[k])\right)\right)\right],\end{aligned} \tag{4.10}$$

where $P_{ij} = (P_{fore}, P_{back})$ 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

$$\begin{aligned}\mathcal{L}_{flow} &= Mean\left[\left\|F - \tilde{F}\right\|_F\right] \\ &= Mean\left[\sqrt{\left(F_{ij}^x - \tilde{F}_{ij}^x\right)^2 + \left(F_{ij}^y - \tilde{F}_{ij}^y\right)^2}\right],\end{aligned} \tag{4.11}$$

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

$$\mathcal{L}_{rho} = Mean\left[\left(A_{ij} - \tilde{A}_{ij}\right)^2\right], \tag{4.12}$$

where A_{ij} is the output attenuation map at (i, j), and $\tilde{A}_{ij}$ 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 high-resolution ground-truth background and compare it to the ground truth high-resolution input image. As with the attenuation map loss, we use the MSE loss

$$\mathcal{L}_{rec} = Mean\left[\left(V_{ij} - \tilde{V}_{ij}\right)^2\right], \tag{4.13}$$

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

4.2.2 Refinement phase

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

$$\mathcal{L}_{refine} = \lambda_1\mathcal{L}_{mask} + \lambda_2\mathcal{L}_{flow}, \tag{4.14}$$

where λ_1, λ_2 are the weights of each component of the loss, and the $\mathcal{L}_{mask}$ and the $\mathcal{L}_{flow}$ are the same as in the main phase.

4.3 Comparison

Here, we compare the similarities and differences between TOM-Net[4] and our proposed method ETOM-Net. Fig. 4.4 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 low-resolution images as input and predicts high-resolution 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. 4.2.

In the refinement phase, both methods use residual learning to refine the mattes predicted by the main phase. However, our refinement phase takes upsampled low-resolution 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.

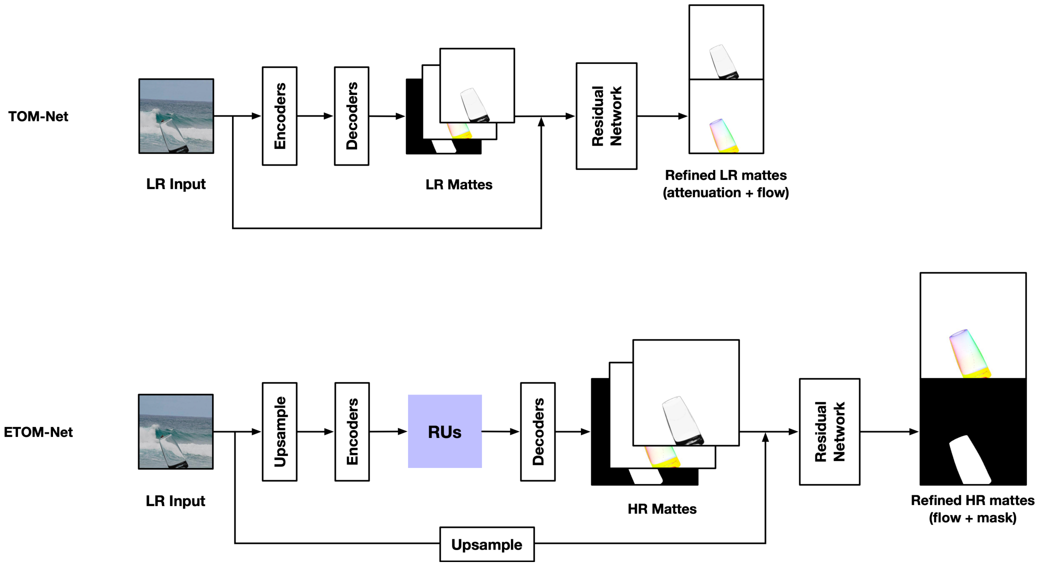

Figure 4.4: A brief comparison between TOM-Net and ETOM-Net.

Chapter 5

Experimental Results

In this chapter, we first present the dataset used for training and evaluation and the details of implementing our method. Then, we present the experimental results and some analysis of our method.

5.1 Dataset

Chen et al.[4] 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.

To facilitate future research, we created a high-resolution dataset with each image of size 1024 × 1024 pixels called ETOM-Synthetic. The background images were randomly sampled from the high-resolution training data of the DIV2K dataset, and then we crop them to 1024 x 1024 as needed.

For transparent objects, we used Blender to generate 3D models. Since we needed random shapes and many models, we used Blender's internal scripting capabilities and a script we wrote to automate this. As a result, 10,000 high-quality 3D models with different shapes were generated. A research-oriented rendering system called Mitsuba 2 was used to render the 3D models onto the background. We rendered them with random rotation and scaling and placed them in a random position in the background. Each background was paired with several different 3D models, and for the super-resolution environment matting, we also rendered corresponding low-resolution images as input.

To generate the ground truth environment matte, we used Mitsuba 2 as well. Similar to rendering 3D models to background images, we replaced the background with a light emitter to generate the attenuation map and turned the 3D model into a light emitter with no background to generate the segmentation mask. For the refractive flow, inspired by [4], we first generated a set of high-resolution gray code patterns with ten horizontal and ten vertical directions for a total of 20 images, and then we rendered the 3d models in front of them in sequence to generate the corresponding refractive flows. In the end, a high-resolution dataset with 60,000 samples was generated.

