This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

Asynchronous Event-based Fourier Analysis

Q. Sabatier[1,2], S.H. Ieng[1], R. Benosman[1].

***Abstract*—This paper introduces a method to compute the FFT of a visual scene at a high temporal precision of around 1 µs output from an asynchronous event-based camera. Event-based cameras allow to go beyond the widespread and ingrained belief that acquiring series of images at some rate is a good way to capture visual motion. Each pixel adapts its own sampling rate to the visual input it receives and defines the timing of its own sampling points in response to its visual input by reacting to changes of the amount of incident light. As a consequence, the sampling process is no longer governed by a fixed timing source but by the signal to be sampled itself, or more precisely by the variations of the signal in the amplitude domain. Event-based cameras acquisition paradigm allows to go beyond the current conventional method to compute the FFT. The event-driven FFT algorithm relies on an heuristic methodology designed to operate directly on incoming gray level events to update incrementally the FFT while reducing both computation and data load. We show that for reasonable levels of approximations at equivalent frame rates beyond the millisecond, the method performs faster and more efficiently than conventional image acquisition. Several experiments are carried out on indoor and outdoor scenes where both conventional and event-driven FFT computation is shown and compared.**

I. Introduction

CONVENTIONAL imaging devices sample scenes at a fixed frequency; all pixels acquire luminance simultaneously by integrating the amount of light over a fixed period of time. Often only very few pixels change between two consecutive frames, leading to the acquisition of large amounts of redundant data. Often only very few pixels change between two consecutive frames, leading to the acquisition of large amounts of redundant data. When a conventional frame-based camera observes a dynamic scene, no matter where the frame rate is set to, it will always be wrong because there is no relation whatsoever between dynamics present in a scene and the chosen frame rate, over-sampling and/or under-sampling occur, and moreover both usually happen at the same time. When acquiring a natural scene with a fast moving object in front of static background with a standard video camera, motion blurring and displacement of the moving object between adjacent frames will result from under-sampling the fast motion, while repeatedly sampling and acquiring static background over and over again. This will lead to large amounts of redundant, previously known data that do not contain any new information. As a result, the scene is simultaneoulsy under- and over-sampled. This strategy of acquiring dynamic visual information has been accepted by the machine vision community for decades, likely due to the lack of convincing alternative.

An alternative to fixed-frequency is to sample a time-varying signal not on the time axis but using its the amplitude axis, leading to non uniform sampling rates that match the dynamics of the input signal. This sampling approach is often referred to as asynchronous delta modulation [1] or continuous-time level-crossing sampling [2]. Recently, this sampling paradigm has advanced from the recording of 1-D signals to the real-time acquisition of 2-D image data. The asynchronous time-based image sensor (ATIS) described in [3] contains an array of autonomously operating pixels that combine an asynchronous level-crossing detector and an exposure measurement circuit. Each exposure measurement by an individual pixel is triggered by a level-crossing event measuring a relative illuminance change at the pixel level. Hence, each pixel independently samples its illuminance, through an integrative measurement, upon detection of a change of a certain magnitude in this same illuminance, establishing its instantaneous gray level after it has changed. The result of the exposure measurement (i.e., the new gray level) is asynchronously transmitted off the sensor together with the pixels xy-address. As a result, image information is not acquired frame-wise but conditionally only from parts in the scene where there is new information. Only information that is relevant — because unknown — is acquired, transmitted and processed. Fig.1 shows the general principle of asynchronous imaging spaces. Frames are absent from this acquisition process. They can however be reconstructed, when needed (e.g. for display purposes), as shown at the top part of Fig.1 and at frequencies limited only by the temporal resolution of the pixel circuits (up to hundreds of kiloframes per second). Static objects and background information, if required, can be recorded as a snapshot at the start of an acquisition henceforward moving objects in the visual scene describe a spatio-temporal surface at very high temporal resolution shown in the bottom part of Fig.1 . This novel paradigm of visual data acquisition calls for a new methodology in order to efficiently process the sparse, event-based image information without sacrificing its beneficial characteristics. Several methods have been recently published, which outperform conventional approaches both in computational costs and robustness. These cover all topics of machine vision: stereovision[4], [5], [6], [7], object recognition[8], [9], optical flow[10], [11], robotics[12], [13], [14], [15], tracking[16], [17] image processing[18] and retina prosthetics[19], [20].

This paper contributes to the field of asynchronous event-based vision by proposing an algorithm that computes the

1 Sorbonne Universités, UPMC Univ Paris 06; UMR_S 968, Institut de la Vision, Paris, F-75012, France; CNRS, UMR_7210, Paris, F-75012, France.

2 Gensight Biologics, 74 Fbg Saint Antoine, 75012 Paris, France.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org.,provided by the author. The material includes a video file "Car5thresholds.avi". Contact ryad.benosman@upmc.fr for further questions about this work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

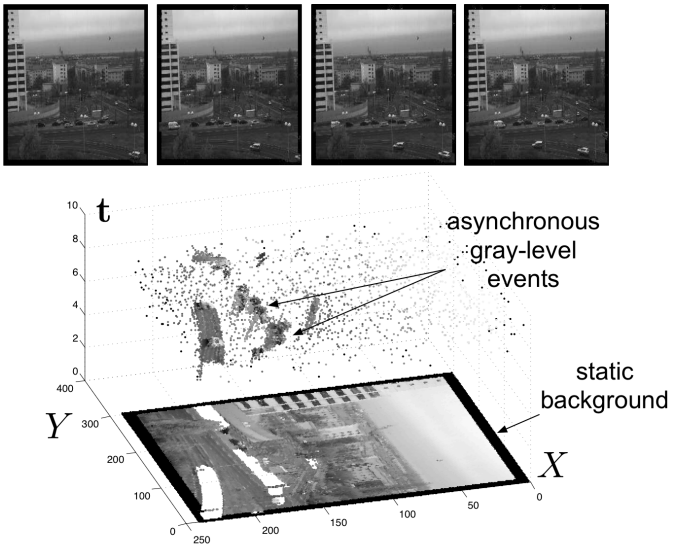


Fig. 1. The spatio-temporal space of imaging events: Static objects and scene background are acquired first. Then, dynamic objects trigger pixel-individual, asynchronous gray level events after each change. Frames are absent from this acquisition process. Samples of generated images from the presented spatio-temporal space are shown in the upper part of the figure.

spatial Discrete Fourier Transform (DFT) iteratively for each incoming high temporal resolution event (1 µs time precision). The method computes the exact spatial DFT on event-based visual signals, without the need to reprocess already acquired information. This work also extends the event-based formulation of the DFT by introducing a lossy transformation methodology that can reduce even more computations by estimating a trade off between the quality of a reconstructed signal and the processing time. The time-varying visual signals are provided by the Asynchronous Time-based Image Sensor (ATIS) [3]. Conventional frame-based algorithms cannot be applied unchanged to event-based representation without leading to an immediate loss of the benefits inherent to the new sensing paradigm. Discrete Fourier Transform (DFT) is widely used in digital signal processing and scientific computing applications. The two-dimensional (2D) DFT is used in a wide variety of imaging applications that need spectral and frequency-domain analysis. The image sizes of many of the applications have increased over the years reaching 2048 $\times$ 2048 in synthetic aperture radar image processing [21], digital holographic imaging [22]. Existing 2D DFT implementations include software solutions, such as FFTW [23], Spiral [24], Intel MKL [25] and IPP [26] which can run on conventional computers, multicore architecture [27], or supercomputers [28]. There are several hardware solutions using the dedicated FFT processor chips [22],[29],[30],[31],[32] and field programmable gate array (FPGA) based implementations [33],[33],[34][35],[36],[37]. These implementations are efficient, however they are incompatible with an asynchronous event based acquisition, as only a fraction of the signal changes over a short time period, therefore applying the FFT on the whole signal would not be efficient and would not make a full use of the advantages of this acquisition process.

