EasyChair Preprint

№ 6694

Fast Approximation of Color Morphology

Vivek Sridhar, Michael Breuß and Marvin Kahra

September 25, 2021

Fast Approximation of Color Morphology

Vivek Sridhar[1], Michael Breuss[1], and Marvin Kahra[1]

Institute for Mathematics, Brandenburg Technical University Cottbus-Senftenberg,
03046 Cottbus, Germany
{sridhviv,breuss,marvin.kahra}@b-tu.de

Abstract. The basic filters in mathematical morphology are dilation and erosion. They are defined by a flat or non-flat structuring element that is usually shifted pixel-wise over an image and a comparison process that takes place within the corresponding mask. The algorithmic complexity of fast algorithms that realise dilation and erosion for color images usually depends on size and shape of the structuring element.
In this paper we propose and investigate an easy and fast way to make use of the fast Fourier transform for an approximate computation of dilation and erosion for color images. Similarly in construction as many other fast algorithms, the method extends a recent scheme proposed for single-channel filtering. It is by design highly flexible, as it can be used with flat and non-flat structuring elements of any size and shape. Moreover, its complexity only depends on the number of pixels in the filtered images. We analyse here some important aspects of the approximation, and we show experimentally that we obtain results of very reasonable quality while the method has very attractive computational properties.

Keywords: Mathematical Morphology · Fourier Transform · Fast Algorithms

1 Introduction

Mathematical morphology is a highly successful field in image processing with abundant applications, see [10,11,13]. An elementary mechanism in morphology is to compare (and order) tonal data within a certain mask sliding over an image. The mask is defined as part of a so-called structuring element (SE), which is fundamental in morphological filtering. The SE is characterised by shape, size and centre location. There are in addition two types of SEs, flat and non-flat [19]. For a given centre pixel a flat SE defines a neighbourhood on which the tonal comparison takes place, whereas a non-flat SE also contains additive offsets in tonal space. The basic operations in morphology are dilation and erosion, where the tonal value at a pixel is set to the maximum and minimum respectively within the SE centred upon it. Many useful morphological processes, like e.g. opening or closing, are formulated by combining dilation and erosion.

Let us briefly review some concepts for color morphology. As mentioned above, a key issue in morphology is to perform a useful comparison of the tonal values within the SE. In standard grey value morphology this is done relying

on the concept of a complete lattice, making use of a total ordering of tonal values [10]. However, the construction of a comparison mechanism is in general a delicate issue when turning to color imagery, since there is no natural total ordering of color values. There are thus basically two approaches to tackle color morphology [5]: *(i)* channel-based, where each channel is individually treated like a grey-value image, and *(ii)* vector-based, where each color is processed as a vector in an underlying color space, see [3,4] for some approaches to construct a vector-based ordering. In this article we will resort to channel-wise filtering as do most fast algorithms.

As indicated, morphological algorithms aiming for computational speed are usually designed methodically for single-channel data, and applied channel-wise for color images. Let us refer to [2] for an example where in addition hardware optimized for channel-wise RGB processing is considered. Most fast (single-channel) methods either aim to reduce the size of a SE or to decompose it, or alternatively they attempt to reduce redundant comparison operations, compare [9]. Most of them refer to specific shape or size of the SE, sometimes also specific hardware like GPUs is addressed, see for instance [15,17,18,20]. There are just a few fast algorithms for non-flat SEs of arbitrary shape and size, even for single-channel data. Let us mention [22] which employs histogram updates during translation of a SE over an image. However, in general the algorithmic complexity of these fast algorithms inherently relies on size and shape of the SE.

Here we will build upon a different approach which eventually leads to a fast method having no limitations with respect to size, shape or flatness of the SE. In [8] it was shown that binary dilation / erosion may be performed employing specific convolutions. By the convolution theorem this enables the possibility of computing binary dilation / erosion using the Fast Fourier Transform (FFT) [12]. In [14] this approach was extended to grey value imagery, by processing each grey value level set as a binary image using the technique from [8], and combining thereafter the results. The proceeding implies that the method is limited to flat SEs. We proposed in [23] an alternative construction that circumvents this limitation by approximating the original convolutions. However, as we observed experimentally in [23], this comes at the expense of a certain shift in grey value data.