5.2 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. 4.9 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. 4.14, 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

Table 5.1: Quantitative results comparison. We use mean square error (MSE) to evaluate the reconstructed image and attenuation, intersection over union (IoU) to evaluate the segmentation mask, and endpoint error (EPE) to evaluate the refractive flow field. For MSE and EPE, the lower, the better, but for IoU, the higher, the better.

	Rec↓	Attenuation↓	Flow↓	Mask↑
Baseline	0.696	1.426	2.569	0.102
TOM-Net	0.220	0.252	**1.500**	0.964
SR+TOM-Net	0.278	0.340	1.658	0.884
ETOM-Net: main	0.201	0.203	1.579	0.958
ETOM-Net: refine	**0.192**	**0.203**	1.516	**0.968**

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 a high-resolution environment matte using a single low-resolution input image in a single pass.

Using the Adam optimizer, the training process took four days for the main phase and five days for the refinement phase on an Nvidia 2080Ti GPU. The best model for each phase is selected based on the validation results of each epoch using early stopping. The code for ETOM-Net and rendering can be found on GitHub: `https://github.com/1asso/ETOM-Net` and `https://github.com/1asso/Rendering`.

5.3 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. 5.1. Here, we use the EDRN[5] to achieve super-resolution because it employs

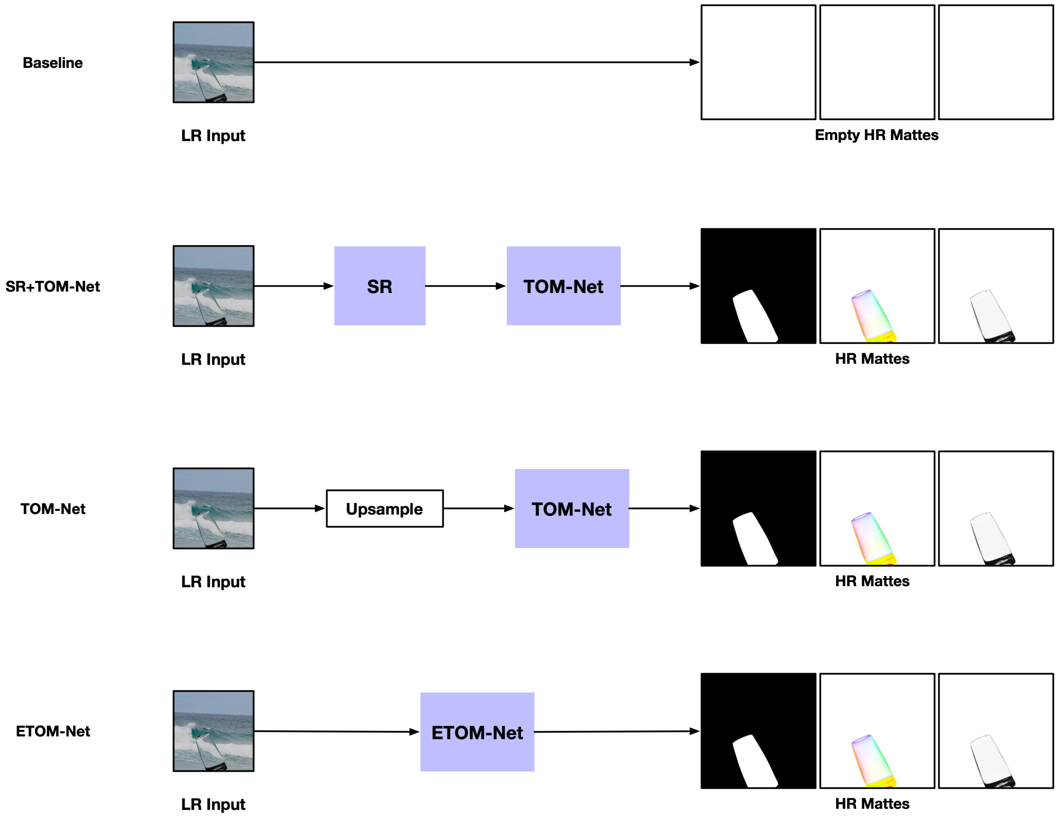

Figure 5.1: The models we used in this experiment. Note that the HR refractive flows are converted to HSV (meaning Hue, Saturation, Value) images, in order to look more intuitive. Therefore, an empty refractive flow $(0, 0, 100\%)$ looks white.

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 5.1. 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. 5.2 presents some qualitative results of our ETOM-Net compared to 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. See Fig. A.2, A.3, A.4 and A.5 for more results.

