Ultrasound Resolution Image/Video Matting with Spatio-Temporal Sparsity

Yanan Sun  
HKUST  
now.syn@gmail.com

Chi-Keung Tang  
HKUST  
cktang@cs.ust.hk

Yu-Wing Tai  
HKUST  
yuwing@gmail.com

Abstract

Commodity ultrahigh definition (UHD) displays are becoming more affordable which demand imaging in ultrahigh resolution (UHR). This paper proposes SparseMat, a computationally efficient approach for UHR image/video matting. Note that it is infeasible to directly process UHR images at full resolution in one shot using existing matting algorithms without running out of memory on consumer-level computational platforms, e.g., Nvidia 1080Ti with 11G memory, while patch-based approaches can introduce unsightly artifacts due to patch partitioning. Instead, our method resorts to spatial and temporal sparsity for addressing general UHR matting. When processing videos, huge computation redundancy can be reduced by exploiting spatial and temporal sparsity. In this paper, we show how to effectively detect spatio-temporal sparsity, which serves as a gate to activate input pixels for the matting model. Under the guidance of such sparsity, our method with sparse high-resolution module (SHM) can avoid patch-based inference while memory efficient for full-resolution matte refinement. Extensive experiments demonstrate that SparseMat can effectively and efficiently generate high-quality alpha matte for UHR images and videos at the original high resolution in a single pass. Project page is in https://github.com/nowsyn/SparseMat.git.

1. Introduction

Ultrahigh resolution (UHR) matting is an important problem [36, 49], and with increasing demand due to the fast advent and accessibility of commodity ultrahigh definition displays in real-world applications, such as gaming, TV/movie post-production, and image/video editing, UHR matting becomes ever relevant. However, modern consumer-level GPU and mobile devices still have limited hardware resources. Despite good technical contributions, guided filters and patch-based techniques (Figure 1) are not applicable, when unsightly blurry and seams artifacts are unacceptable in UHR imaging.

Matting is a primary technique for image/video editing and plays an important role in many applications. The goal of matting is to extract a detailed alpha matte of the foreground object from a given image/video. As known, matting is an ill-posed problem defined as Equation 1 with the given image I, foreground F, background B and alpha \( \alpha \in [0, 1] \) to be extracted:

\[
I = \alpha F + (1 - \alpha) B.
\] (1)

Most of the existing state-of-the-art matting methods [28, 33, 48] take the whole image as input in a forward pass, and thus the resolution they can handle is bounded by available memory. Given limited memory, to process UHR images, a straightforward approach is to first process the downsampled input image which will inevitably lead to blurry artifacts. Thus, super-resolution methods, such as guided filter (GF) [21] or deep guided filter (DGF) [47], have been proposed to recover missing details. However, guided filter or deep guided filter easily produces fuzzy artifacts when processing complex hairy structures as shown in Figure 1-(a). Patch-based inference [23, 35] is another plausible strategy. However, small patch can cause artifacts due to insufficient global context and inconsistent local context as shown in Figure 1-(b). On the other hand, using large patch (e.g., 2048) with large overlap produces defective alpha matte with missing details or blurry artifacts in long hair region due to the lack of long-range dependency in UHR images as shown Figure 1-(b), not to mention that heavy computation and memory overhead are introduced with increasing patch size making some methods cannot even run, as shown in Figure 1-(c). In conclusion, super resolving based on (deep) guided filter or patch-based inference are not ideal choices for handling UHR matting.

In this paper we propose a general image/video matting framework SparseMat to address the problem of UHR matting, which is both memory and computation efficient while generating high-quality alpha mattes. The core idea of our method is to skip a large amount of redundant computation on many pixels during processing UHR images or videos.