State of the art level-crossing sampled signals and more generally stochastic sampled signals (not complying with the Shannon sampling theory) have been studied with the goal of achieving 1D signals accurate reconstructions [38] and frequency analysis (e.g. filtering and Fourier-like transformations) [39], [40], [41], [42]. This work differs from that topic of research because we are dealing for the first time, with the computation of spatial Fourrier transforms of a dynamic scene acquired using an asynchronous event-based image sensor.

II. The Asynchronous Time-based Image Sensor

The ATIS used in this work is a time-domain encoding image sensor with 240×304 pixel resolution [3]. The sensor contains an array of fully autonomous pixels that each combines an illuminance change detector circuit and a conditional exposure measurement block. As shown in the functional diagram of an ATIS pixel in Fig.2, the change detector individually and asynchronously initiates the measurement of an exposure/gray level value only if — and immediately after — a brightness change of a certain magnitude has been detected in the field-of-view of the pixel at time t_0. The ATIS encodes visual information as a stream of events. An event is a set $\{type, \mathbf{p}, t_0, pol\}$: where $type$ is the flag signaling a change, or a gray level event, $\mathbf{p} = (x, y)^T$ the spatial coordinate, pol the polarity, and t_0 the time of occurence. The polarity pol has two meanings according to the event's type. For a change event, the polarity encodes the increase or the decrease of the luminance. For a gray level measurement mechanism it differentiates between the two events encoding the temporal measurment of luminance. Luminance in our case is encoded by a pair of gray level events such that the inverse I of the time difference between the two is proportional to the luminance (as shown in Fig.2). The linear correspondance between the measured timing and the absolute luminance value is set by design. The first gray level event is triggered right after the change event at $t_1 \sim t_0$ and the second at t_2 such that :

$$I \propto \frac{1}{t_2 - t_0}. \tag{1}$$

Readers are advised to refer to [3] for further details about the ATIS.
Since we are focusing on the luminance information at time t_0, we define a simplified event e_{cam} as:

$$e_{cam}(\mathbf{p}, t_0) = \{I, \mathbf{p}, t_0\} \tag{2}$$

This integration duration only depends on the measured gray level intensity, not on the time t at which it started. Since the ATIS is not clocked like conventional cameras, the timing of events can be conveyed with a temporal resolution of the order of 1 μs. The time-domain encoding of the intensity information automatically optimizes the exposure time separately for each pixel instead of imposing a fixed integration time for the entire array. This results in an exceptionally high dynamic range of $143\,\mathrm{dB}$ and an improved signal to noise ratio of $56\,\mathrm{dB}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

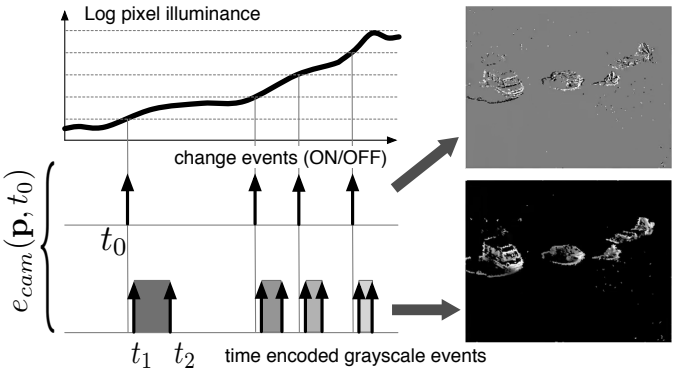


Fig. 2. Functional diagram of an ATIS pixel. Two types of asynchronous events, encoding change and brightness information, are generated and transmitted individually by each pixel in the imaging array.

III. Design and representation of event-based algorithms

A. General framework

The event-driven acquisition and the absence of a global sampling frequency radically changes the signal representation and update, compared to the conventional frame-based representation. Only few components of the acquired signal are updated when a change happens. This implies that to benefit from the high temporal accuracy algorithms should be event-driven. This means that processing must only be carried out when an event is acquired in the same spirit to what has been presented in [18]. The whole chain of processing is event driven meaning that an iterative computation must be developed and updated for each incoming event.
The solution proposed in this work is reached in two-stages: a first event-driven naive formulation is built from the standard definition of the DFT. This form, as we will show, is exact but not optimal since a single event updates all Fourier coefficients. A second form is then derived based on the Stockham algorithm [43], [44] using a decomposition into sparse matrices of the DFT operator. This decomposition, combined with the event-based formulation, allows to achieve a lossy DFT that discards non significant events from the DFT computation, hence enabling a strategy based on trade-off between accuracy and computation time.

B. The naive approach: a simple algorithm to compute the Discrete Fourier Transform

The first step in designing event-based algorithms is to study the impact of sparse event-based updates of an acquired signal and its implications on the computation of the Discrete Fourier Transform. Let's consider a real valued signal $\mathbf{s}(n,t)$ which is a function of space, indexed by n, and of time t. We denote $\mathbf{S}(k,t) = \mathcal{F}[\mathbf{s}(n,t)]$ its spatial DFT computed at t :

$$\begin{array}{rccc} \mathcal{F}: & \mathbb{R}^N & \longrightarrow & \mathbb{C}^N \\ & \mathbf{s}(n,t) & \longmapsto & \mathbf{S}(k,t) = \mathcal{F}[\mathbf{s}(n,t)]. \end{array} \tag{3}$$

$\mathcal{F}$ is the N-point DFT operator with $N \in \mathbb{N}^*$ (set of non null natural numbers) applied to the spatial components of $\mathbf{s}$. To ease understanding, we develop the methodology for spatially unidimensional signals. The same methodology can be extended to multidimensional signal. Experimental results on 2D signals will be presented in section V.
The DFT of $\mathbf{s}$, referred to as the analysis equation, is defined for integer k satisfying $0 \leq k \leq N-1$ or equivalently $k \in [\![0, N-1]\!]$:

$$\mathbf{S}(k,t) \coloneqq \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \mathbf{s}(n,t) \exp\left(-2i\pi \frac{nk}{N}\right) \tag{4}$$

and the inverse transform, referred to as the synthesis equation is:

$$\mathbf{s}(n,t) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \mathbf{S}(k,t) \exp\left(+2i\pi \frac{nk}{N}\right) \tag{5}$$

Note that the definition of the DFT is not always the one we present here. We chose this convention for the normalization factor because it enables us to keep the same rules for forward and inverse transforms.

Using this normalization convention, Plancherel's theorem, applied to the spatial component of $\mathbf{s}$ at a given t and to its spatial Fourier transform, is:

$$\forall t, \|\mathbf{s}(n,t)\|_2 = \|\mathbf{S}(k,t)\|_2, \tag{6}$$

where $\|.\|_2$ is the Euclidean norm in each respective space. Because of that property, the same threshold, referred to as the "significance threshold T", can be used both in the focal plane and in the frequency space. This thresholding mechanism is the basis of the idea developed in section IV.

