THE COMPUTATIONAL COMPLEXITY OF TIME-FREQUENCY DISTRIBUTIONS

Mohan Vishwanath *Robert M. Owens* *Mary Jane Irwin*

Department of Computer Science
The Pennsylvania State University
University Park, PA 16802

ABSTRACT

A number of lower bounds on the communication and multiplicative complexity of time-frequency distributions are derived. The (Area)×(Time)^2 (AT^2) bound for the Discrete Short Time Fourier Transform, the Discrete Wigner-Ville Distribution, the Discrete Ambiguity Function and the Discrete Gabor Transform is shown to be $AT^2 = \Omega(N^3 \log^2 N)$, where N^2 is the number of output points. The lower bound on multiplicative complexity for these is shown to be $\Omega(N^2)$. For the N-point Discrete Wavelet Transform we derive a lower bound of $AT^2 = \Omega(N^2 \log^2 N)$ and a multiplicative complexity of $\Omega(N)$, these are same as the lower bounds for the DFT.

1. INTRODUCTION

For many years now, researchers have been trying to develop efficient architectures and algorithms for computing time-frequency distributions (TFDs). A tight lower bound is a very good metric against which one could compare one's design. In this paper we derive lower bounds on both the communication complexity (VLSI) and the multiplicative complexity of time-frequency distributions. The former complexity is considered because with current VLSI technology it is possible to build single chip implementations of the TFDs (given their huge data rate requirements). The study of VLSI circuits is one of the areas where asymptotic analyses have proven to be useful and meaningful. This can be attributed to the fact that VLSI circuits are enormously complex, thus even an asymptotic approximation can give clear insights into the design(area) and the performance(speed)[1]. A lower bound on the communication complexity of the Discrete Short Time Fourier Transform (DSTFT)[2] is derived and is presented in terms of the area and time required. It is shown to be $AT^2 = \Omega(N^3 \log^2 N)$ for computing a DSTFT with N^2 outputs (Here $f(k) = \Omega(g(k))$ means that there exist positive constants c and k_0 such that $0 \leq cg(k) \geq f(j)$ for $k \geq k_0$). We also show that the discrete versions of the Wigner-Ville Distribution[3], Ambiguity Function[4] and the Gabor Transform[5] have the same complexity. For the N-point Discrete Wavelet Transform (DWT)[6] we derive a lower bound of $AT^2 = \Omega(N^2 \log^2 N)$, which is exactly the same as that of the N-point DFT[7].

We also derive the traditional sequential complexity (we consider multiplicative complexity) of computing the TFDs. This does not have much relevance in many modern general purpose computers wherein the clock period is adjusted to permit single cycle MACs (Multiply-Accumulate) instructions. But since most of the signal processing hardware is special purpose, a multiplication count is useful because, in VLSI, multiplications are still the dominating (area and time) arithmetic operation. A lower bound on the multiplicative complexity for computing the DSTFT is shown to be $\mu = \Omega(N^2)$. This is also shown to hold for the discrete versions of the Wigner-Ville Distribution, Ambiguity Function and the Gabor Transform. For the N-point DWT we show that $\mu = \Omega(N)$ (this is same as the lower bound for an N-point DFT[8][9]).

First, we present a brief description of the computational model under which we derive our bounds, section 3 contains the AT^2 bounds while section 4 contains the multiplicative complexity bounds.

2. THE COMPUTATIONAL MODEL

We use the model proposed by Thompson in [7]. A circuit is made up of gates and I/O ports (pads) which are all connected together by wires or interconnections. It takes each bit at least time τ to be transmitted across a wire; this is taken to be independent of the wire length. This is valid because it can be easily shown that the drivers take up at most 10% of the wires length and can thus be absorbed into the wire length[1]. This assumption does not affect the lower bound result at all, except to make it tighter in technologies where the assumption does not hold. A constant number of layers is considered and hence does not affect the lower bound proof. The chip has width w and length h. The minimum feature size is λ and each gate occupies a constant area $a_g \geq \lambda^2$. The circuit is digital and all the inputs and outputs are equally likely. We are interested in the communication complexity, hence we are concerned with on chip complexity, ie., we want to find the complexity given that the input data is present on chip.