In general, our method takes the low-resolution prior as input to generate spatio-temporal sparsity, which serves as the gate to activate pixels consumed by a sparse convolution module. Both temporal and spatial information contribute to the sparsity estimation. Specifically, we compute color difference between adjacent frames to obtain the temporal
Figure 1. (a) Blurry artifacts when UHR images are not matted in the original full resolution: guided filter (GF), deep guided filter (RVM [36]), small patch replacement method (BGMv2 [35]). The low-resolution alpha matte obtained by the self-trained low-resolution prior network (LPN) is for reference here. Our sparse high-resolution module (SHM) produces high-quality alpha matte for UHR matting. (b) Seam artifacts of patch-based inference from FBA [13] under different patch sizes with a fixed ratio (1/8) of overlap. Full-resolution result is ours. (c) Memory consumption and computation (GMACs) with different patch sizes. When the patch size exceeds 2K, some methods cannot even run on Nvidia 1080Ti with 11G memory.

sparsity. For the spatial sparsity, we can derive from any lightweight low-resolution matting model such as [28, 36]. The two sparsity maps are combined together to determine the pixels which need high-resolution processing by the sparse module. Unlike super-resolving strategy performed on the whole image, our method with sparse high-resolution module (SHM) can safely skip expensive computations in large solid pixel regions, only paying attention to irregular, sparse and (oftentimes) thin border regions surrounding the object or transitional regions within the object. In contrast to restrictive views of patch-based strategy to local regions, our method takes a more global perspective of the foreground object thus avoiding potential artifacts due to inadequate context consideration. This design allows SparseMat to process an ultrahigh resolution image or video frame in only one shot without suffering any information loss caused by down-sampling or patch partition, which thus produces high-quality alpha matte for ultrahigh resolution images or videos.

Our contributions are summarized below:

1. This is the first work for general UHR image/video matting which enables full-resolution inference in one shot without running out of memory, thus eliminating the need of patch partitioning and patch artifacts.
2. We show how to obtain accurate spatio-temporal sparsity for general UHR matting, which has never been adequately discussed in previous works. In addition, this is the first work that proposes to apply sparse convolution network to skip unnecessary computations in dealing with UHR matting.
3. We conduct extensive experiments in multiple popular image/video matting datasets, including Adobe Image Matting Dataset [48], VideoMatte240K [42] and our self-collected UHR matting dataset, and provide promising qualitative results, which demonstrate the superiority of our SparseMat in dealing with general image/video matting.

2. Related Work

2.1. Image/Video Matting


Video matting focuses on the temporal coherence of the predicted alpha matte. Previous methods [9, 26, 29, 40] base on optical flow to align information in different frames for consistency. Recently, deep learning approaches attract more attention in tackling video matting for their superior performance. For example, subsequent methods [8, 32, 53] utilize non-local matting Laplacian to encode coherency. Deep Video Matting [45] develops trimap propagation module and spatio-temporal feature aggregation module to simultaneously eliminate dense trimaps while preserving temporal coherence.
2.2. High-Resolution Matting

Handling very high resolutions has been a long-standing issue in the above conventional image matting methods. Traditional methods adopt shared sampling [14] to save computation, or employ large kernel [20] to collect enough color samples when processing high-resolution images. While real-world UHD applications require UHR image matting, existing state-of-the-art works either cannot run at full resolution in one shot, or lend themselves to patch-based inference or adopt super-resolution to produce high-resolution mattes given limited memory. However, patch-based inference often introduces inconsistency artifacts due to inadequate context information. In [23] contextual information is aggregated via a context encoder, which may easily fail to align alpha mattes among patches due to lack of long-range context. In [49] contextual dependency is computed using a cross-patch contextual module (CPC) which is computationally intensive. [35] proposes an efficient small patch replacement strategy based on coarse alphas to tackle high-resolution images. But this method suffers replacement artifacts due to the inconsistent resolution between coarse and refined patches as shown in Figure 1-(a).

The super-resolution approach first processes a down-sampled image and then super-resolves the low-resolution alpha matte to obtain a high-quality alpha matte. In [36], deep guided filtering is proposed taking as input the low-resolution alpha matte with foreground, hidden features and high-resolution image to produce high-resolution alpha. However, deep guided filter is unstable in handling matting which can produce undesirable blurry artifacts as observed in Figure 1-(a).

2.3. Sparsity and Sparse Convolution Network

Sparse convolution (SC) [12, 16] takes only active pixels, which are responsible for generating next layer feature maps, for computing regular convolution. Since the ignorance of inactive pixels, it is popular among the field of dealing with sparse input data, like point cloud.

However, in SC, the inactive pixels within the receptive field of the active input pixels will be activated after convolution, which leads to a rapid growth of active pixels and constrains the depth of network architecture. Therefore, submanifold sparse convolution (SSC) is proposed, in which only pixels which are active in the input feature map can be activated in the output feature map. In doing so, the number of active pixels will not increase after a convolution layer and a deeper sparse convolution network is thus feasible.