When an event $e_{cam}(q,t)$ is detected and acquired it implies that a single component of the input signal s at location q changes significantly. Here, we are considering a 1D case to ease the notation, therefore q is a scalar representing the index corresponding to the location where the change occurs. We denote $\boldsymbol{s}(n,t)$ the acquired signal, at time t. In order to disambiguate the value of the acquired signal at the event times, we use the following notations :

$$\forall p \in [\![0, N-1]\!], \begin{cases} \boldsymbol{s}(p,t^-) & \coloneqq & \lim\limits_{u \to t^-} \boldsymbol{s}(p,u) \\ \boldsymbol{s}(p,t^+) & \coloneqq & \lim\limits_{u \to t^+} \boldsymbol{s}(p,u) \end{cases}$$

The acquired input signal is then updated at its component q, so that $e_{cam}(q,t) = \{\boldsymbol{s}(q,t^+), q, t\}$. The values of the acquired signal right before and right after the event $e_{cam}(q,t)$ are related by :

$$\forall p \in [\![0, N-1]\!], \boldsymbol{s}\big(p,t^+\big) = \boldsymbol{s}\big(p,t^-\big) + \alpha\delta_{p,q} \tag{7}$$

where δ is the Kronecker delta and α is the difference between the old value and the new value of the signal.
We can establish the relation between $\boldsymbol{S}(k,t^+) \coloneqq \mathcal{F}[\boldsymbol{s}(n,t^+)]$ and $\boldsymbol{S}(k,t^-) \coloneqq \mathcal{F}[\boldsymbol{s}(n,t^-)]$:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

$\forall k \in [\![0, N-1]\!]$,

$$\begin{aligned}
\boldsymbol{S}(k,t^+) &= \tfrac{1}{\sqrt{N}} \sum_{p=0}^{N-1} \boldsymbol{s}(p,t^+) \exp\Big(-2i\pi \tfrac{pk}{N}\Big) \\
&= \tfrac{1}{\sqrt{N}} \sum_{p=0}^{N-1} [\boldsymbol{s}(p,t^-) + \alpha\delta_{p,q}] \exp\Big(-2i\pi \tfrac{pk}{N}\Big) \\
&= \tfrac{1}{\sqrt{N}} \sum_{p=0}^{N-1} \boldsymbol{s}(p,t^-) \exp\Big(-2i\pi \tfrac{pk}{N}\Big) \\
&\qquad + \tfrac{\alpha}{\sqrt{N}} \exp\Big(-2i\pi \tfrac{qk}{N}\Big) \\
&= \boldsymbol{S}(k,t^-) + \tfrac{\alpha}{\sqrt{N}} \exp\Big(-2i\pi \tfrac{qk}{N}\Big),
\end{aligned} \tag{8}$$

This is showing that every term of the DFT has to be updated with an increment with the same module $|\alpha|/\sqrt{N}$, and a phase which depends on the indices of both the pixel component and the Fourier component. Consequently, updating the exact Fourier spectrum after one event on the camera requires a number of operations linear with the number N of samples. It is straightforward to conclude that no exact iterative method can be implemented in less than $\mathcal{O}(N)$ operations per event.

Finally, it suggests that no approximation can be made *a priori*. Since all components should be updated by an increment of the same amplitude, we can not know which ones to favor and which ones to exclude (except possibly for a specific application for which a fraction of components are more important than others, which is not the case here as we intend to provide a general method).

The algorithm consists in three successive blocks as represented in Fig.3. The sensor is delivering events $e_{cam}(q,t)$ to each node of the DCT block. Each node then uses its index $k \in [\![0, N-1]\!]$ to compute the value of the update, $\frac{\alpha}{\sqrt{N}} \exp\Big(-2i\pi \frac{qk}{N}\Big)$ that is is added to the previous value of the k^{th} Fourier component $\boldsymbol{S}(k,t^-)$ to compute the new value of the component $\boldsymbol{S}(k,t^+)$. This value replaces the previous one in memory and an is output by that third layer. It is important to notice that the rate of events (i.e. the number of events triggered/ processed per unit of time by each layer) is multiplied by N at this stage, where N is the number of spatial samples of the input signal.

Remarks :

The algorithm has two main limitations. The first one is that the number of operations carried out per event is linear with the number of pixel of the sensing device. As an example, the ATIS camera has $240 \times 304 = 72960$ pixels. Performing that much operations per event is too resource demanding and does not make full use of the event-driven properties of the sensor.

The second limitation is that it also increases the rate of generated events by a factor 72960, because for each event output by the ATIS, each node representing a Fourier component triggers an event containing its new value.

It is important to emphasize that the terminology of "event" applies for all the entire processing chain, there are two types of events, those provided by the ATIS are actual measurements of the absolute luminance, the remaining ones are purely computational output by each layer.

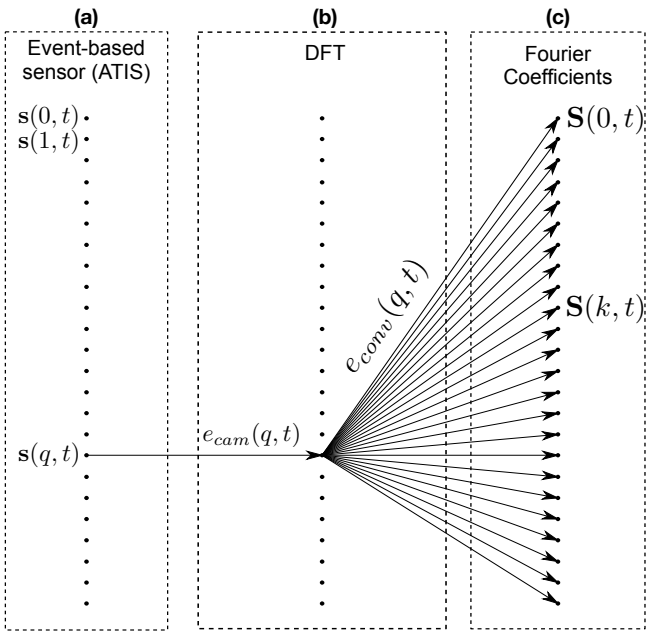


Fig. 3. Graphical representation of the naive event-based algorithm (unidimensional case, $N = 24$). (a) the gray level events are generated from the ATIS. They are connected in a one-to-one manner to the converter (b) which computes the difference between two successive events. The nodes of the converter implement the computation of variable I of (7). The Converter layer is connected in an all-to-all fashion to the Fourier layer (c). The Fourier nodes implement the computation introduced in (8).

IV. Event-based Discrete Fourier Transform

A better methodology is to use a decomposition of the matrix representing the DFT operator into a product of sparse matrices. We can apply Stockham's algorithm [43], [44] that computes the DFT in $\mathcal{O}(N \times \ln(N))$ operations, and has both the input and output signals sorted in the natural order.

The computation of the DFT using this technique is equivalent to building a network consisting of several layers, each containing the same number of nodes which is also the number of samples of the input signal and storing the result of intermediate calculations. If we denote M the matrix representing the DFT operator (i.e. $\mathcal{F}[\boldsymbol{s}(.,t)] = M\boldsymbol{s}(.,t)$), FFT algorithms provide us with decompositions of the matrix M as a product of L sparse matrices, where L is of the order of $\ln(N)$:

$$M = M^L \dots M^2 M^1 \tag{9}$$

Given an input signal which is stored in the nodes of the first layer, the values of the nodes of the following layers are equal to a weighted sum of the nodes in the previous layer which connect to it. A weight is associated with each edge. Computation of the connections and associated weights within the network can be performed based on the knowledge of the sequence of factors chosen to build the network. The signal contained in each layer is obtained as a linear combination of the signal contained in the previous layer. A