Our contribution. In this paper we extend the method described in [23] to color imagery. The basic construction is extended channel-wise in a straightforward way. We confirm that favourable computational properties are maintained in the color setting. Furthermore, we discuss the occurring tonal shift in detail, as this may be of importance when filtering multiple channels. We derive an estimate for the size of the shift and give experimental evidence that its effect appears to be negligible in practice, since the vast majority of morphological applications naturally relies on combinations of dilation and erosion. In various experiments, we show that our method is extremely fast and yields competitive quality compared to other possible methods for color morphology.

2 Basic Definitions

We start by considering a two dimensional, discrete image domain $\Omega \subset \mathbb{Z}^2$. A single-channel, grey value image can be represented as a function $f : \Omega \to L$, where L is the set of possible grey values. In non-flat morphology, the SE itself can be perceived as a grey value image. A non-flat SE b can thus be defined as a function $b : B \to L$, for B denoting a suitable set centred at the origin. A flat filter is just a special case where $b(x) = 0$ for all $x \in B$.

Dilation, Erosion and its Combinations The fundamental building blocks of mathematical morphology are dilation and erosion. The *dilation* of an image f by a SE b is given by $f \oplus b : \Omega \to L$, where $(f \oplus b)(x) = \max_{u \in B} \{f(x - u) + b(u)\}$. The *erosion* of an image f by a SE b is given by $f \ominus b : \Omega \to L$ and can be computed by $(f \ominus b)(x) = \min_{u \in B} \{f(x + u) - b(u)\}$.

Many morphological operations of practical interest can be composed by dilation and erosion, cf. [10]. As important examples let us mention here *opening* $f \circ b = (f \ominus b) \oplus b$ and *closing* $f \bullet b = (f \oplus b) \ominus b$. Similarly, composite operation of *white top hat* is defined as $f \boxminus (f \circ b)$, where $(f \boxminus (f \circ b))(x) = f(x) - (f \circ b)(x)$. The *black top hat* is $(f \bullet b) \boxminus f$, and the *Beucher gradient* is $(f \oplus b) \boxminus (f \ominus b)$, *internal gradient* is $f \boxminus (f \ominus b)$ and *external gradient* is $(f \oplus b) \boxminus f$.

Channel-Wise Color Morphology The most intuitive and simple approach to color morphology we will also follow in this work, is to deal with the image component wise. There are many formats to represent a digital image [6]. We employ here the classic RGB format using three separate channels to store the red, green and blue value for each pixel of an image. On a discrete domain $\Omega \subset \mathbb{Z}^2$, a RGB color image can be thus represented as $f_C : \Omega \to L_C$. Here, $L_C = L_r \times L_g \times L_b$, where L_r, L_g and L_b are the possible values of pixels in red, green and blue channel respectively. For RGB it holds $L_r = L_g = L_b = L$, and L is the set of non-negative integers ≤ 255.

Let us extend the concept of single-channel non-flat SE, to non-flat SE with 3 channels, in a straightforward fashion. A SE is given by the function $b_C : B \to L_C$. Thus an image f_C can be separated into three channels (f_r, f_g, f_b) and each of these channels can be treated like a grey value image. Similarly, a SE b_C can be separated into (b_r, b_g, b_b) and each of the components acts on the corresponding channel of the image. Thus, for instance, we define color dilation of image f_C by SE b_C as $f_C \oplus_C b_C = (f_r \oplus b_r, f_g \oplus b_g, f_b \oplus b_b)$. The definitions of other component-wise color morphological operations follow in accordance.

