Python-based Reconstruction of Blurry Frames using Event Cameras

Ahmed Alghfeli Hassan Alhosani Reem Almheiri

I. Abstract

This paper introduces the Event-based Double Integral method for deblurring using events. The proposed method aims to generate a sharp video from a single blurry image and its corresponding event data. This method offers significant contributions, including the restoration of a high frame-rate sharp video from a single image and event data, a stable and general approach for generating sharp videos under different types of blur, and the potential for a high temporal resolution in the reconstructed video. The methodology encompasses the event camera model, intensity image formulation, event-based double integral model, and high frame-rate video generation. The paper also presents a Python implementation of the algorithm and discusses the results and findings, including the impact of a slicing approach, the effect of the parameter c, and the challenges faced in replicating the time shifting technique proposed in the original paper. The results demonstrate the algorithm's ability to identify and restore blurred regions, and the potential for further enhancements through parameter tuning and advanced techniques.

II. Introduction

Blurred images are a common problem in various imaging applications, resulting from factors such as camera shake or object motion. In this paper, we propose an approach called the Event-based Double Integral (EDI) method for deblurring using events. This method aims to create a sharp video from a single blurry image and its corresponding event data, leveraging the advantages of event cameras and exploiting the connection between events and blur.

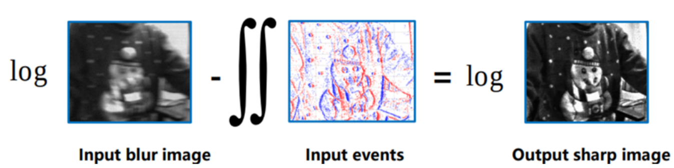


Fig. 1. Graphical Representation of EDI Method

The EDI method offers several key contributions. First, it enables the restoration of a high frame-rate sharp video from a single image, even if the image is blurry, by utilizing the available event data. Second, our method provides a stable and general approach for generating sharp videos under various types of blur, achieved through solving a single-variable non-convex optimization problem. Lastly, the reconstructed video can theoretically have a frame rate as high as the event rate, allowing for high temporal resolution in the output video.

III. Methodology

In this section, the approach and techniques used for the development of deblurring using events is presented. Although blur is typically undesired, it encodes the relative motion between the camera and the scene. This characteristic can be advantageous when utilized to construct high frame-rate videos. Therefore, the proposed method is the *Event-based Double Integral* method, which offers the following main contributions:

1) Restoration of a high frame-rate sharp video from a single image and its corresponding event data.
2) A stable and general method for generating a sharp video under various types of blur by solving a single-variable non-convex optimization problem.
3) The reconstructed video can theoretically have a frame rate as high as the event rate.

A. Event Camera Model

Event cameras are sensors that report logarithmic intensity changes and trigger events whenever the changes in intensity at a given pixel exceed a threshold. If nothing moves in the scene, no events are triggered. In the theory of event cameras, the concept of a latent image is present, denoted as:

$$l_{xy}(t) \tag{1}$$

which represents the instantaneous intensity at pixel (x, y) at time t.

However, the latent image is not directly outputted by the event camera. Instead, the camera outputs a sequence of *events* denoted by (x, y, t, σ), where (x, y) are the image coordinates, t is the time of the event, and $\sigma = +1$ or -1 denotes the polarity, which indicates the direction, whether it is an increase or a decrease, of intensity change. Polarity is determined by the following function:

$$\sigma = \tau\left(\log\left(\frac{l_{xy}(t)}{l_{xy}(t_{\text{ref}})}\right), c\right) \tag{2}$$

where $\tau(\cdot,\cdot)$ is a truncation function described as:

$$\tau(d\ ,\ c) = \begin{cases} +1, & d \geq c, \\ 0, & d \in (-c, c), \\ -1, & d \leq -c. \end{cases} \tag{3}$$

Here, c is a threshold parameter that determines whether an event should be recorded or not, and t_{ref} denotes the timestamp of the previous event.

B. Intensity Image Formulation

Event cameras can also provide a full-frame grey-scale intensity image, but at a much slower rate compared to the event sequence. These images can suffer from motion blur due to their long exposure time.

$$B = \frac{1}{T} \cdot \int_{f-\frac{T}{2}}^{f+\frac{T}{2}} L(t) \cdot dt \tag{4}$$

A general model of the image is given by the equation above, where B is a blurry image, equal to the average value of latent images during the exposure time between $f - \frac{T}{2}$ and $f + \frac{T}{2}$, where f is the frame and T is the change in time.