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

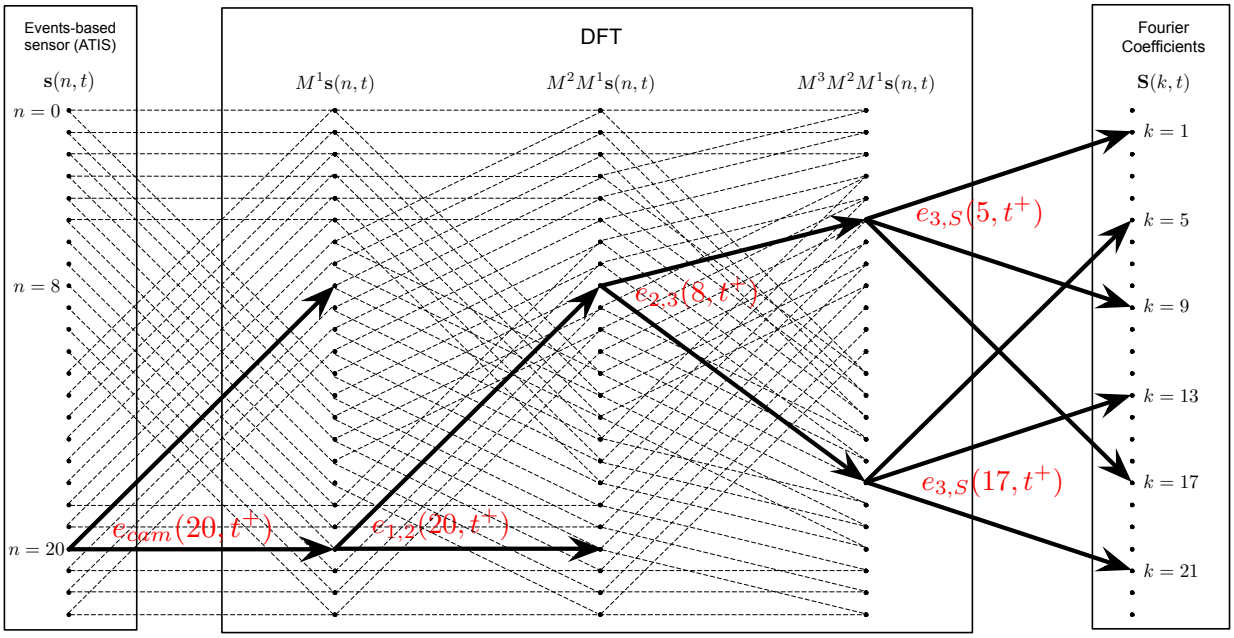


Fig. 4. A graph used for the computation of a unidimensional dynamic signal with 24 samples. The values of the input signal are stored in the leftmost nodes. Edges show the relationship between the values contained in the nodes of the network. The value contained in a node is a linear combination of the values contained in the nodes of the previous layer (i.e. to the left) connected to the node through an edge. The weights in the linear combination depends on the edge, but are omitted in the figure. The solid edges show the path of an update from input node number 20 to all output nodes.

network is equivalent to a sequence of matrices, where each matrix is associated with the transition from one layer to the next. We can compute the number of operations — defined as a multiplication of a complex by a complex exponential followed by a complex addition — such propagation requires. Each edge accounts for one such operation. Fig.4 shows such a network for a unidimensional input signal, with $N = 24$. It is built using the decomposition $24 = 2\times2\times2\times3$, an update would require $2+4+8+24 = 38$ operations. This is an expected result as updating exactly the Fourier representation requires at least N operations. Unsurprisingly, there is a path from each node of the input layer to all nodes of the output layer.

To make full use of the event-driven acquisition we ensure that an incoming event introduces a significant change to the signal. Instead of sending an event each time a value is updated, events are only propagated if the update they communicate is significant. As shown in Fig.4 an incoming event from the event-based camera $e_{cam}(20, t^+)$ from node 20 is sent to the layer 1 of the DFT block, because the amount of change it provides is larger than a percentage of the previously received value of the $\boldsymbol{s}(20, t^-)$. Otherwise this new information is stored locally by updating the value of the node until the amount of the change is significant enough with respect to a fixed threshold T. A example of successful propagations of the signal changes within the DFT block can be depicted as follows:

- The ATIS outputs an event $e_{cam}(20, t^+)$ at its node 20. This event is transferred to nodes 8 and 20 of the first layer of the DFT block for its first layer processing.
- Out of the new values at nodes 8 and 20 in the 1st layer, only the one for node 20 is supposed to be significant enough. This triggers the event $e_{1,2}(20, t^+)$ that is sent to nodes 8 and 20 of the layer 2 for the 2nd layer processing.
- Now only change at node 8 of layer 2 is supposed to be significant. Due to the same mechanism, event $e_{2,3}(8, t^+)$ is generated at node 8 and transmitted to nodes 5 and 17 of the last layer for processing.
- Finally, the new values at nodes 5 and 17 of the last layer are significantly larger that the previous stored values, this triggers events $e_{3,S}(5, t^+)$ and $e_{3,S}(17, t^+)$ that are updating some Fourier Coefficients $\mathbf{S}(k, t^+)$.

All connections corresponding to a full computation are shown in dashed lines (for the last block providing Fourier coefficients, these connections are not shown to emphasize the sparsity of the event-based DFT algorithm and to preserve readability). The event-based thresholding optimization pathway of information triggered by a single incoming event is displayed as plain arrows.

This process introduces an approximation in the computation of the DFT, since the output of the Fourier layer is not the exact DFT of the acquired signal but a signal which is considered to be so close that the difference between the exact and the approximate signals is not worth communicating. Fig.5 shows an experimental distribution of the number of Fourier components for an outdoors urban complex scene which are updated in a significant manner each time the input signal is updated significantly for a local change of 1%. The number of operations saved using this heuristic algorithm will be measured practically in the experiments section, as expected there will be a tradeoff of quality versus number of operations imposed by the chosen threshold.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

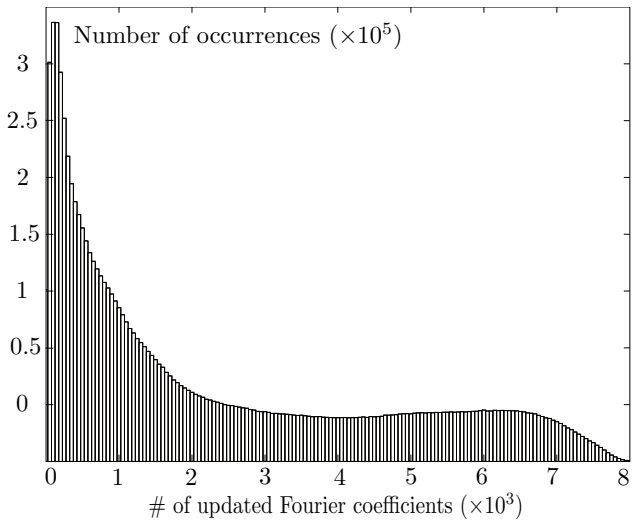


Fig. 5. Histogram of updated Fourier coefficients from the naive DFT. We performed the event-based Fourier algorithm on a set of natural scenes videos recorded with the ATIS. For each input event representing an update of more than 1% of the dynamic range, we counted the number of Fourier components which were updated by an increment superior to the same threshold. The worse condition happens when all the Fourier coefficients are updated despite of the significance thresholding, in that case, the number of updated coefficient is equal to the number of pixels in the sensor i.e. 72960.

V. Experiments

A. Methods

1) Implementation: Recorded sequences of events from the event-based cameras are used. We used the maximum number of layers by decomposing height and the width of the frame into prime numbers :

$$\begin{cases} 240 &= 2^4 \times 3 \times 5 \\ 304 &= 19 \times 2^4 \times 1 \end{cases}$$

Fig. 6. Snapshots from the 8 event-based sequences used for the experiments.

2) Image comparison: The quality of transforms is assessed using the Mean Structural SIMilarity index (MSSIM) introduced in [45] computed every 10 ms. The Structural SIMilarity (SSIM) index is a full-reference measure of similarity between two thumbnail images of 11×11 pixels. Larger images are compared by building 11×11 neighborhoods centered in each pixel. The Mean of the SSIM value is computed over all these possible neighborhoods in the image. This method is suited to measure image distortions, hence it is used to assess to which extent our method is able to maintain the structure of the input signal with respect to the approximations made.

The index is a combination of three measurements : (i) a luminance similarity index, (ii) a contrast similarity index and (iii) a structure similarity index based on the normalized correlation between the two thumbnails. The values for the different parameters were all set using the values recommended in [45]. The index in this DFT context is:

$$\mathrm{MSSIM}(t) = \mathrm{MSSIM}\Big(\boldsymbol{s}(n,t), \tilde{\mathcal{F}}^{-1} \circ \tilde{\mathcal{F}}[\boldsymbol{s}(n,t)]\Big), \qquad (10)$$

where $\tilde{\mathcal{F}}$ is the approximation of the DFT operator resulting from our algorithm.