Besides SC and SSC operations, activation functions, batch normalization (BN) and deconvolution operation (DC) are also necessary in constructing a sparse network in pixel-wise tasks. Activation functions and BN are restricted to the set of active pixels. In [17] DC is defined as the inverse of the SC convolution. The set of active output pixels in DC is the same as the set of input active pixels to the corresponding SC convolution. The implementation of these sparse operations can be achieved efficiently using look-up table. According [18], SSC requires only 0.6% of the workload of a regular convolution, when 1.3% of the input are active pixels, indicating that significant speed-up can be achieved if the input satisfies high sparsity.

3. SparseMat Framework

In this section, we elaborate our framework SparseMat, illustrated in Figure 2.

3.1. Observations

For UHR image/video matting, we have three observations. First, given an UHR image of a foreground object, most image regions consist of a lot of solid pixels, that is, definite foreground or background pixels. According to the statistics of published matting datasets [41, 48], solid foreground pixels and fractional alpha pixels respectively account for 43.7% and 6.5% on average, indicating that a large amount of computation may be wasted on large solid foreground and background regions. Such wastage is more unforgivable on mobile devices with limited memory, making it infeasible to perform full-resolution inference without out-of-memory problem.

The second observation is that alpha mattes of most foreground objects exhibit a highly consistent distribution. For instance, alpha matte of human can be regarded as the composition of the body region and its boundary. Since the body region is composed of solid pixels, estimating the body region matte is similar to object segmentation, which does not require high-resolution input, where a working resolution at 512p suffices to extract accurate solid body region as shown in Figure 3. On the other hand, boundaries can be hairy, sharp, consist of thin structures, which call for higher resolution demand in learning such complex spectrum in boundary pattern.

The last observation focuses on the consistency between adjacent video frames. Two neighboring frames usually share a lot of color pixels, which indicates that a large amount of computation on the latter frame is unnecessary and thus can be skipped. The aforementioned observations motivate us to build a reasonable spatio-temporal sparsity map, used to guide the model to only focus on valuable pixels and neglect those inessential pixels.

3.2. Spatio-Temporal Sparsity Estimation

Spatial Sparsity. The first two observations motivate us to apply low-resolution predictions as guidance to generate spatial sparsity map. Since it is usually efficient to process low-resolution input image or video frames with delicately designed real-time model, we can obtain low-resolution predictions for the target foreground object at an extremely low cost. Here, we can utilize any lightweight
Experiments on accuracy and efficiency, the dilation kernel size is set at 15. Foreground from background colors in matting. To balance formation of local solid regions is crucial for disentangling pixels of the low-resolution prediction, as neighboring undesirable to directly select sparse index from fractional alpha which are the focus of our next step. Note that it is undetectable to activate pixels for the sparse high-resolution module. The temporal sparsity comes from the color difference of two neighboring frames while the spatial sparsity is dilated and eroded from the output of the low-resolution prior network (LPN). Our high-resolution module applies sparse convolution to skip inactivate pixels while only paying attention to activated pixels, which saves a lot of computations. Some connections are skipped here for clarity.

Temporal Sparsity. When dealing with high-resolution video, we can utilize temporal information to reduce the computation cost further. Specifically, we take the subtracted result between two neighboring frames, and turn the color difference to binary map through clipping the values by a threshold (0.05 used in the paper). This binary map indicates the changed pixels between two adjacent frames for any region including the target object, which serves as temporal sparsity. Dilation and erosion operations are also adopted for covering local context information.

Spatio-Temporal Sparsity. We obtain the final spatio-temporal sparsity map by taking the intersection of spatial and temporal sparsity maps and use it as a gate to activate pixels for our sparse high-resolution module.

3.3. Sparse High-Resolution Module (SHM)

Our sparse high-resolution module (SHM) applies sparse convolution and submanifold sparse convolution operations. After generating a reasonable spatio-temporal sparsity map, we apply the sparse high-resolution module to predict the alpha values for the active pixels to reconstruct the alpha matte at original ultrahigh resolution. For simplicity, we denote $\alpha_l$ as the low-resolution prediction from the low-resolution prior network, and $\alpha_h$ as the high-quality alpha from the high-resolution module. $M$ stands for the spatio-temporal sparsity map while $M_s$ and $M_t$ represents the spatial and temporal sparsity map respectively. Superscript denotes the frame index.