C. Event-based Double Integral Model

The aim is to exploit both the blur model and the event mode in this implementation. We start by defining a function that detects whenever there is an event:

$$e_{xy}(t) = \sigma \cdot \delta_{t_0}(t) \tag{5}$$

Where an event is defined by:

$$(x, y, t_0, \sigma)$$

Here, $\delta_{t_0}(t)$ is an impulse function with unit integral at time t_0 and the sequence of events is turned into a sequence of impulses over continuous time. During an exposure period $[f - \frac{T}{2}$ and $f + \frac{T}{2}]$, we define $E(t)$ as the sum of events between f and t at a given pixel, representing the proportional change in intensity:

$$E(s) = \int_f^t e(s) ds \tag{6}$$

Except under extreme conditions, the latent image sequence $L(t)$ is expressed as:

$$\tilde{L(t)} = \tilde{L(f)} + c \cdot E(t) \tag{7}$$

Note that in Eq. 9, the tilde above the latent image function refers to it's log. Given a sharp frame, a high frame-rate video can be reconstructed from the sharp starting point $L(f)$.

When the input image is blurry, a trivial solution would be to first deblur the image with an existing deblurring method and then reconstruct the video using the previous equation. However, in this way, the event data between intensity images is not fully exploited, resulting in inferior performance. Instead, a video reconstruction is done by exploiting the connection between event and blur:

$$B = \frac{1}{T} \cdot \int_{f-\frac{T}{2}}^{f+\frac{T}{2}} L(t) \cdot dt = \frac{L(f)}{T} \int_{f-\frac{T}{2}}^{f+\frac{T}{2}} \exp\left(c \int_f^t (e(s) \cdot ds)\right) dt \tag{8}$$

This equation is called the *Event-based Double Integral (EDI) model*. Another version of the equation is denoted as:

$$\tilde{L(f)} = \tilde{B} - \log\left(\frac{1}{T} \int_{f-\frac{T}{2}}^{f+\frac{T}{2}} \exp(c \cdot 1E(t)) dt\right) \tag{9}$$

Which shows a linear relation between the blurry image, the latent image and the integral of the events in the log space. The right-hand side of the equation is known except for c, and the first term comes from the grey-scale image while the second from the event sequence.

D. High Frame-Rate Video Generation

The latent image $L(t)$ can be computed at any time, which is an advantage when trying to reconstruct videos. To avoid accumulated errors of constructing a video from many frames of a blurred video, it is more suitable to construct each frame $L(t)$ using the closest blurred frame. In the paper, a reconstructed frame is generated every 50-100 events, so for their experiments, the frame-rate of the reconstructed video is usually 200 times greater than the input low frame-rate video.

E. Python Implementation

The script starts by defining a class called 'EventDataProcessor', which encapsulates the event data processing and image manipulation tasks. The initializer imports necessary data by reading an Excel file and converting it to a Pandas data frame. It is important to note that before importing all the frames, they were normalized by the global maximum and minimum of all the frames. This is to ensure that the pixel values in the image are on a consistent scale.

Next, based on the inputs of the desired start and end time, a window of events is created and processed for each frame. Following the formulation in Eq. 4, the window is defined as being between half the time before and half the time after the frame of interest. The mathematical operations are then applied to the event frames and are visualized. This is done by creating an empty image matrix and then iterating over each row in the data frame and updating the corresponding pixel in the image matrix based on the event polarity.

Finally, the resultant event frame is subtracted from its intensity frame version. The intensity frame is logged prior to the subtraction as indicated in Eq. 9. By applying a logarithmic transformation to the intensity frame, it compress the dynamic range and enhances the visibility of details, making it easier to distinguish events from the background. The result is then exponentiated to bring the intensity values back to their original scale.

IV. Results & Discussion

After deblurring the image frames, we observed that the algorithm successfully identified the regions affected by blurring and provided a faint representation of those regions. The blurred pixels were enhanced and appeared lighter in the frames, indicating an improvement in visibility. This suggests that the algorithm effectively identified the blurring artifacts and restored the underlying details.