3) Computation time: Conventional frame-based FFT algorithms take advantage of the structure of the Fourier basis such as the symmetries and periodicities of the trigonometric functions while event-based acquisition implies that the number of operations depends on the the signal. Consequently, our assessment of the results will be experimental based on the statistics of recorded scenes. We ensured a wide variety of indoor and outdoor scenes. We recorded the number of operations required to process each acquired sequence for a range of threshold over the relative change of the signal values : 0%, 1%, 2%, 5%, 10% and 20%.

As shown in section IV (compare Fig.3 and Fig.4) the approximation algorithm introduces additional operations when computing the exact Fourier transform. Consequently, our goal is to be more efficient than the naive event-based and the conventional frame-based FFT (when dealing with high frame rates) when using the heuristic approach in terms of computational time.

The number of operation for the frame-based DFT is proportional to the frame rate while the event-based DFT is proportional to the number of measured events. There is no direct and obvious relation between the two numbers. Our only way to determine which of the DFT techniques is requiring the less number of operations for a given quality of reconstruction is to test for different values of T, the DFT algorithms applied to one of our sequences. The comparison is scene dependent as it is shown in section V-A4, with the moving vehicle sequence.

4) Results: We present the results obtained for indoor and outdoor scenes shown in Fig.6. We first consider the most complex recorded scene corresponding to a dynamic urban scene where the camera in mounted inside a moving vehicle (the first row of Fig.6).

Results of recomposing the output after the heuristic approach for different significance threshold values are shown

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

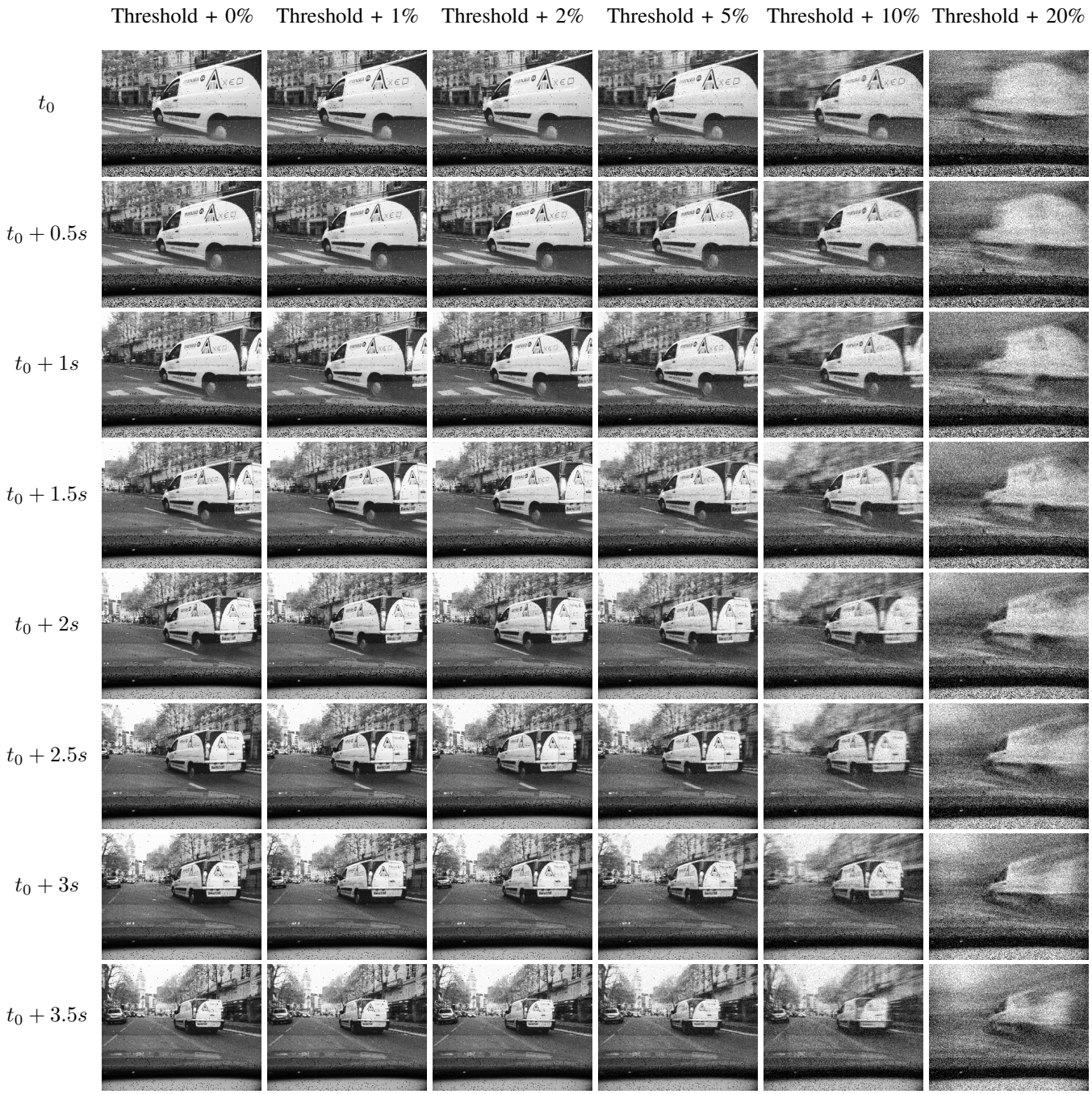


Fig. 7. "Moving vehicle" sequence: signals obtained after computing the Discrete Fourier Transform and its inverse using the algorithm presented in IV. The same sequence (leftmost column : Threshold 0%) is processed using different values for the significance threshold T, ranging from 1% to 20% of the dynamic range. Images are reconstructed using the flow of gray level events every half second.

in Fig.7. Up to a threshold of 5% of the dynamic range, the structure of the image is very well preserved as well as the details of the image. For a threshold of 10%, the structure is still preserved, but most of the details are lost. Finally, for thresholds higher than 20%, the structure of the resulting image is distorted. Only large objects are recognizable, but their details are lost and their shapes are also distorted. In particular, for a threshold above 10%, it appears that the spatial position of the van is slightly delayed in the image, there is a latency in the update of the spatial position due to the large threshold value. The higher the threshold, the less new incoming events will update the FFT, hence the delay can be large when we are comparing the output with frames generated at the same time. However the rate of events is also largely scene dependent (multi targets entering the sensor field of view, change of the relative speeds,...) and it can impact significantly the delay. At that stage, there is no straightforward way to keep tract of the delay w.r.t. the scene, hence we are evaluating the performance of the approach in the least favorable case where we are not taking the latency into account. Fig.8 provides the evolution the similarity index for all thresholds for the sequence. It can be noticed that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

similarity index value degrades to 0 as the threshold increases to 20%.

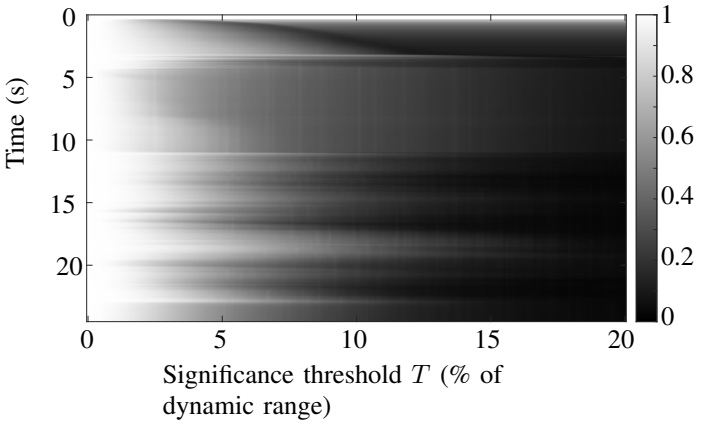


Fig. 8. Variations of the Mean SSIM index for the dynamic urban sequence, w.r.t. the threshold (in % of the dynamic range) and time.

