Lucky DCT Aggregation for Camera Shake Removal
Sanjay Ghosh, Satyajit Naik, and Kunal N. Chaudhury
Department of Electrical Engineering, Indian Institute of Science.
Email: {sghosh, satyajit, kunal}@ee.iisc.ernet.in

**INTRODUCTION**

- Modern cameras come with a burst mode for capturing a series of images in quick succession.
- Multiple-image blind deconvolution [1, 2] recovering a sharp image from such a burst.
- The mathematical model is that we have blurred versions \(y_1, y_2, \ldots, y_N\) of a sharp image \(x\):
  \[ y_i = k_i \ast x + \sigma n_i \quad (i = 1, \ldots, N), \]
  where \((k_i)\) are the blurring kernels, \((n_i)\) are i.i.d. \(N(0,1)\), and \(\sigma\) is the noise level.
- The problem is to recover the unknown image \(x\) from the burst images \(y_1, y_2, \ldots, y_N\).

**OBJECTIVES**

- To remove the effect of camera shake during a long exposure in hand-held photography.
- A simple and cheap algorithm that can effectively recover the original sharp image from multiple burst images.

**PROPOSED ALGORITHM**

- Input: Images \(y_1, y_2, \ldots, y_N\) of size \(M_1 \times M_2 \times c\).
- Parameters: Integers \(p \geq 0\) and \(N_0 \leq N\).
- Output: Output image \(\hat{x}\).

1. for \(i = 1, 2, \ldots, N\) do
   - Compute \(\mathcal{E}_i\) (using (1));
2. end
3. Select first \(N_0\) images for \(i = 1, 2, \ldots, N_0\) do
   - \(Y_i = D(y_i)\) \% DCT
   - \(Z_i = G(Y_i)\) \% Smoothing
   - \(\tilde{Y}_i = [Y_i^p \odot Z]_+ = G(\tilde{w}_i)\) \% Smoothing
   - \(Y = Y \oplus (\tilde{w}_i \odot Y_i)\) \% Aggregation
   - \(w = w \oplus w_i\)
4. end
5. \(\bar{Y} = Y \odot w\) \% Normalization
6. \(\hat{x} = D^{-1}(\bar{X})\) \% Inverse DCT

\[ \mathcal{E} = \sum_{\ell \in \text{support}(y)} \left\| \nabla y(\ell) \right\|^2. \]

where \(\nabla y\) is the gradient of \(y\) and \(\Omega_\ell\) is an \(n \times n\) window around pixel \(\ell\).

- Let \(Y_i\) denote the Gaussian-smoothed version of the DCT of burst image \(y_i\), that is, \(Y_i = G(D(y_i))\). For some non-negative integer \(p\), we define the weights:
  \[ \tilde{w}_i(\nu) = \frac{\left| Y_i(\nu) \right|^p}{\sum_{j=1}^N \left| Y_j(\nu) \right|^p}, \]

where \(\nu\) is the frequency index for the DCT. The integer \(p\) controls the nature of the aggregation.

- The DCT of the aggregated image is:
  \[ \bar{X}(\nu) = \sum_{i=1}^{N_0} \tilde{w}_i(\nu) Y_i(\nu), \]

The aggregated image is: \(\hat{x} = D^{-1}(\bar{X})\), where \(D^{-1}\) stands for the inverse DCT.

The complete algorithm is summarized at the left. The symbols \(\oplus, \odot, \odot, \|\) denote pixelwise addition, multiplication, and division, performed on each of the \(c\) channels.

The kernels \(\hat{a}\) are used to model the camera shake which occurs due to hand tremor.

- The proposed method is built upon - lucky imaging using Dirichlet energy [3] and Fourier Burst Accumulation (FBA) [4].
- The images (total \(N\)) are sorted according to decreasing Dirichlet energy. Then the top \(N_0\) images with large energies aggregated.
- The Dirichlet energy for an image \(y\) is defined as [3]:

**REFERENCES**


**RESULTS**

- For simplicity, we have assumed that the images in the burst are perfectly aligned [4].

- The proposed method is faster than [4] by a factor of about \(N_0/N\), neglecting the overhead of computing the Dirichlet energies and ranking them.

- The proposed method is faster than [4] by a factor of about \(N_0/N\), neglecting the overhead of computing the Dirichlet energies and ranking them.

- Better and faster camera shake correction is achieved through outlier rejection.
- Similar idea can be extended to remove camera shakes from videos.

**CONCLUSIONS**

**ACKNOWLEDGEMENTS**

- Centenary Conference Travel Fund, Department of Electrical Engineering, IISc.
- SPCOM Student Travel Grant.

**Fig. 1.** Deblurring results for the blurred images (also with additive Gaussian \(\sigma = 5\)) using FBA [4] and the proposed method. We used \(N_0 = 2\) for our method, from the dataset of total \(N = 14\) blurred images.

**Fig. 2.** Zoomed versions of the boxed portions of the deblurred images in Fig. 1.

**Fig. 3.** PSNR vs \(\sigma\), where \(x\) is in Fig. 1 (a).