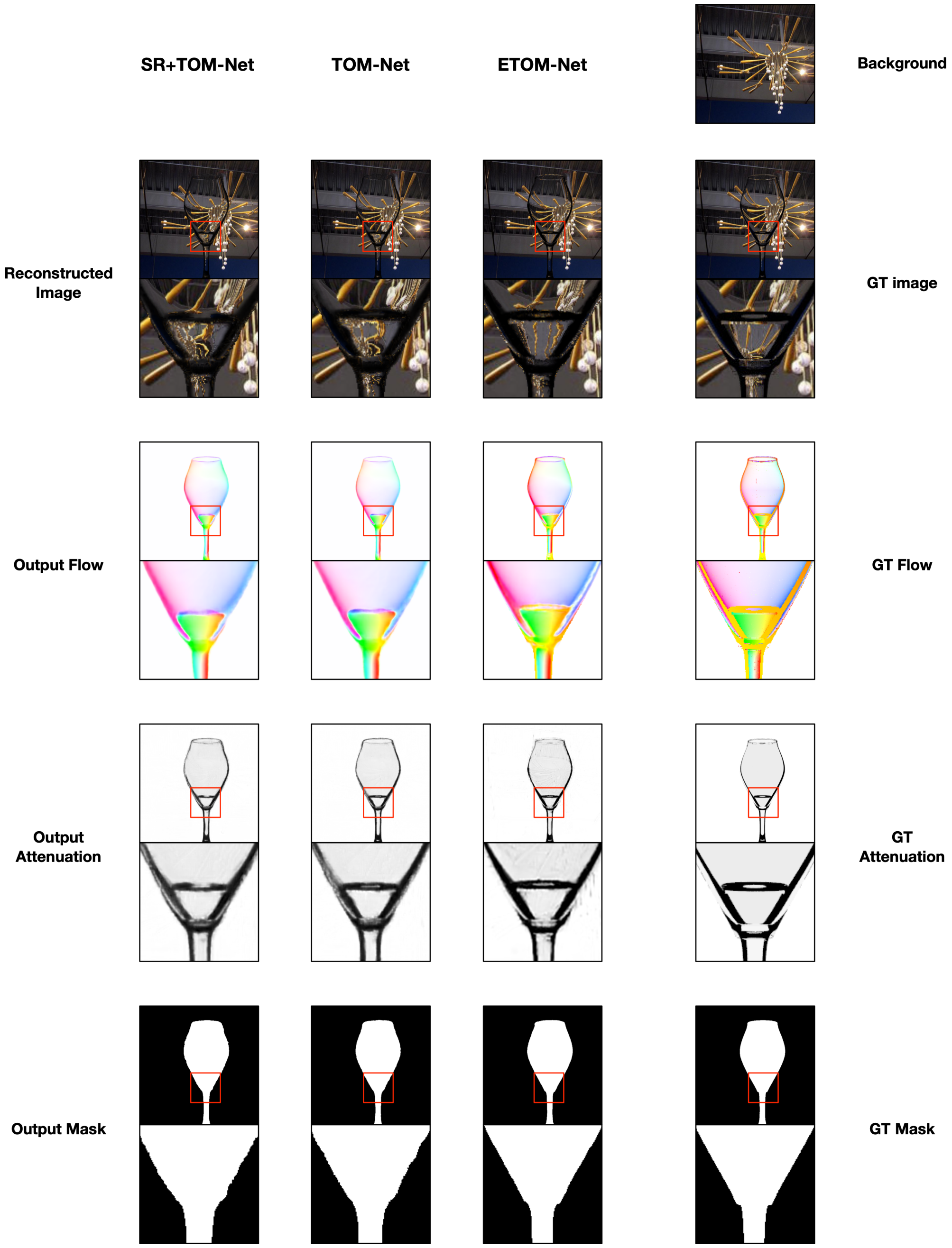

Figure 5.2: Qualitative results comparison. The first three columns are the results from SR+TOM-Net, TOM-Net and ETOM-Net, respectively. The last column are the background and ground truth. The input image for each model is a degraded ground truth image.

5.4 Ablation Study

In order to understand the effectiveness of each component in our ETOM-Net, we created several variants of ETOM-Net and analyzed them quantitatively. Here, we focus on the RU, the object mask loss ($\mathcal{L}_{mask}$) and the attenuation loss ($\mathcal{L}_{rho}$) 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 $\mathcal{L}_{mask}$ in the refinement phase to create model refine - ($\mathcal{L}_{mask}$), and include $\mathcal{L}_{rho}$ to create model refine + ($\mathcal{L}_{rho}$). 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 5.2.

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 RIRB inside the RU does have an impact on performance.

For the refinement phase, removing $\mathcal{L}_{mask}$ or adding $\mathcal{L}_{rho}$ to the loss function leads to a decrease in performance, and although the presence of $\mathcal{L}_{rho}$ 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. 5.4 and Fig. A.1 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. 5.3, 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. 5.3 (b), we can see that the GPU cost is positively correlated with the number of RIRBs. We used a

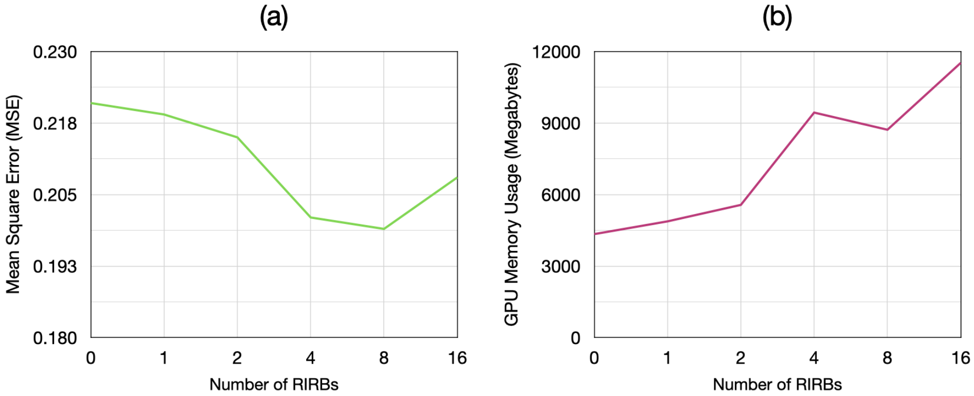

Figure 5.3: Analysis of the effect of the number of RIRBs in each RU on the results

Table 5.2: 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 - ($\mathcal{L}_{mask}$)	0.204	0.206	1.553	0.957
refine + ($\mathcal{L}_{rho}$)	0.211	**0.202**	1.605	0.957
main + refine	**0.192**	0.203	**1.516**	**0.968**

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. 5.3 (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.