3 Fast Approximation of Color Morphology

We first recall briefly the basics and implementation of the method from [23]. Observe that by channel-wise extension to color images, we are considering the most general form of color SE with no restriction on its shape or size. After that we investigate an important property of the method, namely the tonal shift arising by approximation in each color channel.

3.1 Fourier-Based Approximative Morphology

We have found in [23] an efficient way to dilate or erode grey scale images by means of Fourier transforms. For this purpose, we approximated the sup contained in the continuous-scale formulation of grey value dilation using

$$\sup(x_1, \ldots, x_k) \doteq \lim_{m\to\infty} \frac{1}{m} \log\left(\sum_{i=1}^{k} \mathrm{e}^{mx_i}\right). \tag{1}$$

Thus one may obtain for dilation in an individual channel the representation

$$\begin{aligned}(f \oplus b)(x) &= \lim_{m\to\infty} \frac{1}{m} \log\left(\sum_{y\in\mathbb{E}} \mathrm{e}^{mf(y)}\mathrm{e}^{mb(x-y)}\right) \\ &= \lim_{m\to\infty} \frac{1}{m} \log \mathcal{F}^{-1}\left[\mathcal{F}\left[\mathrm{e}^{mf}\right] \cdot \mathcal{F}\left[\mathrm{e}^{mb}\right]\right](x),\end{aligned} \tag{2}$$

for which the convolution theorem was applied, and where b was extended to the whole domain by setting $b(x)$ to $-\infty$ outside of the set B. In order to retrieve a flat structuring element, one may set $b(x) = 0$ over B. Let us note, $b(x)$ can be suitable defined to realise any non-flat SE over B.

Following our specifications in [23] for implementation in individual color channels, the method is easily extended to channel-wise processing. In order to consider arbitrary sets B as shapes for the SE, we extend $b(x)$ to the whole image domain by setting $b(x)$ to -256 for implementation. Analogously to [23] let us declare the function $\mathbb{E}\mathrm{xpm}(\cdot)$ as $\mathbb{E}\mathrm{xpm}(f)(x) = e^{(m\cdot f(x))}$. It should be noted that we use functions of the NumPy package [24] to implement all the functions described in this section, unless otherwise mentioned.

To obtain the fast convolution of $\mathbb{E}\mathrm{xpm}(f)$ and $\mathbb{E}\mathrm{xpm}(b)$, we use the FFT as in [23]. To this end we employ the function *scipy.signal.fftconvolve(·)* from the SciPy package [21]. When using the FFT with large numbers, which may occur during the discrete transformations, it is in principle possible that some overflow errors or errors of the form $0/0$ or ∞/∞ may occur. Though such an error has not occurred during our experiments, one may handle such exceptions in a straightforward manner which can be included in the definition of a function h that yields the discrete convolution of $\mathbb{E}\mathrm{xpm}(f)$ with $\mathbb{E}\mathrm{xpm}(b)$, cf. [23].

Finally, we need the corresponding Fourier-based inverse of $\mathbb{E}\mathrm{xpm}(\cdot)$. For this we define $\mathrm{In}\mathbb{E}\mathrm{xpm}(\cdot)$ as $\mathrm{In}\mathbb{E}\mathrm{xpm}(h)(x) = \frac{1}{m} \cdot \log(h(x))$. Thus one can completely transfer (1) to the discrete domain in a channel-wise setting.

The fast dilation is therefore given by $f \oplus b \approx \mathrm{Inte}(h)$, where $\mathrm{Inte}(h)(x) = \lfloor h(x) \rfloor$. So by the well-known complexity of the FFT, the whole process takes place in $O(n \log n)$ if the SE size is smaller than the image under consideration, where n denotes the number of image pixels. This theoretical complexity is confirmed by experiments in Section 4.1. The analogous fast erosion technique follows directly from the duality property of erosion and dilation in each channel, as $f \ominus b = 255 - ((255 - f) \oplus \breve{b})$, where $\breve{b}(-x) = b(x)\ \forall x \in B$.