3. AT^2 LOWER BOUNDS

The DSTFT is given by[2],

$$y(n,m) = \sum_{i=0}^{N_w - 1} h(i)x(m-i)W_N^{in} \tag{1}$$

where $h(n)$ is the analyzing wavelet or the window function and $x(n)$ is the input sequence and $W_N = e^{\frac{-j2\pi}{N}}$. Let the size of the support of the window be N_w and let $N(n = 0, 1, \dots N-1)$ be the number of frequencies we want to analyze. Let the number of time shifts be $M(m = 0, 1, \dots M-1)$ and the size of the input sequence be S. Thus at each point in time the DSTFT can be viewed as taking the DFT of the sequence x around that point. It basically consists of multiplying the input sequence by the translates and the modulates of window (or vice-versa). (1) can be rewritten in matrix form as follows,

$$\mathbf{Y} = \mathbf{WHX} \tag{2}$$

where $\mathbf{Y}$ is the $N \times M$ output matrix, $\mathbf{W}$ is the $N \times N_w$ roots of unity matrix , $\mathbf{H}$ is a $N_w \times N_w$ diagonal matrix which contains the window sequence and $\mathbf{X}$ contains the input sequence and has dimensions of $N_w \times M$. Usually $MN_w \geq S$. As far as the lower bounds are concerned, multiplication by $\mathbf{H}$ can be ignored. Thus the problem we are considering now is $\mathbf{Y} = \mathbf{WX'}$.

Clearly, at the highest level this problem has no communication complexity, since one can always ensure that the input is partitioned along the columns. Thus, we will have to look at a lower level to come up with a good bound. To do this we have to partition the input into more than two parts and each one of those parts should not contain more than $N_w/2$ terms (for now assume that $N \geq N_w$). We show that we can always partition the input in a fair manner by showing how we can divide the chip.

The chip is a rectangle of width w and length h and we partition this rectangle. We define the perimeter of a (sub)rectangle as the sum of the lengths of its sides excluding the sides it shares with its parent rectangle. Let the area of the chip be A, then it can be shown easily [10] that this rectangle of area A can be divided into say, d, rectangles each of perimeter $P \leq 6\sqrt{A/d}$. It is also easily shown [10]that if we are given a rectangular region with q inputs, then we can divide it into many regions each containing no more than p inputs by making at most $\frac{cq}{(c-1)p}$ cuts. c is a constant and at most p/c inputs can be present at a point. If there was one region to begin with, then there are at most $1 + \frac{cq}{(c-1)p}$ regions after partitioning it to satisfy the inputs per partition requirement.

As we explained earlier, since $\mathbf{W}$ is a constant matrix, the input ie.. matrix $\mathbf{X'}$ is inherently partitionable at the highest level ie., along columns. So we try to zoom in till we find more difficult structures, in this case, each column of $\mathbf{X'}$ is as far as we need to look.

Theorem 1 *For the DSTFT,* $AT^2 = \Omega(MN_w^2 \log^2 N_w)$ *if* $N \geq N_w$ *and* $AT^2 = \Omega(MN^2 \log^2 N)$ *if* $N_w \geq N$

Proof:(sketch) We want to partition the chip such that each part has at most $N_w/2$ inputs, thus we have to partition it into kM parts where, $k \geq 2$, is a constant. We do not place any restrictions on the position of the outputs. We divide the chip into $3M$ parts, applying the above restrictions , as follows: partition the chip (rectangle) into $M/3$ equal rectangles (we assume WLOG that M is divisible by 3). As mentioned earlier, each of these rectangles has a perimeter of at most $6\sqrt{3}\sqrt{A/M}$. We want to restrict each partition to have at most $N_w/2$ inputs, thus we may have to partition them further. We have $p = N_w/2$, $q = MN_w$ and $c = 4$ (see previous paragraph). The choice of c is arbitrary and to get $3M$ partitions we can use any $c \geq 4$. Thus from above we have that we can always partition the chip further by making at most $\frac{8M}{3}$ more cuts. This gives us a total of $\frac{M}{3} + \frac{8M}{3} = 3M$ parts, with each part containing at most $N_w/2$ inputs.

At this point the prover has shown the capability to partition the chip in a fair manner. Now the designer needs to distribute the inputs such that minimum information flow is required between partitions. This best or minimal distribution of the input obviously consists of pairs of partitions, each pair equally sharing one column of the input and its corresponding row of the output. Consider any partition k and its corresponding pair k'. The computation to be done is multiplying the vector, of length N_w, (a column of the input matrix, due to designer's distribution) contained in this pair, by a roots of unity matrix and store the results back in this pair. In this partitioned form the computation can be written as,

$$\mathbf{Y_k} = \mathbf{W_{11}x_k} + \mathbf{W_{12}x_{k'}} \tag{3}$$
$$\mathbf{Y_{k'}} = \mathbf{W_{21}x_k} + \mathbf{W_{22}x_{k'}} \tag{4}$$