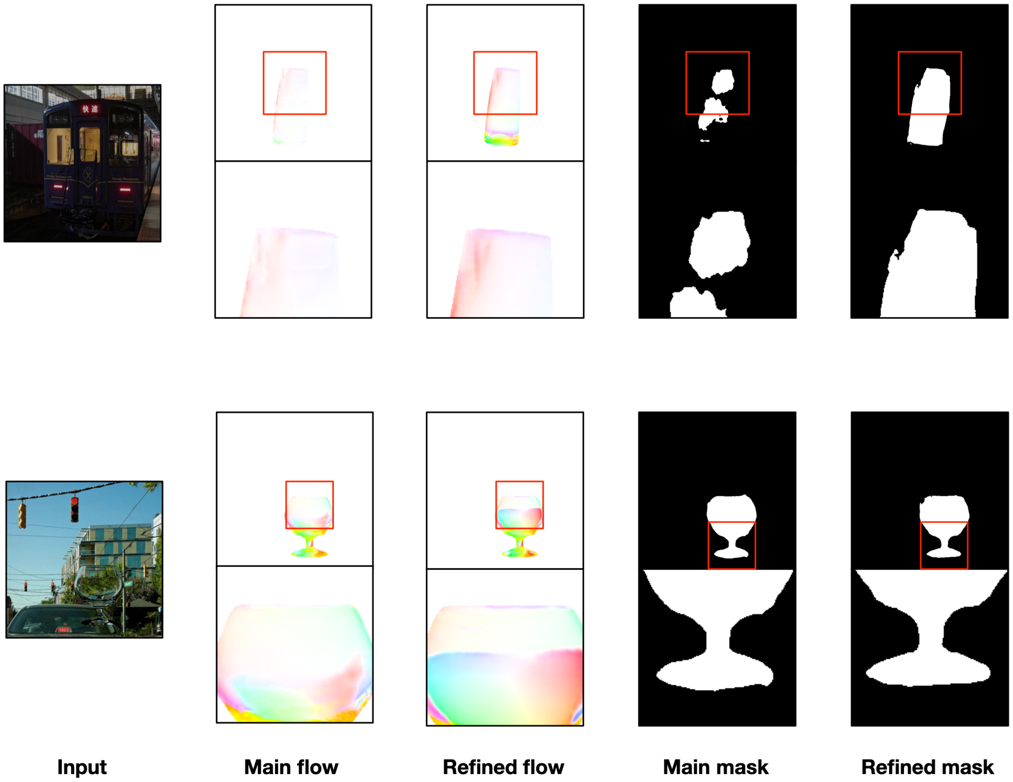

Figure 5.4: Qualitative comparison between the output of the main phase and the refinement phase.

Chapter 6

Conclusion

In this thesis, 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 mattes, 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 mattes. 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. In addition to the ETOM-Net, we create a high-resolution synthetic dataset for super-resolution environment matting called ETOM-Synthetic.

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 we will explore in future work.

References

[1] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *Procedings of the British Machine Vision Conference 2012*, 2012. DOI: 10.5244/c.26.135.

[2] R. Brinkmann, *The art and science of digital compositing*. Elsevier Morgan Kaufmann, 2011.

[3] 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.*, DOI: 10.1109/cvpr.2004.1315043.

[4] 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. DOI: 10.1109/cvpr.2018.00962.

[5] G. Cheng, A. Matsune, Q. Li, L. Zhu, H. Zang, and S. Zhan, "Encoder-decoder residual network for real super-resolution," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. DOI: 10.1109/cvprw.2019.00270.

[6] 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. DOI: 10.1145/344779.344844.

[7] 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. DOI: 10.1109/tpami.2015.2439281.

[8] 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. DOI: 10.1007/978-3-319-46475-6_25.

[9] 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. DOI: 10.1109/cvpr.2007.383001.

[10] R. Fattal, "Image upsampling via imposed edge statistics," *ACM SIGGRAPH 2007 papers on - SIGGRAPH 07*, 2007. DOI: 10.1145/1275808.1276496.

[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. DOI: 10.1109/38.988747.

[12] W. Freeman and E. Pasztor, "Learning low-level vision," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999. DOI: 10.1109/iccv.1999.790414.

[13] 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. DOI: 10.1109/cvpr.2016.90.

[14] 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. DOI: 10.1109/cvpr.2016.182.

[15] ——, "Deeply-recursive convolutional network for image super-resolution," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. DOI: 10.1109/cvpr.2016.181.

[16] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and et al., "Photo-realistic single image super-resolution using a generative adversarial network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. DOI: 10.1109/cvpr.2017.19.