3.2 Tonal Shift Analysis

We have noted a certain shift of tonal values in our work [23], but not analyzed. In the context of filtering in multiple channels, this shift may represent a major hindrance since shifts of varying magnitude in different channels may potentially spoil results of combinations of dilation and erosion.

By Figures 1 and 2, one may observe that there is a positive (negative) shift in each channel when the fast approximation method is used for dilation (erosion). By this dilation (erosion) results appear lighter (darker). For this reason, we want to calculate an upper bound for the positive shift with fast dilation in the respective channels. Similar calculation holds for negative shift with fast erosion.

Let A be a finite set of non-negative integers, i.e $A = \{x_1, x_2, \cdots x_n\}$. Furthermore let the maximum be denoted by $\max A = x_k$ and approximated by $\alpha = \left\lfloor \frac{1}{m_1} \ln \left(\sum_{i=1}^{n} e^{m_1 \cdot x_i} \right) \right\rfloor$, where $m_1 > 0$ is some constant.

Clearly, the difference between the approximation and the actual maximum, $\alpha - x_k$, is non-negative as:

$$x_k \;=\; \left\lfloor \frac{1}{m_1} \ln \left(e^{m_1 \cdot x_k} \right) \right\rfloor \leq \; \left\lfloor \frac{1}{m_1} \ln \left(\sum_{i=1}^{n} e^{m_1 \cdot x_i} \right) \right\rfloor \;=\; \alpha. \tag{3}$$

Thus, we do not observe any negative (positive) shifts during dilation (erosion).

Next, we calculate the *worst-case scenario* for positive shift during dilation, that is, the upper bound of the difference $\alpha - x_k$, as follows:

$$\begin{aligned} \alpha \;&\leq\; \left\lfloor \frac{1}{m_1} \ln \left(\sum_{i=1}^{n} e^{m_1 \cdot x_k} \right) \right\rfloor = \; \left\lfloor \frac{1}{m_1} \ln \left(n \cdot e^{m_1 \cdot x_k} \right) \right\rfloor \\ &= \; \left\lfloor \frac{1}{m_1} \ln (n) + \frac{1}{m_1} \ln \left(e^{m_1 \cdot x_k} \right) \right\rfloor = \; \left\lfloor \frac{1}{m_1} \ln (n) \right\rfloor + x_k. \end{aligned} \tag{4}$$

It follows that $\alpha - x_k \leq \left\lfloor \frac{1}{m_1} \ln (n) \right\rfloor$ and if $m_1 > \ln (n)$ then $\left\lfloor \frac{1}{m_1} \ln (n) \right\rfloor = 0$, i.e $\alpha = x_k$.

Therefore, in Figure 1 an upper bound on the positive shift with fast dilation is $\left\lfloor \frac{1}{0.16} \ln (7 \times 7) \right\rfloor = \left\lfloor \frac{1}{0.16} \ln (49) \right\rfloor = 24$. Similarly, the fast erosion in Figure 1 has a negative shift of magnitude ≤ 24. This estimate is clearly confirmed uniformly over all color channels in the respective histograms, see Figure 2.

Considering the color histograms, let us comment that a certain smoothing of the color distribution is observed for the fast approximations of dilation and erosion. This is an interesting effect not pointed out in [23]. However, it is clearly observed that major characteristics of the histogram as well as peak magnitudes are largely the same as with the standard channel-wise filtering procedure.

With fast closing as an example for combining dilation and erosion, the tonal shifts are not apparent as they appear uniformly over the color channels. Thus we conjecture that the positive and negative shifts from dilation resp. erosion cancel

Fig. 1. Component-wise morphological operations with flat square 7×7 filter on 512×512 *Pepper* image. **Top row:** Classical method. **Bottom row:** Proposed fast approximations. **From left to right:** Dilation, erosion and closing.

each other out. This is an aspect of high practical relevance, as most applications of morphology rely on suitable combinations of dilation and erosion.