High-Resolution Module. The sparse high-resolution module takes as input the concatenation of high-resolution image and upsampled $\alpha_l$. Before forwarding the high-resolution module, we convert the dense representation of the input into sparse representation using the sparsity map $M$, which is a binary mask where 1 and 0 respectively indicate active and inactive pixels.

Our high-resolution module is a U-Net like matting structure implemented using sparse operations (SC, SSC,
etc), which adopts sparse ResNet18 [22] as backbone to extract three levels of encoded semantic features, and feeds them into the sparse decoder. Before reconstructing features through the decoder, we aggregate global information from the bottleneck feature in the low-resolution prior network to enhance high-resolution feature by applying a masked Squeeze-and-Excitation [24] layer (Masked-SE), which is illustrated in Figure 5.

In detail, Masked-SE first applies global average-pooling operation on the low-resolution bottleneck features masked by \( \alpha_l \) and \( 1 - \alpha_l \) respectively, followed by applying a fc layer to obtain two vectors. Then a fc layer followed by a sigmoid function learns a channel-attention weight from the two vectors, which is then multiplied with the bottleneck feature in the high-resolution module, in order to encode global foreground and background information. After the Masked-SE and the sparse decoder, we finally extract the high-resolution alpha matte \( \alpha_h \).

**Sparse to Dense.** \( \alpha_h \) is first converted into dense representation by scatter operation according to indices of active pixels and the original shape. To differ the dense and sparse output, we use \( \tilde{\alpha}_h \) to denote the sparse predicted alpha matte. Then, a full-resolution high-quality alpha matte \( \alpha_h \) for frame \( i \) can be computed from weighted summation as Equation 2 below. \( \uparrow \) is the upsampling operation.

\[
\alpha^i_h = \left(1 - M \right) \cdot \left(1 - M_s \right) \cdot \alpha^i_{l\uparrow} + \\
\left(1 - M \right) \cdot M_s \cdot \alpha^{i-1}_h + M \cdot \tilde{\alpha}^i_h 
\]

**Adaptation to Higher Resolution.** With our sparse high-resolution module, UHR images can be processed smoothly in a forward pass at full resolution on a Nvidia 1080Ti GPU card with 11G memory storage. Our experiments on 1080Ti GPU card validate that our sparse high-resolution module can handle up to about 4.3M active pixels, making up for about 50% of a given 4K image. According to our statistics, an average of about 0.58M active pixels need to be processed for UHR images. Thus, for images at higher resolution, such as 6K/8K, as long as the portion of active pixels is less than 23%/13% respectively, our SparseMat is still applicable. For images with more than 4.3M active pixels (in 1080Ti), cascading inference can be adopted.

Cascading inference will be triggered when the computation load exceeds the memory limitation, i.e., the number of active pixels is more than 4.3M on 1080Ti GPU card. In summary, we simply apply the sparse high-resolution module in a cascading manner [6] to progressively reduce the number of active pixels. Figure 6 provides an example, in which three resolutions are used: \( 0.25 \times 0.5 \times 1.0 \times \). Without cascading, the number of active pixels is 9.23M, far more than 4.3M, the limitation of processed pixels for 1080Ti. Through cascading, the number of active pixels is relatively reduced to 3.99M at the largest resolution.

**3.4. General UHR Matting**

Note that our method is a general UHR matting pipeline and not limited to any specific class. First, general foreground objects exhibits the three observations we mentioned in Section 3.1. For instance, given animal foreground objects, even though the boundary furry region requires computations in high-resolution module, we can safely skip large external background region and internal foreground region through our spatial sparsity map as shown in Figure 7. Second, our low-resolution prior network is independent of specific matting model architectures. The choice can be any lightweight low-resolution approach, which makes our sparse high-resolution module feasible on general UHR matting.

Figure 5. Structure of Masked-SE. GAP is global average pooling.

Figure 6. Adapt SparseMat to images at higher resolution in cascading manner. In this example, the image resolution is \( 6,800 \times 3,825 \) and we use three scales \( 0.25 \times 0.5 \times 1.0 \) to refine the alpha matte progressively. The number of active pixels (marked in red) is greatly reduced to 3.99M from 9.23M through cascading inference.