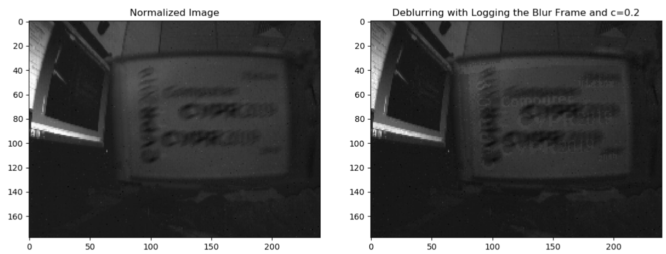
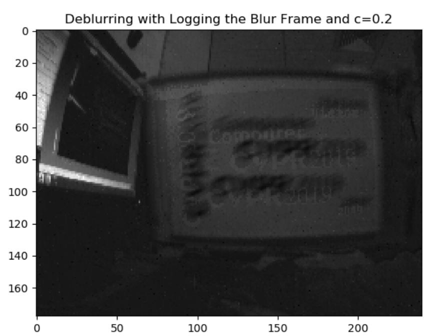
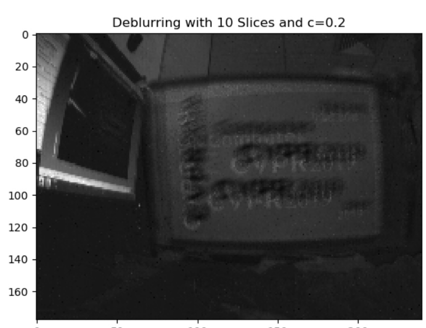


Fig. 2. Left: Original Frame. Right: Deblurred Frame

A. Slicing Approach

To further enhance the deblurring results, we introduced a slicing approach. By dividing the events into time intervals or slices, we aimed to extract and process events within smaller temporal segments. Within each slice, events were extracted based on their corresponding time range. This slicing approach allowed for a more targeted and localized analysis of the events.





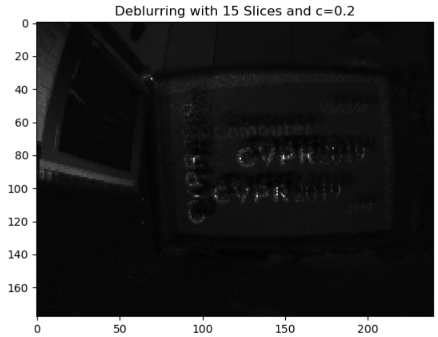


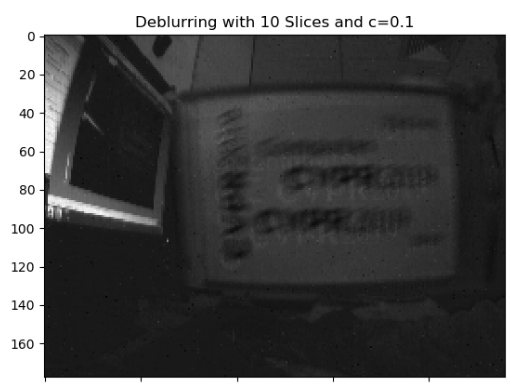
Fig. 3. First: No Slices. Second: 10 Slices. Third: 15 Slices

During the integration process, the polarity values of the events within each slice underwent the transformation. This helped emphasize the relevant features within the deblurred image and highlight the deblurred regions of interest. The transformed events were then visualized and added to an image matrix, which served as a cumulative representation of all the events. By iteratively integrating the events from each slice into the image matrix, the algorithm gradually constructed the deblurred image.

Increasing the number of slices had a notable impact on the deblurring results. As the number of slices increased, the deblurred regions became more prominent and distinct. This suggested that the slicing approach helped to enhance the visibility of the deblurred areas. However, with a higher number of slices, the background of the image became darker. This trade-off between enhancing the deblurred regions and darkening the background should be carefully considered based on the specific requirements.

B. Effect of c Parameter

In addition to the slicing approach, we experimented with adjusting the parameter c in the double integration formulation as seen in Eq. 9. Increasing the value of c further enhanced the deblurred regions, making them more pronounced.



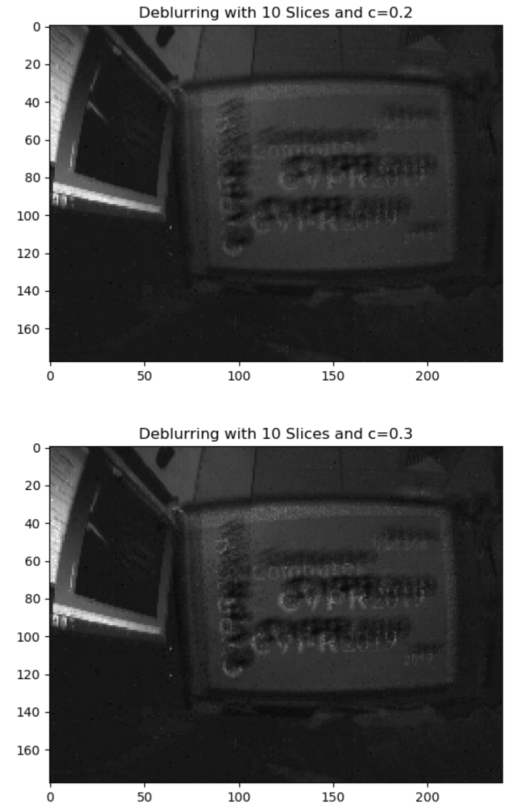


Fig. 4. First: c = 0.1. Second: c = 0.2. Third: c = 0.3

However, this enhancement came at the cost of introducing additional noise to the frames. It was observed that a low c value worked well for frames without blur, but had minimal impact on frames with blur. On the other hand, a high c value yielded good results for blur frames but introduced excessive noise in non-blur frames.