4 Experiments

Let us comment first on the choice of the parameter m. As indicated via formula (1) theory seems to motivate to choose m in accordance to $m \to \infty$. But since we form the maximum over a finite range of discrete values, it is sufficient to choose a finite value of m. Furthermore, the relation (1) is evaluated in the Fourier domain. So in practice, the value of m refers to the implementation of the FFT. Since, we perform the computations using numpy.float datatype, it is observed that the choice of $m = 0.16$ avoids overflow errors and provides reasonable approximations.

For useful comparison we employed a highly efficient channel-wise morphology implementation which we could find openly available, via morphological dilation in the SciPy package [21]. The SciPy routine ndimage.grey_dilation() technically makes use of an efficient implementation of a sliding window technique in the style of [1], to compute channel-wise dilation with arbitrary non-flat SE similarly to the process described in [22]. Note that the worst case complexity of this traditional fast morphological method with non-flat SE, even if optimised in implementation, is theoretically still $O(n_b \times n)$.

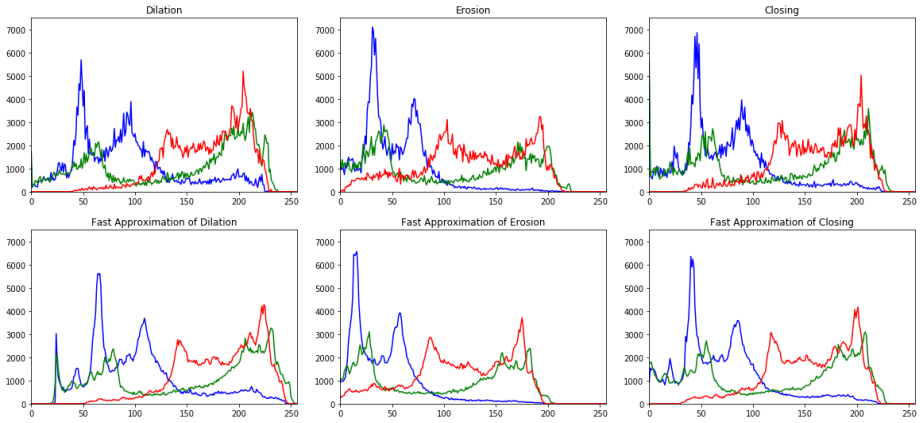

Fig. 2. Histograms of component-wise morphological operations with flat square 7×7 filter on 512×512 *Pepper* image, compare visual results in Figure 1. **Top row:** Classical method. **Bottom row:** Proposed fast approximations, **From left to right:** Dilation, erosion and closing

4.1 Comparison of Computational Times

Let us briefly recall the asymptotic complexity of our process for approximating component-wise color dilation. We start with the trivial assumption that the size n_b of the filter is $\leq$ size n of the image. For each channel, fast dilation involves a sequence of linear time processes, except for FFT and inverse FFT which are $O(n \log n)$. Erosion, computed using duality, involves two linear time operations preceding and one linear time operation following the dilation. Therefore, fast erosion is also asymptotically $O(n \log n)$. Since most of the other morphological operations are combinations of erosions, dilations and linear in time operations, the fast approximations of combined operations are also performed in $O(n \log n)$.

In the first experiment, Figure 3 *Left*, we evaluate the time taken by varying size of filter on a fixed size of image. The image used is again *Pepper* of size 512×512. The filter size increases from 1×1 to 43×43. The filters are color filters generated using np.random.randint(), with range of values from 0 to 255. It is clearly visible that the SciPy method behaves linearly with respect to size of filter, taking 0.12 seconds for filter size 5×5 to around 7.5 seconds for filter size 43×43. The fast method remains constant with respect to the size of the filter, taking on average about 0.09 seconds, i.e. 0.03 for each channel.

