

FFT 'light' - Image Deconvolution based on Real to Real-Space Frequency Transformations

F. Margarine¹ and P.C. Cheng²

¹Department of Physical Optics, University of Sydney, Sydney, NSW, Australia

²AMIL, Department of Electrical Engineering, University at Buffalo, Buffalo, NY 14260 USA

The process of deconvolving wide-field or confocal microscopical image stacks comprises the convolution of the dataset with the inverse point-spread function (PSF) subject to a set of constraints or cost-functions to prevent the amplification of noise. For the implementation as a filter, it generally is difficult to predict and limit the filter size (support) of discrete inverse PSF, and the computational overhead of straightforward filtering is directly proportional to the size of the filter (*i.e.* the number of elements within the filter). The discrete convolution process itself can be expressed as a simple element-wise scalar multiplication if the image has been transformed into frequency space by a FFT, this advantage justifies its popularity in the deconvolution operation. The "cost" of transformation to and from frequency space is lower than straight convolution if the filter of the inverse PSF requires more than some 10 to 50 elements. To be able to exploit this benefit, the convolution and the application of the constraints are implemented as two separate and consecutive steps despite being tightly linked. Due to this implementation, the choice of the constraints and cost functions becomes somewhat arbitrary set from the perspective of the involved physics and is mostly driven by the achievable computational efficiency. As a consequence, the steps of convolution and re-establishing constraints are iterated; if the series of produced images converges, the outcome will depend only on the PSF and the set of constraints, not on the implementation of the algorithm itself.

However, the beauty of the FFT in convolution applications is somewhat shadowed by the fact that its results are expressed in complex vectors. In addition, there is a need to use meaningful cut-off functions at the edge of the image volume to prevent the introduction of artificial ripple frequencies induced by sudden changes in brightness. Both requirements drive up memory allocation and computational resources. The expenditure is of non-negligible magnitude. Despite being an optimized method, the major drawback of iterated deconvolution in microscopy remains the time requirement. In comparison to confocal microscopy which can obtain satisfactory optical sections within minutes, a typical commercial deconvolution system operating on a state-of-the-art personal computer may take several minutes to hours to perform the operation, rendering the technique much less attractive for biological sciences.

Real-space frequency transformations are less general than the FFT, and they do not share its property of offering a trivial convolution operation; but they feature, at most, half the appetite for memory and CPU cycles. The aim of this talk is to demonstrate that meaningful boundary conditions or window functions for image stacks exist which allow for the use of a more restricted frequency transformation [5,7,10] for which efficient implementations exist [1, 2, 5, 6, 8]. The implementation of convolution operators immediately derives from the one-dimensional solutions [4, 5], the implementation stays efficient, however it is far less elegant than the FFT equivalent. For two images I_0 and I_1 , the famous Fourier identity $F(I_0 \otimes I_1) = F(I_0) \cdot F(I_1)$ holds, whereas for the Hartley transform, the bulkier equivalence

$$F(I_0 \otimes I_1) = H(I_0(u, v)) \cdot \frac{1}{2} [H(I_1(u, v)) + H(I_1(-u, -v))] + H(I_0(-u, -v)) \cdot \frac{1}{2} [H(I_1(u, v)) - H(I_1(-u, -v))]$$

has to be employed which illustrates why one should invest effort to bypass the full generality of FFT operations in the process. From the scheme also the benefits of symmetric – or forced symmetric – images become obvious, as four out of six transformations disappear.

The other equivalence heavily exploited for symmetric functions $I(u) = I(-u)$ becomes obvious if the discrete FT is written in components centered around the origin:

$$F(I) = \sum_{k=-N/2}^{N/2} \cos\left(\frac{2\pi k}{N}\right) I(k) + i \sum_{k=-N/2}^{N/2} \sin\left(\frac{2\pi k}{N}\right) I(k),$$

where the imaginary term is zero due to $\sin(u) = -\sin(-u)$ and the transform becomes identical to the discrete cosine transform.

The real-to-real space transformations fail dramatically for PSFs which are not axially symmetric [9], for these cases, hybrid transformations have to be introduced which focus on fast processing of the image data and use pre-computed static inverse PSFs derived via FFT or Hartley transform. Filter algorithms have to be redesigned to handle these now split data paths, forming a class of hybrid algorithms for a mixture of real domain transformations and FFT covering most deconvolution requirements. The performance of the hybrid algorithms directly derives from the ratio of real-to-real transformations vs. required FFTs. The optimization strategy is to minimize the use of the inverse FFT to preferably once per data set and to recycle non-symmetric data to keep ‘FFTed’ data sets static during the computation, employing a result-cache.

References

- [1] F. A. Kamangar and K. R. Rao: Fast Algorithms for the 2-D Discrete Cosine Transform, IEEE Trans. Comput., 31, (1982), 899-906.
- [2] R. F. Ullmann: An Algorithm for the Fast Hartley Transform, Stanford Exploration Project Repots SEP-38 (1984).
- [3] M.A. Haque: A two-dimensional fast cosine transform, IEEE Trans. on Acoustics, Speech, and Signal Processing, 33, (1985) 1532-1539.
- [4] M. Vetterli: Fast 2-D discrete cosine transform, *in*: Proc. International Conference on Acoustics, Speech and Signal Processing, (1985).
- [5] R. N. Bracewell: The Hartley Transform, Oxford Engineering Science Series, No 19 (1986).
- [6] T. Le-Ngoc, M. T. Vo: Implementation and Performance of the Fast Hartley Transform, IEEE Micro Vol. 9, (5) pp. 20-27 (1989)
- [7] K.R. Rao and P. Yip: Discrete Cosine Transform: Algorithms, Advantages, and Applications, Academic Press, (1990).
- [8] E. Feig and S. Winograd: Fast Algorithms for the Discrete Cosine Transform, IEEE Trans. on Signal Proc., 40, (1992), 2174-2193.
- [9] V. Vesely: Fast Algorithms of Fourier and Hartley Transform and Their Implementation, In: MATLAB Proceedings of DATASTAT, Vol. 7 (1997)
- [10] G. Strang: The Discrete Cosine Transform, SIAM Review, (1999).