Results plot in Fig.9 for all acquired sequences show that the quality index, MSSIM, and the average number of operation per event A, applied to the data are independant from the scenes' content. The threshold T is the parameter that sets the compromise between the signal transform quality and the computation used for the transform. Significant computational gains are obtained for threshold values in a range between 2.5% and 5%. As expected, the MSSIM and A functions have the same behaviour with respect to the threshold: they are decreasing functions of T. The proposed algorithm takes advantage of the fact that the MSSIM decreases at a much slower rate than A when T increases. Fig.9) that the the tangent to the MSSIM at zero is almost flat, while the slope of the tangent to A at zero is steep. This behavior allows us to find a threshold such that computations are significantly reduced with a low loss in signal quality. The combination of both functions allows to find a trade-off between the computation time and the quality of the transformed signal, as shown in Fig.10.

The PSNR (in dB) and the MSE (in gray level amplitude squared) of the reconstructed signal are also plotted in Fig.11 to show the impact of the threshold T. The gray levels measured by the ATIS are normalized by the highest value and rescaled so the gray levels values are between 0 and 255. These curves are substantiating the conclusion drawn from the MSSIM: the reconstructed signal quality is degraded quickly as T increases. For $T = 5\%$, the threshold value for which the tradeoff between MSSIM measured quality and computation time is becoming less interesting, the PNSR and the MSE are respectively $22.7dB$ and 346.

To compare the frame-based Fourier transform with the algorithm we introduced, we are estimating the number of operations per unit of time. This quantity is obtained by multiplying the average number of operations performed per event $A(T)$ by the number of events per unit of time. This value is estimated experimentally for the moving vehicle sequence and is equal to 10^6 events per second. Fig.12 shows the rate of operations w.r.t. T as a decreasing function when the sensor is static (blue dots) and when it is put in a car (red dots). Three dashed lines are plotted to show the number of

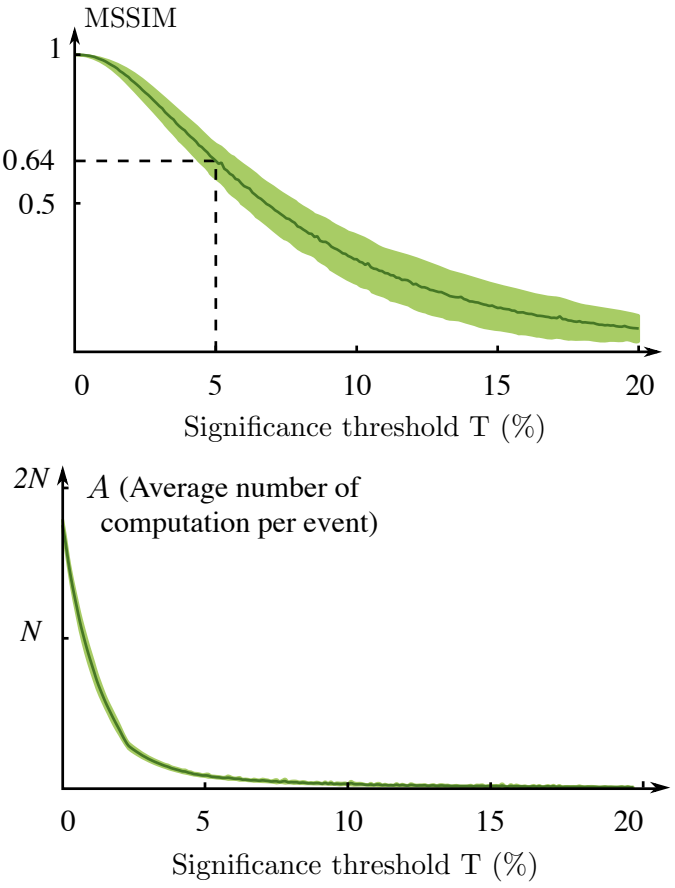


Fig. 9. MSSIM and average number of operations per events are decreasing functions of the threshold. one can however see the MSSIM is decreasing in a much slower rate.

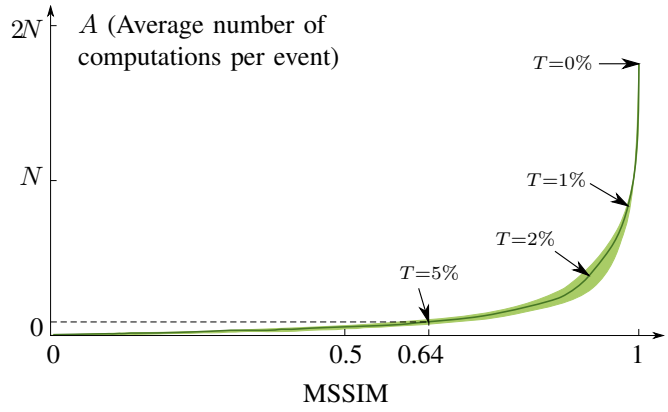


Fig. 10. Average number of operation per events w.r.t. the quality index. This graph emphasizes the slow increase rate of A when the MSSIM increases.

operations used by the frame-based algorithm at three different video sampling frequencies (100 Hz, 1 kHz and 1 MHz). At 1 kHz, for $T = 6\%$, both techniques require the same amount of operations. This amount decreases even more for the event-based algorithm if T increases.

As shown in Fig.12 and Fig.7, threshold values from 6 to 8% of the dynamic range are a compromise for computing with reasonable resource, the Fourier transform with respect to the loss of quality.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

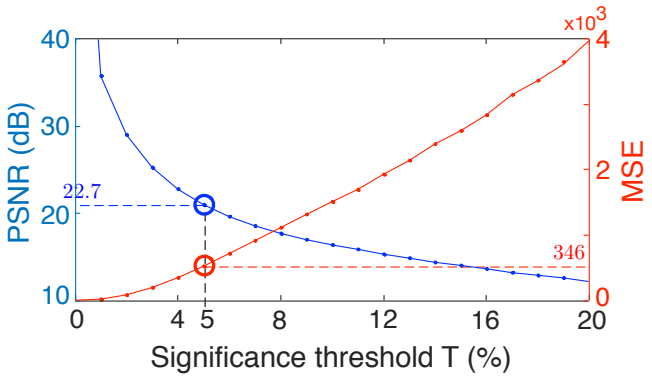


Fig. 11. PNSR and MSE as function of T. The PSNR decreases in a similar fashion as the MSSIM which is reflected by the increasing behavior of the MSE.

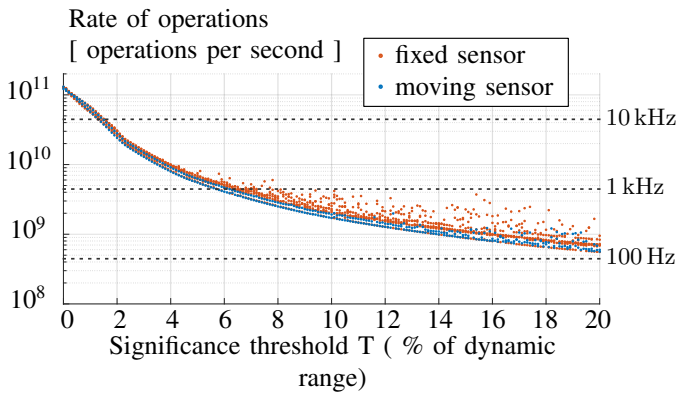


Fig. 12. Number of operations per unit of time carried out during our algorithm assuming a mean event rate of 10^6 events per second. Horizontal dashed lines show the rates of operation required by the frame-based FFT algorithm at different sampling frequencies : 100 Hz, 1 kHz and 10 kHz. The rate for the event-based algorithm is larger than the frame-based algorithm for low sampling frequencies up to 100 Hz. When sampling frequencies go up to the kHz, which corresponds to the typical µs precision of the asynchronous sensor, the event-based algorithm is more computation efficient for threshold values around 6-8% of the dynamic range.