In the second experiment, Figure 3 *Right*, we evaluate the time taken for component-wise color dilation by a fixed size of color filter, 51×51, on varying image sizes. The image size increases from 101×101 to 1201×1201. The images and the filters are generated using np.random.randint(), with range of values from 0 to 255. The SciPy method takes 0.08 seconds for the 101×101 image and increases to about 53 seconds for the 1201×1201 image. The proposed fast method is efficient overall and the rate of increase in time with respect to the

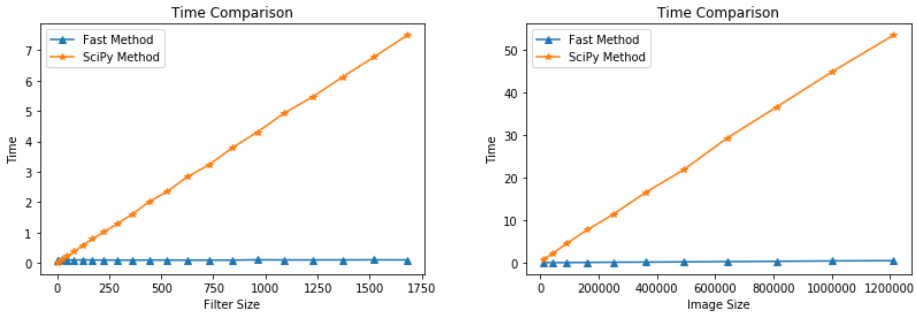

Fig. 3. Time comparison of component-wise color dilation using the proposed fast method versus the dilation function ndimage.grey_dilation() in the SciPy Package. **Left:** Varying SE sizes on an image of size 512×512. **Right:** SE of size 51×51 on varying image sizes.

size of the image is very low. It takes 0.02 seconds for the 101×101 image and 0.48 seconds for the 1201×1201 image.

Let us note that we have not employed GPUs in the above experiments. It is worth pointing out that attempts to use GPUs to compute grey value morphology usually places restrictions of symmetry and/or flatness on the SE, see discussions in [16,20]. However, there are several efficient implementations of FFT and inverse FFT on the GPU, see e.g. [7]. Therefore, our method could be sped up utilising the GPUs, without any restrictions on the SE.

4.2 Visual Quality Comparison

To assess visual quality of filtering, we consider in addition to channel-wise filtering the Loewner ordering method from [25]. For the latter, color images are formulated as fields of symmetric 2×2 matrices and processed by rather complex rules inspired by theoretical physics. We restrict ourselves here to flat SEs and a selection of experiments.

Fig. 4. Dilation of 512×512 *Pepper* image by flat 21×21 diamond shaped SE. **Left:** Loewner. **Centre:** Classic component-wise. **Right:** Proposed fast method

Fig. 5. Beucher Gradient of 512×512 *Pepper* image by flat 3×3 square shaped SE. **Left:** Loewner. **Centre:** Classic component-wise. **Right:** Proposed fast method

Dilation results in Figure 4 are at first glance qualitatively similar. A careful inspection shows that the Loewner scheme shows few more details than the channel-wise methods, and that the fast approximation result appears as expected a little lighter and smoother.

Let us turn to the Beucher gradient as an example of a morphological process that could be useful (as internal or external gradient) for edge detection or segmentation. Results displayed in Figure 5 show that the Loewner method gives apparently a different color distribution than the channel-wise methods. Let us note that the Loewner method relies on the underlying structure of a different color space than the RGB-based channel-wise methods. All methods give cleanly the apparent edges of the input image.

Now let us consider another composed filter, the white top hat, see Figure 6. It is expected that the filtering results highlight points or structures that are lighter than their surrounding (in terms of the individual color channels). This means, it is expected here that results allow to identify shiny highlights as well as the trunks of the two peppers in the foreground of the image. As it can be observed, all the methods give useful results in this respect.