[17] 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. DOI: 10.1109/cvprw.2017.151.

[18] Z. Lin and H.-Y. Shum, "Fundamental limits of reconstruction-based superresolution algorithms under local translation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 83–97, 2004. DOI: 10.1109/tpami.2004.1261081.

[19] P. Peers and P. Dutré, "Wavelet environment matting," *Eurographics workshop on Rendering*, 2003.

[20] Y. Qian, M. Gong, and Y.-H. Yang, "Frequency-based environment matting by compressive sensing," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. DOI: 10.1109/iccv.2015.403.

[21] J. Shi, Y. Dong, H. Su, and S. Yu, “Learning non-lambertian object intrinsics across shapenet categories,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5844–5853, 2017.

[22] 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. DOI: 10.1109/cvpr.2008.4587659.

[23] Y.-W. Tai, 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. DOI: 10.1109/cvpr.2010.5539933.

[24] 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. DOI: 10.1109/cvpr.2017.298.

[25] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman, “Image-based environment matting,” *ACM SIGGRAPH 2002 conference abstracts and applications on - SIGGRAPH 02*, 2002. DOI: 10.1145/1242073.1242211.

[26] 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. DOI: 10.1109/cvpr.2008.4587647.

[27] 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. DOI: 10.1007/978-3-642-27413-8_47.

[28] 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. DOI: 10.1007/978-3-030-01234-2_18.

[29] J. Zhu and Y.-H. Yang, “Frequency-based environment matting,” *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, DOI: 10.1109/pccga.2004.1348371.

[30] 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. DOI: 10.1145/311535.311558.

Appendix A

More Results

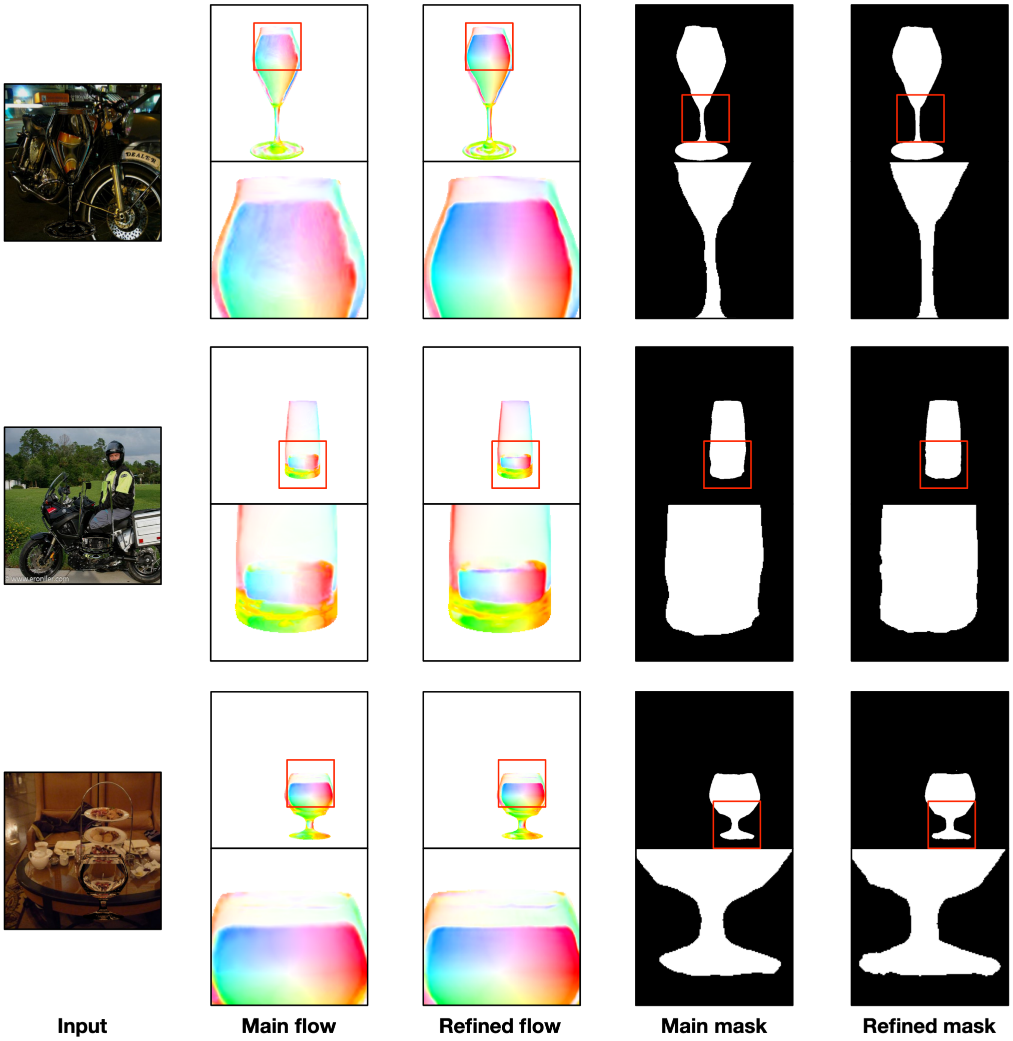

Figure A.1: Qualitative results comparison in ablation study.