VI. Discussion

As shown in the experimental section, the udpate process in a series of steps leads to significant gains in computational time. These gains conventionally imply a loss in accuracy because each intermediate layer filters out events that lead to small increments in all following steps. Interesting gains occur at low threshold values (less than 1% of the dynamic range) where significant low computational time are achieved while virtually no change can be detected in the signal. This shows that the heuristic algorithm determines which components should be updated. The algorithm does not require *a priori* choices regarding the components which should be updated, but rather bases the decision on the incoming signal. As the algorithm is intended to be applied to a wide range of areas which make use of level-crossing sampling, or more generally of asynchronous sampling, it was our choice not to make use of any prior knowledge in the design of the architecture of the system.

However, the algorithm could be adapted to benefit from prior knowledge. Two promising leads could be (i) adapting the thresholds so that more computational resources are allocated to components of higher interest, and (ii) re-arranging the connections in the network in order to filter out events in the earliest possible layers. Regarding the second point, the decomposition we used here is based on Fast Fourier Transforms decompositions of the Fourier operator. These decompositions result in minimal numbers of connections for a given number of samples and a given number of intermediate layers, which is a sensible approach to start with. However, neighboring pixels, which are the most likely to be related in a visual signal, do not connect to neighboring nodes in the first intermediate layer (see Fig.4).

We provided a trade-off (see Fig.10) as a characterization of our algorithm, and did not look for any optimal value for the threshold. Such value for the threshold heavily depends on the application for which it is used. A specific application would provide an objective function for both the computation time and the accuracy of expected results. The trade-off curve, or a similar one which would be produced for the given application, would then allow to turn the objective function of accuracy and computational time into a function of the threshold. Maximizing the resulting function with respect to the threshold would then provide the optimal threshold.

Considering the heuristic method, the perfect algorithm would provide a network and a behaviour for the nodes such that (i) the number of layers scales with $log(N)$ and (ii) the rate of events through each layer is kept constant on average. In such scenario, the number of operations carried out per event would scale with $log(N)$, and the corresponding algorithm would lead to the same improvement as the one provided by FFT algorithms in the frame-based setting. The fact that we filter events out suggests that the approximation method scales between $\mathcal{O}(log(N))$ and $\mathcal{O}(N)$.

Finally, increasing the threshold does not only allow computational gains within the Fourier filter, but it also decreases the rate of events output by the filter. This in turn reduces the computational burden of further processing steps.

VII. Conclusion

In this paper, we provided an algorithm to implement event-based Fourier transform algorithms. As the demand for higher temporal resolution increases, in particular for artificial visual tasks, the need for update methods which are able to operate on sparse data representation will be increasingly high. We showed that a promising lead is to develop heuristics which are able to regroup incoming information in order to detect as soon as possible, i.e. in the earliest possible layer, which part of the information is unnecessary to propagate to the following steps. As in the different frame-based FFT approaches, important work can be carried out by comparing decompositions of the Fourier operator matrix. However, the event-based framework introduces a major shift in the objective of the decomposition. The goal is not to find a decomposition which provides minimal number of edges in the associated network, but rather its ability to filter out useless information which in turn depends heavily on the statistical structure of the input signal, and of its dynamic. Our work introduces a general framework

showing that adding intermediate computation steps can help reducing the computational burden with minimal degradation of the underlying signal.

ACKNOWLEDGMENT

This work received financial support from the LABEX LIFESENSE [ANR-10-LABX-65] which is managed by the french statte fund (ANR) within the Investissements d'Avenir program [ANR-11-IDEX-0004-02]. This work also received financial support from the EU project [644096-ECOMODE].

REFERENCES

[1] T. Hawkes and P. Simonpieri, "Signal coding using asynchronous delta modulation," Communications, IEEE Transcation on, no. 5, pp. 729–731, 1974.

[2] C. Vezyrtzis and Y. Tsividis, "Processing of signals using level-crossing sampling," Proc. IEEE Int. Symp. Circuits Syst, no. 1, pp. 2293–2296, 2009.

[3] C. Posch, D. Matolin, and R. Wohlgenannt, "An asynchronous time-based image sensor," 2008 IEEE International Symposium on Circuits and Systems, pp. 2130–2133, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4541871

[4] J. Kogler, C. Sulzbachner, F. Eibensteiner, and M. Humenberger, "Address-event matching for a silicon retina based stereo vision system," in 4th International Conference from Scientific Computing to Computational Engineering, 2010.

[5] R. Benosman, S. Ieng, P. Rogister, and C. Posch, "Asynchronous event-based hebbian epipolar geometry," Neural Networks, IEEE Transactions on, vol. 22, no. 11, pp. 1723–1734, Nov 2011.

[6] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck, "Asynchronous event-based binocular stereo matching," Neural Networks and Learning Systems, IEEE Transactions on, vol. 23, no. 2, pp. 347–353, Feb 2012.

[7] M. Firouzi and J. Conradt, "Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas," Neural Processing Letters, 2015.

[8] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "Hfirst: A temporal approach to object recognition," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2015.

[9] P. Xi, Z. Bo, Y. Rui, T. Huajin, and Y. Zhang, "Bag of events: An efficient probability-based feature extraction method for aer image sensors," IEEE Transaction on Neural Networks and Learning System, vol. PP, no. 99, pp. 1–13, 2016.

[10] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," Neural Networks and Learning Systems, IEEE Transactions on, vol. 25, no. 2, pp. 407–417, Feb 2014.

[11] B. Rueckauer and T. Delbruck, "Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor," Frontiers in neuroscience, vol. 10, no. 176, 2016.

[12] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of aer dynamic vision sensors," in Proceedigns of the IEEE International Symposium on Circuits and Systems, 2009, pp. 781 –784.

[13] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Regnier, "Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics," Robotics, IEEE Transactions on, vol. 28, no. 5, pp. 1081–1089, Oct 2012.

[14] R. Berner, C. Brandli, M. Yang, and S. C. Liu, "A 240 180 10mW 12us latency sparse-output vision sensor for mobile applications," in VLSI Circuits (VLSIC), 2013 Symposium on, 2013, pp. 186–187.

[15] C. Brandli, R. Berner, M. Yang, S. C. Liu, and T. Delbruck, "A 240x180 130 dB 3 us latency global shutter spatiotemporal vision sensor," IEEE Journal of Solid-State Circuits, vol. 49, no. 10, pp. 2333–2341, 2014.

[16] Z. Ni, C. Pacoret, R. Benosman, S. Ieng, and S. Régnier, "Asynchronous event-based high speed vision for microparticle tracking," Journal of Microscopy, vol. 245, no. 3, pp. 236–244, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1365-2818.2011.03565.x

[17] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," Neural Networks and Learning Systems, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2014.

[18] S.-H. Ieng, C. Posch, and R. Benosman, "Asynchronous neuromorphic event-driven image filtering," Proceedings of the IEEE, vol. 102, no. 10, pp. 1485–1499, Oct 2014.

[19] H. Lorach, O. Marre, J. Sahel, R. Benosman, and S. Picaud, "Neural stimulation for visual rehabilitation: Advances and challenges," Journal of Physiology-Paris, vol. 107, no. 5, pp. 421 – 431, 2013, special issue: Neural Coding and Natural Image Statistics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0928425712000678

[20] H. Lorach, R. Benosman, O. Marre, S. Ieng, J. Sahel, and S. Picaud, "Artificial retina: the multichannel processing of the mammalian retina achieved with a neuromorphic asynchronous light acquisition device," Journal of Neural Engineering, vol. 9, no. 6, p. 066004, 2012. [Online]. Available: http://stacks.iop.org/1741-2552/9/i=6/a=066004

[21] Y. K. Chan and S. Y. Lim, "Synthetic aperture radar (sar) signal generation," Progress In Electromagnetics Research, no. B, pp. 269–290, 2008.