Fig. 6. White top hat of 512×512 *Pepper* image by flat 21×21 diamond shaped SE. **Left:** Loewner. **Centre:** Classic component-wise. **Right:** Proposed fast method

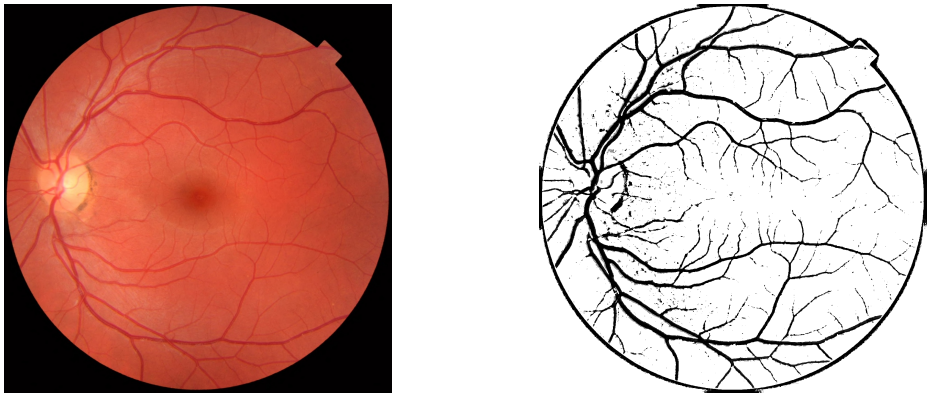

Fig. 7. As an example for potential applications, we take a retinal image of size 1411×1411 (source: Wikipedia), set the red channel as 0, and perform a black top hat with the proposed method, using a 25×25 flat square S.E. We then do a gamma correction followed by thresholding and an inversion for visualization of the blood vessels. The whole process takes ≤ 1.6 seconds, without the use of GPU. **Left:** Retinal test image. **Right:** Inversion after thresholding (≥ 11) after Gamma Correction with $\gamma = 0.85$.

As a last example for visual quality assessment and at the same time as a proof-of-concept for the usefulness of the developments, we consider to process a retinal image, see Figure 7. Such images arise in ophthalmic examination, and by advances in medical imaging the image resolution for given tasks of automatic processing is ever-growing. Therefore the need arises for very fast methods for processing such data, compare e.g. [26]. As Figure 7 shows, our method is capable of useful filtering such images.

5 Conclusion

Our method not only works with complexity $O(n \log n)$ with respect to the size of the image, it is also extremely fast in practice. Let us stress again that it performs overall much faster than standard efficient implementations of morphology as available in SciPy. At the same time it is highly versatile, as there are no restrictions to shape, size or flatness of the SE.

As we have demonstrated, our method may perform very fast with imagery from modern acquisition devices, which may often be of high resolution and in color. By morphological filtering of high resolution images, naturally also the size of useful SEs is at least moderate. We conjecture that a method whose complexity is independent from the SE size parameter may be useful for applications.

Future work may include extension to multi-spectral imagery. In this context it may be important that the method offers high potential to accelerate it even further using GPUs for the incorporated FFT. Furthermore, we aim to study alternative transforms that may reduce the tonal shift visible in pure dilation/erosion filtering.