Figure 7. Sparsity map of an animal foreground object. Background aside, we can achieve 93.3% sparsity and thus reduce a large amount of computations.
Table 1. Quantitative comparisons of our sparse high-resolution module (SHM) with different low-resolution prior networks on AIM [48], VM [35] and our self-collected UHR matting testing set HHM2K. LPN is our self-trained lightweight human matting model used as the low-resolution prior network.

<table>
<thead>
<tr>
<th>Method</th>
<th>AIM-Human</th>
<th></th>
<th></th>
<th>VM</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAD</td>
<td>MSE</td>
<td>Grad</td>
<td>Conn</td>
<td>MAD</td>
<td>MSE</td>
</tr>
<tr>
<td>MODNet [28] + SHM</td>
<td>25.06</td>
<td>15.32</td>
<td>10.22</td>
<td>21.60</td>
<td>8.13</td>
<td>4.74</td>
</tr>
<tr>
<td>RVM [36] + SHM</td>
<td>17.33</td>
<td>9.89</td>
<td>10.02</td>
<td>17.39</td>
<td>13.77</td>
<td>9.02</td>
</tr>
<tr>
<td>LPN</td>
<td>18.23</td>
<td>11.44</td>
<td>11.33</td>
<td>18.71</td>
<td>8.21</td>
<td>4.38</td>
</tr>
<tr>
<td>LPN + SHM (Ours)</td>
<td>16.47</td>
<td>9.18</td>
<td>9.85</td>
<td>15.38</td>
<td>7.90</td>
<td>4.29</td>
</tr>
</tbody>
</table>

Table 2. Quantitative comparisons of our SparseMat on the full testing set of AIM [48]. We adopt the general matting methods, FBA [13] and SIM [44], with the officially released weight, as the low-resolution prior networks. FBA-512 and SIM-512 denotes the input resolution is 512 for FBA and SIM models.

<table>
<thead>
<tr>
<th>Method</th>
<th>SAD</th>
<th>MSE</th>
<th>Grad</th>
<th>Conn</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBA [13]</td>
<td>26.5</td>
<td>5.3</td>
<td>10.6</td>
<td>21.8</td>
</tr>
<tr>
<td>FBA-512 + SHM</td>
<td>26.2</td>
<td>5.2</td>
<td>10.4</td>
<td>20.7</td>
</tr>
<tr>
<td>SIM [44]</td>
<td>28.0</td>
<td>5.8</td>
<td>10.8</td>
<td>24.8</td>
</tr>
<tr>
<td>SIM-512 + SHM</td>
<td>27.3</td>
<td>5.4</td>
<td>10.5</td>
<td>24.1</td>
</tr>
</tbody>
</table>

Figure 8. Structure of our sparse high-resolution module.

ResNet18 with downsampling stride of 8. In the decoder, we use the inverse version of sparse convolution to reconstruct the features through three levels. Our sparse convolution implementation is based on GitHub:spconv.

During training, we train the sparse high-resolution module for totally 30 epochs with a batch size of 12. The learning rate is set to 0.0001 which decays at a rate of 10 in every 10 epochs. During inference, the low-resolution prior network takes the downsampled input image of 512 while the sparse high-resolution module processes the original high-resolution image. We conduct all the experiments on the Nvidia 1080Ti GPU card.

4.4. Main Results

Results on Human Matting. For the low-resolution prior network, we adopt three choices, MODNet [28], RVM [36] and our self-trained lightweight human matting model denoted as LPN, whose structure is provided in the supplementary materials. All the low-resolution prior networks are trained on HHM50K. We evaluate our method on three human testing datasets, AIM-Human, VM and HHM2K. The evaluation results are tabulated in Table 1. Based on the low-resolution prior information, our sparse high-resolution module promotes the quality of the alpha matte, benefiting from our full-resolution inference pipeline taking use of continuous local and global information on the transitional region.

Figure 9 compares qualitative results on natural images from HHM2K, which shows that our method produces high-quality alpha matte for UHR human images. Human alpha matte typically entails hair structures and sharp boundaries, where such sparse alpha distribution according was shown in SIM [44]. Leveraging such sparsity, our method can gain much efficiency in human matting.