[22] T. Lenart, M. Gustafsson, and V. Owall, "A hardware acceleration platform for digital holographic imaging," Journal of Signal Processing System, vol. 52, no. 3, pp. 297–311, 2008.

[23] M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, vol. 3, Seattle, Washington, 1998, pp. 1381–1384.

[24] M. Püschel, J. M. F. Moura, J. Johnson, D. Padua, M. Veloso, B. Singer, J. Xiong, F. Franchetti, A. Gacic, Y. Voronenko, K. Chen, R. W. Johnson, and N. Rizzolo, "Spiral: Code generation for DSP transforms," Proceedings of the IEEE, special issue on Program Generation, Optimization, and Adaptation, vol. 93, no. 2, pp. 232–275, 2005.

[25] "Intel Math Kernel Library," http://software.intel.com/en-us/articles/intel-mkl/, Jan. 2016.

[26] "Intel integrated performance primitives," http:// software.intel.com/en-us/intel-ipp/, Jan. 2016.

[27] F. Franchetti, M. Püschel, Y. Voronenko, S. Chellappa, and J. M. F. Moura, "Discrete fourier transform on multicore," IEEE Signal Processing Magazine, special issue on Signal Processing on Platforms with Multiple Cores, vol. 26, no. 6, pp. 90–102, 2009.

[28] B. Fang, Y. Deng, and G. Martyna, "Performance of the 3d FFT on the 6D network torus qcdoc parallel supercomputer," Computer Physics Communications, vol. 176, no. 8, pp. 531–538, April 2007.

[29] Z. Qian and M. Margala, "A novel low-power and in-place split-radix fft processor," in Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI, ser. GLSVLSI '14. New York, NY, USA: ACM, 2014, pp. 81–82. [Online]. Available: http://doi.acm.org/10.1145/2591513.2591563

[30] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A 1-gs/s fft/ifft processor for uwb applications," in IEEE Journal of Solid-State Circuits, 2005, pp. 1726–1735.

[31] "Powerfft asic." http://www.eonic.com/index.asp?item=32., Jan. 2016.

[32] B. Baas, "A low-power, high-performance, 1024-point fft processor," in IEEE Journal OF Solid-state Circuits, vol. 34, no. 3, 1999, pp. 380–387.

[33] I. Uzun, A. Amira, and A. Bouridane, "Fpga implementations of fast fourier transforms for real-time signal and image processing." in IEEE Proceedings. Vision, Image, and Signal Processing, vol. 152, no. 3, 2005, pp. 283–296.

[34] P. D'Alberto, P. A. Milder, A. Sandryhaila, F. Franchetti, J. C. Hoe, J. M. Moura, M. Puschel, and J. R. Johnson, "Generating fpga-accelerated dft libraries," Field-Programmable Custom Computing Machines, Annual IEEE Symposium on, vol. 0, pp. 173–184, 2007.

[35] P. Kumhom, "Design, optimization, and implementation of a universal FFT processor," Ph.D. dissertation, Electrical and Computer Engineering, Drexel University, 2001, also Tech. Report DU-MCS-01-01, Drexel University, 2001.

[36] P. A. Milder, F. Franchetti, J. C. Hoe, and M. Püschel, "Formal datapath representation and manipulation for implementing dsp transforms," in Proceedings of the 45th Annual Design Automation Conference, ser. DAC '08. New York, NY, USA: ACM, 2008, pp. 385–390. [Online]. Available: http://doi.acm.org/10.1145/1391469.1391572

[37] T. Dillon, Two virtex-II FPGAs deliver fastest, cheapest, best high-performance image processing system, 2001, vol. 41.

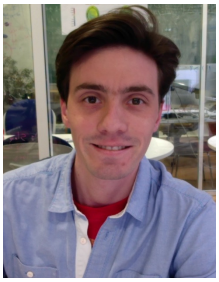
[38] Y. Tsividis, "Event-driven data acquisition and digital signal processing - a tutorial," IEEE transactions on circuits and systems II : Express Briefs, vol. 57, no. 8, 2010.

[39] N. Persson, "Event Based Sampling with Application to Spectral Estimation," Ph.D. dissertation, 2007.

[40] F. Eng and F. Gustafsson, "Frequency transforms based on nonuniform sampling? Basic stochastic properties," Proceedings of Radiovetenskap och kommunikation 2005, pp. 1–4, 2005.

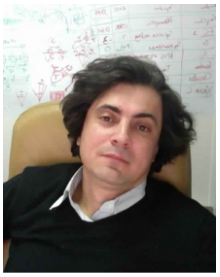
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2017.2661702, IEEE Transactions on Image Processing

[41] F. Eng, "Non-Uniform Sampling in Statistical Signal Processing," Ph.D. dissertation, 2007.
[42] M. Greitans and R. Shavelis, "Extended Fourier series for time-varying filtering and reconstruction from level-crossing samples," European Signal Processing Conference, no. 4, pp. 1–5, 2013.
[43] W. T. Cochran, T. W. Cooley, D. L. Favin, H. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. Welch, "What is the Fast Fourier Transform," audio and electroacoustics, IEEE Transactions on, vol. AU-15, no. 2, pp. 45–55, June 1967.
[44] C. Van Loan, Computational frameworks for the fast Fourier transform. SIAM, 1992.
[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing: a publication of the IEEE Signal Processing Society, vol. 13, no. 4, pp. 600–12, Apr. 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15376593



Quentin Sabatier Quentin Sabatier received the B.Sc. degree and the M.Sc. degree from the Ecole Polytechnique in 2010 and 2012 respectively as well as the M.Sc. degree in neurotechnology from Imperial College in 2013. He is currently pursuing a the Ph.D. degree in neuroscience and neurally inspired machine vision with the Vision Institute, Paris.

SioHoi Ieng Sio-Hoi Ieng received the Ph.D. degree in computer vision from the University Pierre and Marie Curie, Paris, France, in 2005. He is an Associate Professor at the University Pierre and Marie Curie and a member of the Vision Institute, Paris, France. He worked on the geometric modeling of noncentral catadioptric vision sensors and their link to the caustic surface. His current research interests include computer vision, with special reference to the understanding of general vision sensors, cameras networks, neuromorphic event-driven vision and event-based signal processing.



Ryad Benosman Ryad Benosman is a full Professor with University Pierre and Marie Curie, Paris, France, leading the Natural Computation and Neuromorphic Vision Laboratory, Vision Institute, Paris. He received the M.Sc. and Ph.D. degrees in applied mathematics and robotics from University Pierre and Marie Curie in 1994 and 1999, respectively. His work covers neuromorphic visual computation and sensing and event based computation. He is currently involved in the French retina prosthetics project and in the development of retina implants and cofounder of Pixium Vision a french prosthetics company. He also actively works on retina stimulation using optogenetics with Gensight Biologics. He is also a cofounder of Chronocam a company developing Event based cameras and event driven computation systems. He is an expert in complex perception systems, which embraces the conception, design, and use of different vision sensors covering omnidirectional 360 degree wide-field of view cameras, variant scale sensors, and non-central sensors. He is among the pioneers of the domain of omni-directional vision and unusual cameras and still active in this domain. He has been involved in several national and European robotics projects, mainly in the design of artifcial visual loops and sensors. His current research interests include the understanding of the computation operated along the visual systems areas and establishing a link between computational and biological vision. Ryad Benosman has authored more than 100 scientific publications and holds several patents in the area of vision, robotics, event-based sensing and prosthetics. In 2013 he was awarded with the national best French scientific paper by the Journal La Recherche for his work on neuromorphic retinas and their applications to retina stimulation and prosthetics.