References

1. Douglas, S.C., Running max/min calculation using a pruned ordered list. IEEE Transactions on Signal Processing, 44(11), pp. 2872-2877. 1996.
2. Pedrino, E.C., Saito, J.H., Senger, H. and Roda, V.O., Color Mathematical Morphology In A FPGA. In Proceedings of the International Conference on Systems, Signals and Image Processing. 2010.
3. Aptoula, E. and Lefèvre, S., On lexicographical ordering in multivariate mathematical morphology. Pattern Recognition Letters, 29(2), pp. 109-118. 2008.
4. Lezoray, O., Elmoataz, A. and Meurie, C., Mathematical morphology in any color space. In Proceedings of the International Conference of Image Analysis and Processing – Workshops, pp. 183-187. 2007.
5. Barnett, V., The ordering of multivariate data. Journal of the Royal Statistical Society: Series A (General), 139(3), pp. 318-344. 1976.
6. Sharma, G. and Bala, R. (eds.), Digital color imaging handbook. CRC Press. 2017.
7. Moreland, K. and Angel, E., The FFT on a GPU. In Proceedings of the ACM SIGGRAPH/Eurographics conference on Graphics hardware, pp. 112-119. 2003.
8. Tuzikov, A.V., Margolin, G.L. and Grenov, A.I., Convex set symmetry measurement via Minkowski addition. Journal of Mathematical Imaging and Vision, 7(1), pp. 53-68. 1997.
9. Van Droogenbroeck, M. and Buckley, M.J., Morphological erosions and openings: fast algorithms based on anchors. Journal of Mathematical Imaging and Vision, 22(2), pp. 121-142. 2005.
10. Serra, J. and Soille, P. (eds.), Mathematical morphology and its applications to image processing (Vol. 2). Springer Science & Business Media. 2012.
11. Najman, L. and Talbot, H. (eds.), Mathematical morphology: from theory to applications. John Wiley & Sons. 2013.
12. Cooley, J.W., Lewis, P.A. and Welch, P.D., Historical notes on the fast Fourier transform. Proceedings of the IEEE, 55(10), pp. 1675-1677. 1967.
13. Roerdink, J.B., Mathematical morphology in computer graphics, scientific visualization and visual exploration. In Proceedings of the 10th International Symposium on Mathematical Morphology, pp. 367-380. 2011.
14. Kukal, J., Majerová, D. and Procházka, A., Dilation and erosion of gray images with spherical masks. In Proceedings of the Annual Conference of Technical Computing. 2007.
15. Déforges, O., Normand, N. and Babel, M., Fast recursive grayscale morphology operators: from the algorithm to the pipeline architecture. Journal of Real-Time Image Processing, 8(2), pp. 143-152. 2013.
16. Moreaud, M. and Itthirad, F., Fast algorithm for dilation and erosion using arbitrary flat structuring element: Improvement of Urbach and Wilkinson's algorithm to GPU computing. In Proceedings of the International Conference on Multimedia Computing and Systems, pp. 289-294. 2014.
17. Lin, X. and Xu, Z., A fast algorithm for erosion and dilation in mathematical morphology. In Proceedings of the WRI World Congress on Software Engineering, Vol. 2, pp. 185-188. 2009.
18. Van Herk, M., A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. Pattern Recognition Letters, 13(7), pp. 517-521. 1992.
19. Haralick, R.M., Sternberg, S.R. and Zhuang, X., Image analysis using mathematical morphology. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(4), pp. 532-550. 1987.

20. Thurley, M.J. and Danell, V., Fast morphological image processing open-source extensions for GPU processing with CUDA. IEEE Journal of Selected Topics in Signal Processing, 6(7), pp. 849-855. 2012.
21. Virtanen et al., SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods, 17(3), pp. 261-272. 2020.
22. Van Droogenbroeck, M. and Talbot, H., Fast computation of morphological operations with arbitrary structuring elements. Pattern Recognition Letters, 17(14), pp. 1451-1460. 1996.
23. Kahra, M., Sridhar, V. and Breuß, M., Fast Morphological Dilation and Erosion for Grey Scale Images Using the Fourier Transform. In Proceedings of the 8th International Conference on Scale Space and Variational Methods, pp. 65-77. 2021.
24. Harris et al., Array programming with NumPy. Nature, 585(7825), pp. 357-362. 2020.
25. Burgeth, B. and Kleefeld, A., An approach to color-morphology based on Einstein addition and Loewner order. Pattern Recognition Letters, 47, pp. 29-39. 2014.
26. Dachsel, R., Jöster, A. and Breuß, M., Real-time retinal vessel segmentation on high-resolution Fundus images using Laplacian pyramids. In Proceedings of the 9th Pacific Rim Symposium on Image and Video Technology, pp. 337-350. 2019.