4.1. Datasets

We conduct experiments on self-collected UHR matting datasets and public available image/video matting datasets following [36]. Existing datasets suffer from limited resolution [1,41,43,48,48] (usually no more than 1920p). Thus, in this paper we contribute the first UHR human matting dataset, composed of HHM50K for training and HHM2K for evaluation. HHM50K and HHM2K consist of respectively 50,000 and 2,000 unique UHR images (with an average resolution of 4K) encompassing a wide range of human poses and matting scenarios. More details about the datasets can be found in the supplementary materials.

Besides, we also evaluate our method on public available matting datasets, including Adobe Image Matting dataset [48] (AIM) and VideoMatte240K [35] (VM). AIM is a general matting dataset covering human images. We follow [36] to pick all human images and form a human matting testing set. For simplicity, we denote this subset as AIM-Human. VM is a synthetic human video matting dataset [36], which can be used to validate the model’s capability on processing videos.

4.2. Implementation Details

The structure of the sparse high-resolution module is detailed in Figure 8. The backbone network is a sparse
Figure 9. Qualitative results on HHM2K among our SparseMat and popular matting methods including trimap-free methods, MODNet [28] and RVM [36], and trimap (or mask)-based methods including DIM [48], GCA [33], MG [50], SIM [44] and FBA [13]. We provide the visualization of the spatial sparsity map on the bottom-left of the image. Ours are the results from our sparse high-resolution module based on LPN.

Figure 10. Qualitative results on general UHR matting. In each example, we provide the spatial sparsity map on the bottom-right of the image and the zoom-in results in the right column. Our sparse high-resolution module is capable of processing various foreground objects, which not only resolves the resolution but also eliminates background noise and recovers missing details.

Table 3. Computation cutoff under different sparsity settings. SSM and TSM denote spatial sparsity map and temporal sparsity map respectively. * denotes approximate GMACs.

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>GMACs</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ 0% 90.5% 92.3%</td>
<td>✓ 1589.76 151.02 122.41</td>
</tr>
</tbody>
</table>

Results on General Matting. SparseMat is applicable to general matting. To validate, we replace the low-resolution prior network with recent state-of-the-art general matting approaches, i.e., FBA [13] and SIM [44], then train and evaluate the sparse high-resolution branch on AIM dataset. The evaluation results are tabulated in Table 2. See captions for more details. Figure 10 shows qualitative comparisons.

Results on Video Matting. In addition to the quantitative comparisons on the video matting dataset, we visualize the predictions of our method on two adjacent frames with the spatio-sparsity map in Figure 11. More video results can be found in the supplementary materials.

4.4. Ablation Studies

Unless otherwise stated, all the ablation studies are conducted based on our self-trained low-resolution prior network (LPN) and sparse high-resolution module (SHM) on HHM2K.

Analysis of Sparsity and Computations. Our method processes UHR images at full resolution in one shot due to the sparsity design. We show the cutoff of computations in Ta-
### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>MAD</th>
<th>MSE</th>
<th>Mem. (GB)</th>
<th>Lat. (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBA [13]</td>
<td>5.28</td>
<td>1.62</td>
<td>1.51</td>
<td>7.95</td>
</tr>
<tr>
<td>SIM [44]</td>
<td>9.30</td>
<td>4.59</td>
<td>2.40</td>
<td>10.09</td>
</tr>
<tr>
<td>FBA-512 + SHM</td>
<td>3.92</td>
<td>1.00</td>
<td>3.20</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Table 4. Comparisons of our framework with different patch-based methods aggregated with Cross-Patch Contextual module (CPC) [49]. The trimaps for FBA and SIM are generated from the groundtruth alpha matte. The adopted patch size is 512. Mem. and lat. are short for memory and latency.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAD</th>
<th>MSE</th>
<th>Grad</th>
<th>Conn</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPN</td>
<td>82.87</td>
<td>26.68</td>
<td>64.70</td>
<td>76.02</td>
</tr>
<tr>
<td>LPN + DGF</td>
<td>87.40</td>
<td>24.38</td>
<td>42.94</td>
<td>77.82</td>
</tr>
<tr>
<td>LPN + SHM</td>
<td><strong>63.05</strong></td>
<td><strong>19.98</strong></td>
<td><strong>28.80</strong></td>
<td><strong>53.47</strong></td>
</tr>
</tbody>
</table>

Table 5. Ablation results of sparse high-resolution module (SHM) on unknown region compared to DGF.

