procedure CUDACONVOLUTION(signal, kernel, K, L, M, norm)
    cuMemcpy(gpu_s, signal, HostToDevice)
    cuMemcpy(gpu_k, kernel, HostToDevice)
    gpu_s ← cuFFT(gpu_s)
    gpu_k ← cuFFT(gpu_k)
    gpu_s ← pwProd(gpu_s, gpu_k, K, L, M, norm)
    gpu_s ← cuIFFT(gpu_s)
    cuMemcpy(signal, gpu_s, DeviceToHost)
end procedure

Fig. 1. Convolution in CUDA. The function called cuMemcpy provides data transfers between CPU (Host) and GPU (Device). The pwProd provides a pointwise multiplication of two signals on GPU. The norm constant normalizes the convolution if required. The result is stored in signal variable in order to save memory. The result could be also stored in another parameter in case we needed to keep the input data.

### TABLE I

**METHODS FOR DECOMPOSITION OF THE CONVOLUTION PROBLEM AND THEIR REQUIREMENTS**

<table>
<thead>
<tr>
<th>method</th>
<th>no. of operations</th>
<th>no. of memory transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIF</td>
<td>$c(M_f + M_g) \log(M_f + M_g) + (M_f + M_g)$</td>
<td>$3(M_f + M_g)$</td>
</tr>
<tr>
<td>DIT</td>
<td>$c(M_f + M_g) \log(M_f + M_g) + 2(M_f + M_g)$</td>
<td>$(2 + P)(M_f + M_g)$</td>
</tr>
<tr>
<td>Tiling</td>
<td>$c(M_f + PM_g) \log(M_f + PM_g) + (M_f + 2PM_g)$</td>
<td>$2M_f + (P + 1)M_g$</td>
</tr>
</tbody>
</table>

separability of the resulting sub-problems. This leads to a lot of redundant data transfers. Secondly, in the spatial domain, the convolved image can be divided into small parts. This method will be referred to as the tiling. However, all the sub-images have to be extended with neighbouring pixels of at least size of the filter kernel [17]. Thus, a lot of redundant computation needs to be performed, proportionally to the PSF size.

The comparison of three approaches to divide the convolution problem is shown in Table I. $M_f$ and $M_g$ denote the size of the signal and kernel, respectively, $c$ is a constant corresponding to a per-pixel number of arithmetic operations needed for the computation of FFT, and $P$ denotes the number of parts the input data is divided into. The DIF method is the only one which is not depend on the $P$ parameter.