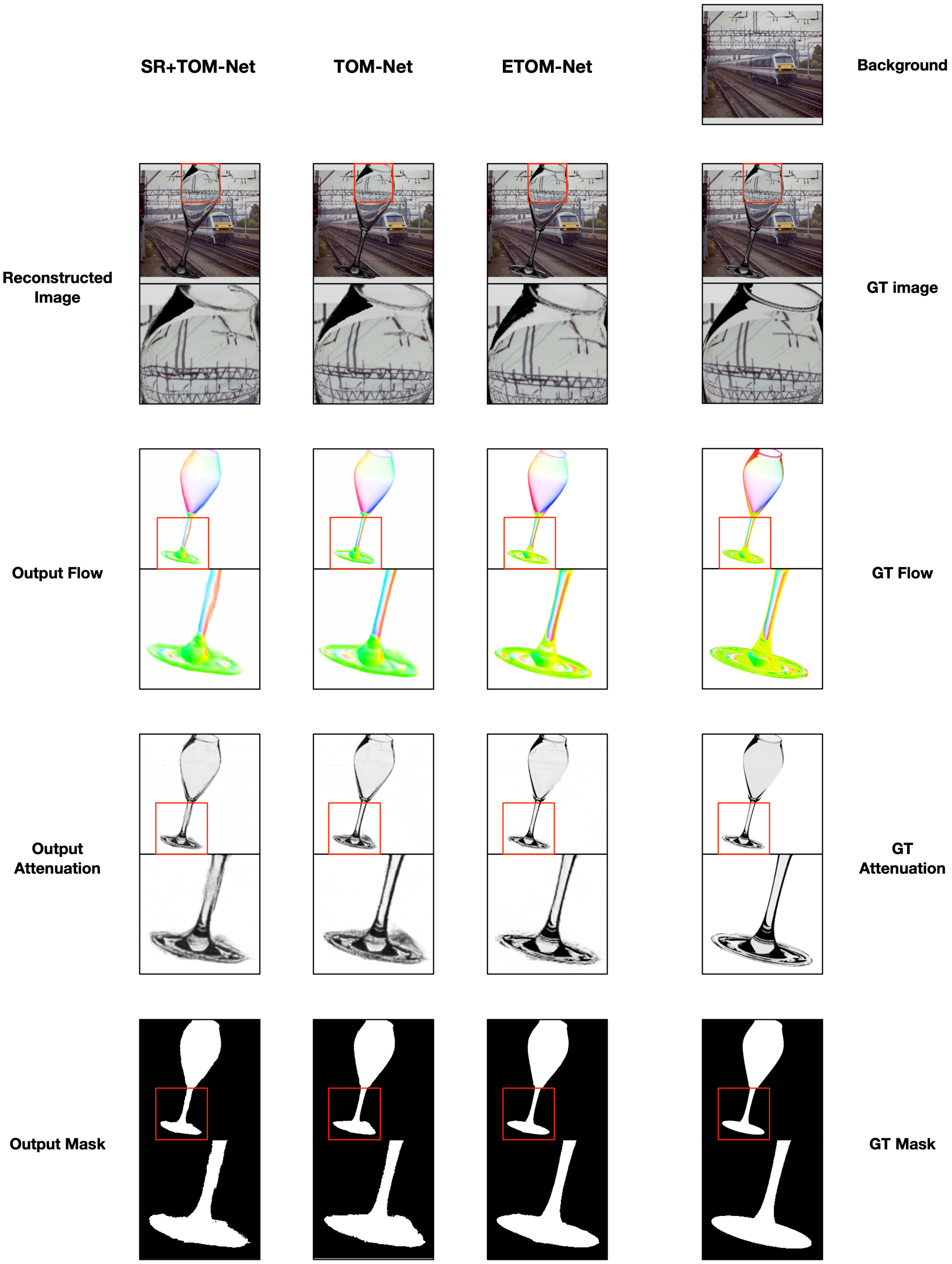

Figure A.2: Qualitative results comparison (a).