C. Time Shift

In our deblurring experiments, we explored the technique of shifting the time to improve the alignment between the deblur frames and the intensity images before subtracting them. This approach was inspired by the original paper and had shown promising results in the MATLAB version of the algorithm. However, we encountered challenges when attempting to replicate these results, as the alignment between the shifted frames and the intensity images did not meet our expectations.

The purpose of shifting the time was to ensure that the deblur frames and the intensity images were temporally aligned, allowing for a more accurate subtraction. By shifting the time, we aimed to compensate for any time discrepancies or synchronization issues between the two sources of data. This adjustment was expected to bring the corresponding pixels of the deblur frames and intensity images into alignment, facilitating a more effective subtraction process.

Although the original paper reported successful outcomes with the shifted time approach, we faced difficulties in achieving the same level of alignment. Despite manually shifting some pixels in an attempt to improve the alignment, we were unable to achieve the desired results. The shifted frames and intensity images did not align properly, leading to inaccurate subtractions and degraded deblurring outcomes.

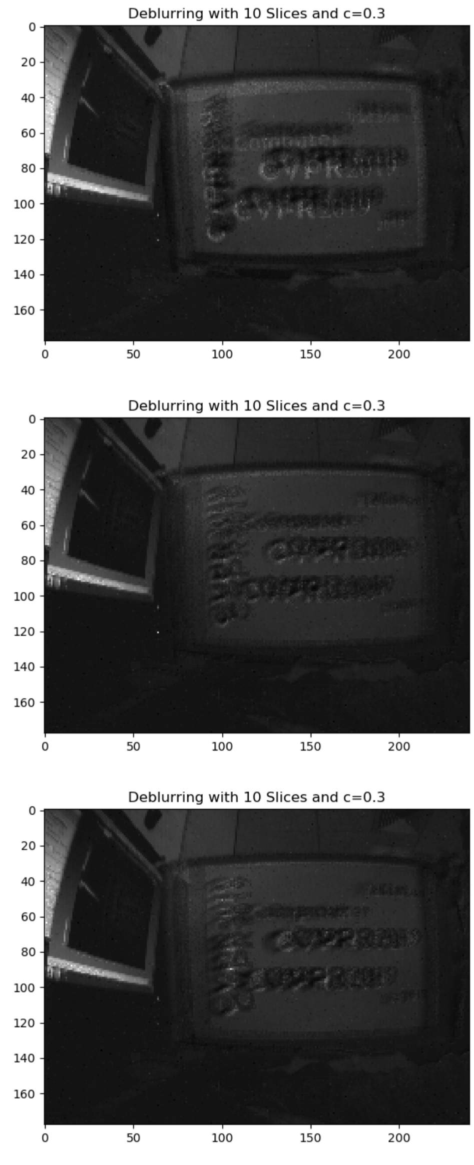


Fig. 5. First: No Time Shift. Second: 100k Shift. Third: 150k Shift

The lack of successful replication could be attributed to various factors. One possible reason is the difference in the implementation between the original MATLAB version and our own implementation. The discrepancy in the programming environment and the specific implementation details might have contributed to the observed misalignment.

V. Conclusion

In conclusion, the proposed Event-based Double Integral method showed a promising approach for deblurring using event data. By leveraging the advantages of event cameras and exploiting the relationship between events and blur, the EDI method enabled the restoration of high frame-rate sharp videos from single blurry images. The algorithm offered stable and general deblurring capabilities by solving a single-variable non-convex optimization problem. The results presented improved visibility of deblurred regions and highlighted the effectiveness of the slicing approach. However, we faced challenges in replicating the results obtained by shifting the time to align the deblur frames with intensity images. These difficulties might be attributed to implementation differences and specific details of the algorithm.

VI. References

[1] L. Pan et al., “Bringing a Blurry Frame Alive at High Frame-Rate with an Event Camera,” Computer Vision Foundation, [PDF Link] (accessed Jun. 9, 2023).