<table>
<thead>
<tr>
<th>kernel size $k$</th>
<th>5</th>
<th>15</th>
<th>25</th>
<th>35</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAD</td>
<td>7.98</td>
<td>7.90</td>
<td>7.90</td>
<td>7.90</td>
<td>7.91</td>
</tr>
</tbody>
</table>

Table 6. Ablation results of different dilation kernel size $k$.

<table>
<thead>
<tr>
<th>Method</th>
<th>MAD</th>
<th>MSE</th>
<th>Grad</th>
<th>Conn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours w/o SE</td>
<td>8.44</td>
<td>4.59</td>
<td>2.45</td>
<td>7.80</td>
</tr>
<tr>
<td>Ours w. SE</td>
<td>8.11</td>
<td>4.32</td>
<td>2.17</td>
<td>7.34</td>
</tr>
<tr>
<td>Ours w. Masked-SE</td>
<td><strong>7.90</strong></td>
<td><strong>4.29</strong></td>
<td><strong>1.96</strong></td>
<td><strong>7.19</strong></td>
</tr>
</tbody>
</table>

Table 7. Ablation studies on Masked-SE. We compare the models without SE, with SE, and with Masked-SE.

### Analysis of Sparse High-Resolution Module

As known, many approaches are proposed to address the high-resolution matting in previous works, such as Cross-Patch Contextual module (CPC) [49] and deep guided filter (DGF) [36]. CPC module is proposed to address the long-range dependency issue of patch-based inference while DGF is applied to upsample the predicted alpha matte for low-resolution matting methods. We claim that both the two are not the best choice when dealing with UHR matting.

Table 4 tabulates the comparisons with two state-of-the-art matting methods with and without CPC module. The adopted patch size is 512. Without any additional optimization, our sparse high-resolution module achieves superior results than the patch-based SIM and FBA, and even slightly better to patch-based FBA cooperated with CPC module. Notably, our method consumes much less memory at a about 30× faster speed compared to patch-based FBA with CPC module. Table 5 provides the performance comparison of DGF and our sparse high-resolution module on the boundary region.

**Analysis of Dilation.** $M$ specifies the input sparsity for the sparse high-resolution module. We use max pooling to simulate dilation operation, where $k$ is the kernel size of max pooling layer. Intuitively, large $k$ introduces computation overhead while undersized $k$ leads to insufficient neighboring context. To balance accuracy and efficiency, we conduct experiments on different $k$. Table 6 indicates that $k = 15$ is a suitable choice for both accuracy and efficiency.

**Analysis of Masked-SE.** Masked-SE bridges the low-resolution prior network and sparse high-resolution module. By separately encoding global foreground and background, Masked-SE learns a channel-wise attention weight. Table 7 shows SparseMat achieves an MAD performance of 7.90 with Masked-SE and 8.44 without Masked-SE, indicating a performance promotion due to Masked-SE. Furthermore, Masked-SE slightly outperforms SE [24] layer, showing the benefits of foreground and background separation.

### 5. Conclusion

This paper proposes a general matting framework **SparseMat** for UHR image/video matting, which is the first novel work on exploring the sparsity of general alpha matte. Our method is memory and computationally efficient, using low-resolution prior and temporal information to generate spatio-temporal sparsity map and a sparse high-resolution module (SHM) to refine alpha matte at full resolution in one shot without using patch-based strategy. This is the first work to explore the sparsity of UHR matting and utilize sparse convolution for solving general UHR matting problem.
References


[27] Levent Karacan, Aykut Erdem, and Erkut Erdem. Image matting with kl-divergence based sparse sampling. In IEEE International Conference on Computer Vision, 2015. 2

[28] Zhanghan Ke, Kaican Li, Yurou Zhou, Qiuhua Wu, Xianyu Mao, Qiong Yan, and Rynson W. H. Lau. Is a green screen really necessary for real-time portrait matting? Arxiv Preprint, 2020. 1, 2, 4, 6, 7

[29] Sun-Young Lee, Jong-Chul Yoon, and In-Kwon Lee. Temporal coherent video matting. In SIGGRAPH, 2010. 2


[33] Yaoyi Li and Hongtao Lu. Natural image matting via guided contextual attention. In AAAI Conference on Artificial Intelligence, 2020. 1, 2, 7

[34] Yaoyi Li, Qingyao Xu, and Hongtao Lu. Hierarchical opacity propagation for image matting. Arxiv Preprint, 2020. 2


14120