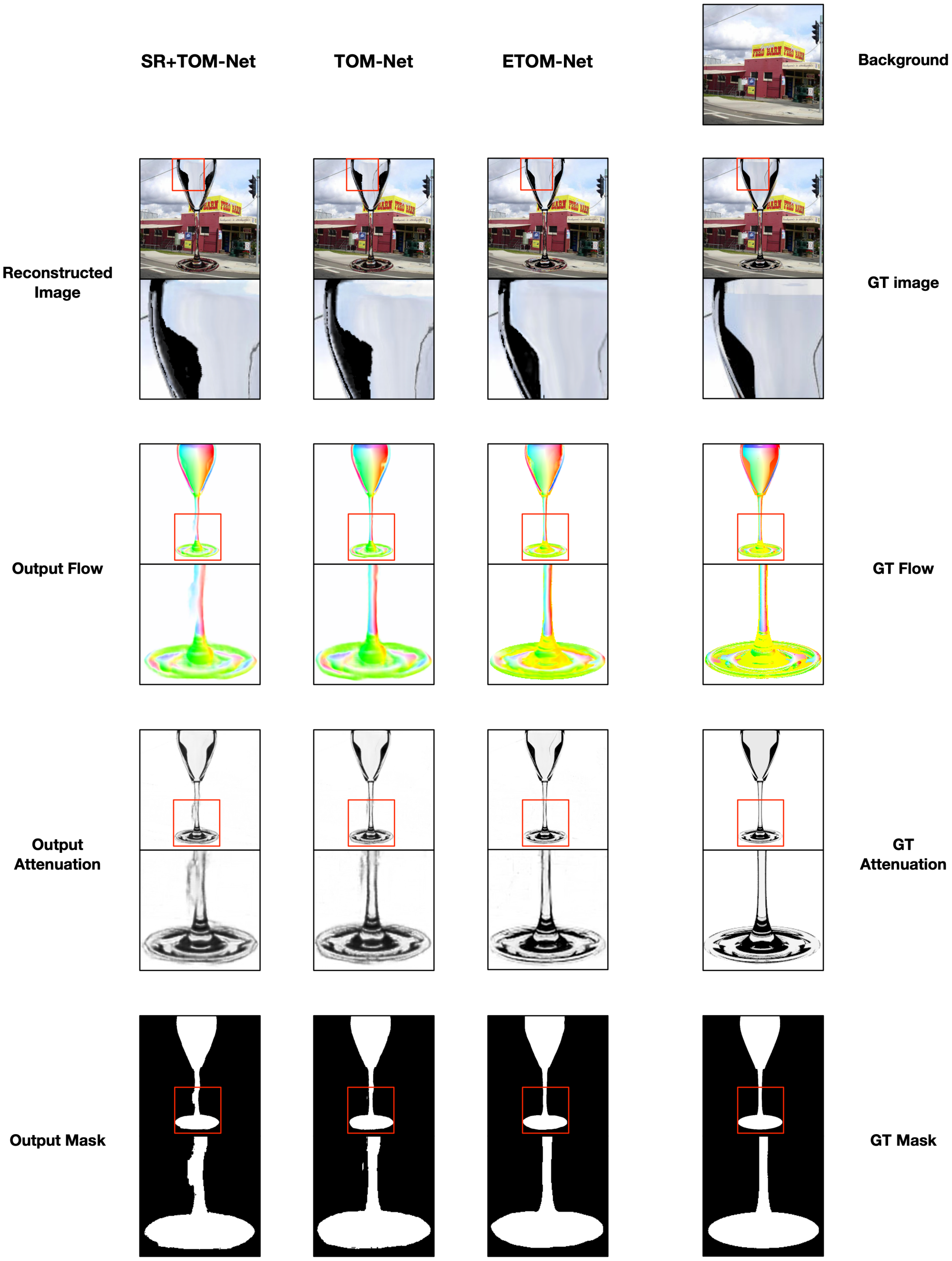

Figure A.3: Qualitative results comparison (b).

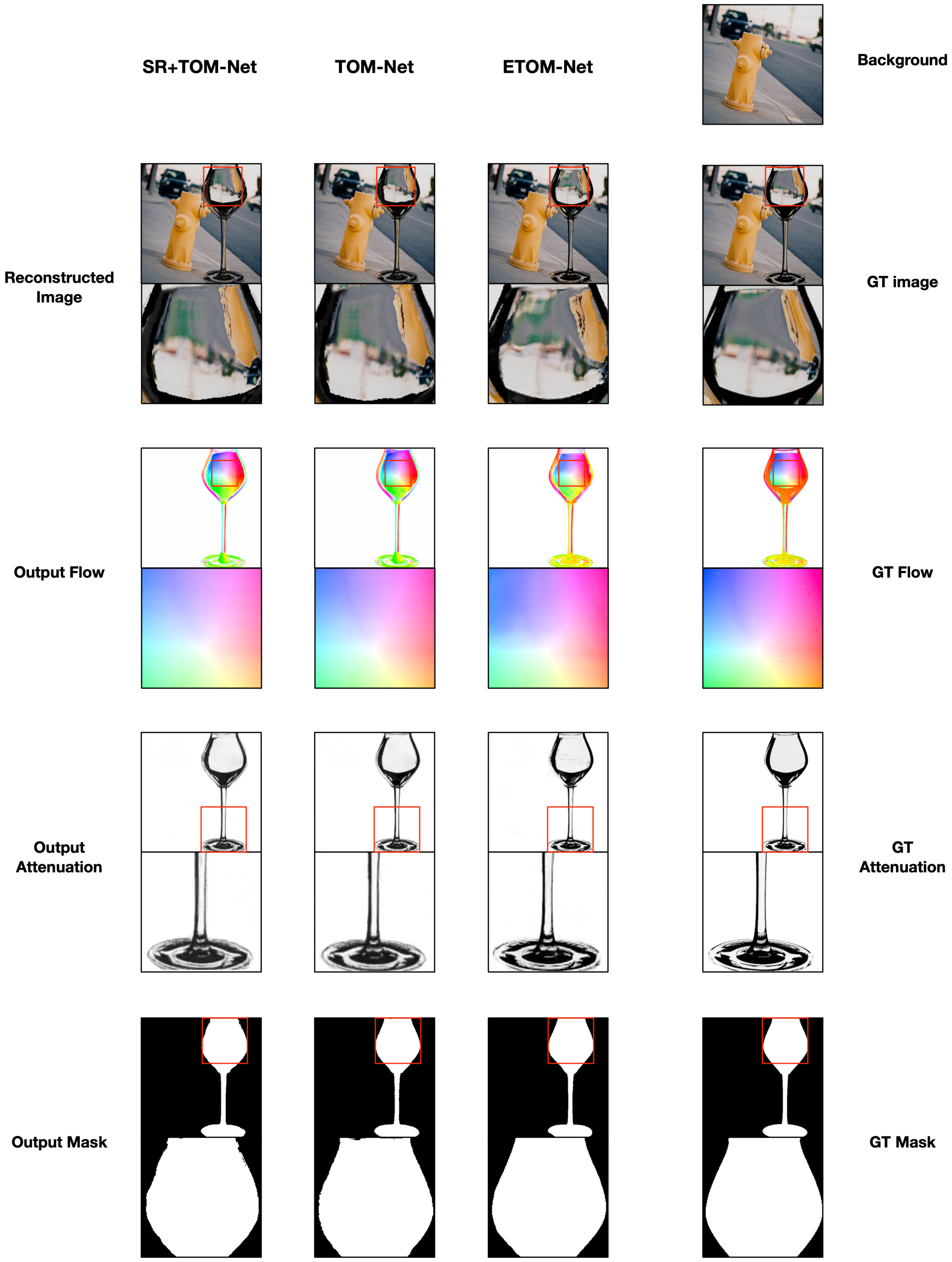

Figure A.4: Qualitative results comparison (c).

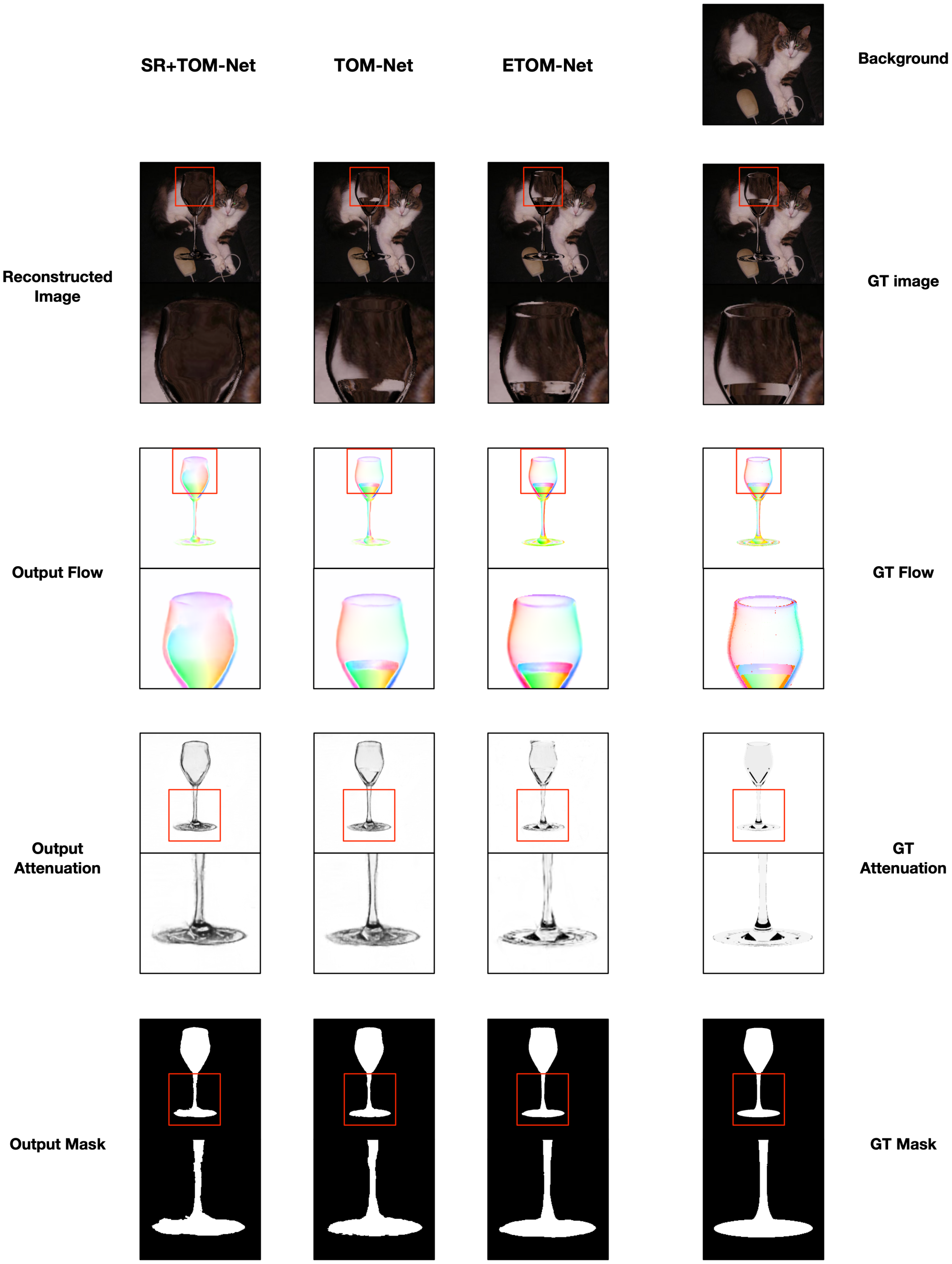

Figure A.5: Qualitative results comparison (d).