The information flow between the partitions k and k' takes place due to the matrices $\mathbf{W_{12}}$ and $\mathbf{W_{21}}$. Since the roots of unity matrix is a Vandermonde matrix (over the field of complex numbers) the sum of the ranks of $\mathbf{W_{12}}$ and $\mathbf{W_{21}}$ is atleast $N/2$ [7]. Let $N_w \geq N$ and the precision of the the computations be $\log N$. Since the sum of the ranks is $\geq N/2$, therefore the number of distinct combinations of the two cross terms is $\geq N^{N/2}$. Thus, the information flow is atleast $\log(N^{N/2})$, ie., $\frac{N}{2}\log N$. Let $N \geq N_w$ and the precision be $\log N_w$. Since each of the partition contains $N_w/2(\leq N/2)$ inputs, the information flow is atleast $\log(N_w^{N_w/2})$, ie., $\frac{N_w}{2}\log N_w$. The total information flow from a partition during the course of the computation (over time T) is given by (perimeter of partition)×T. Therefore we have $6\sqrt{3}\sqrt{A/MT} \geq \frac{N}{2}\log N$ or $6\sqrt{3}\sqrt{A/MT} \geq \frac{N_w}{2}\log N_w$. In other words, $AT^2 = \Omega(MN_w^2 \log^2 N_w)$ if $N \geq N_w$ and $AT^2 = \Omega(MN^2 \log^2 N)$ if $N_w \geq N$. □

The discrete (both time and frequency) versions of the Wigner-Ville distribution (DWVD), the Ambiguity function and the Gabor transform also have the same lower bound, since they all can be essentially written in the form of multiplying a roots of unity matrix with a input matrix. Note that we are considering transforms with N^2 (or MN) outputs, also the number of outputs is atleast as large as the size of the input (though the inputs and the outputs are quasi-infinite sequences, in real life only finite sequences are handled at any point in time and the "quasi-infiniteness" is managed by using some form of pipelining). We show briefly how the DWVD can be written exactly in the same form as (2). Consider the discrete-time, discrete-frequency Wigner-Ville distribution as defined by Peyrin and Prost

[3],

$$y(n,m) = \frac{1}{2N} \sum_{k=0}^{N-1} x(k)x^*(m-k)W_N^{(km - \frac{ma}{2})} \quad (5)$$

N is the number of time and frequency samples which are of interest, $0 \leq n, m \leq (N-1)$, $x(k)$ is the periodic input signal and $*$ denotes complex conjugation. This can be written in a matrix form as,

$$\mathbf{Y} = \mathbf{W}' \circledast \mathbf{WDX} \quad (6)$$

where $\mathbf{Y}$ is the $N \times N$ output matrix, $\mathbf{W}'$ is an $N \times N$ matrix with $W_N^{-\frac{ma}{2}}$ as its entries, $\circledast$ represents elementwise multiplication, $\mathbf{W}$ is the $N \times N$ roots of unity matrix, $\mathbf{D}$ is an $N \times N$ diagonal matrix containing the sequence $x(k)$ and $\mathbf{X}$ is an $N \times N$ matrix containing $x^*(m-k)$. For a lower bound the multiplication by the diagonal matrix and the elementwise multiplication can be ignored. Hence we have exactly the same form as for the DSTFT. The same result can be obtained even if we use Boashash's form of the DWVD, ie., using an analytic equivalent of the input signal and also windowing it[11]. Similarly, we can show that the discrete Ambiguity function and the Gabor transform are also of this form.

Corollary 1 $AT^2 = \Omega(N^3 \log^2 N)$ *for the discrete versions of the Wigner-Ville distribution, the Ambiguity function and the Gabor transform.*

Proof:
Since the above mentioned transforms/distributions have the form of a roots of unity matrix multiplied by an input matrix, we have the result by way of the proof of Theorem 1. The interplay between the number of time samples, the number of frequency samples and the window size is not visible here, but, wherever it applies, it remains same as in Theorem 1. □

The DWT can be looked at as the multiresolution decomposition of a sequence[6]. And it takes a length N sequence $x(n)$ and generates a length N sequence as the output. The output can be viewed as the multiresolution representation of $x(n)$, and has $N/2$ values at the highest resolution and $N/4$ values at the next resolution and so on. It essentially consists of multiplying the input sequence by the translates and dilates of the wavelet (though this will not be obvious from the equation shown below). Let $N = 2^P$ and let the number of frequencies or resolutions, be P, ie., we are considering octaves. Therefore the frequency index, j, varies as, $1, 2, \cdots, P$ corresponding to the scales $2^1, 2^2, \cdots, 2^P$. The DWT is given by,

$$\overline{W}(n,j) = \sum_{m=0}^{2n} \overline{W}(m, j-1)\overline{w}(2n-m) \quad (7)$$

$$W(n,j) = \sum_{m=0}^{2n} \overline{W}(m, j-1)w(2n-m) \quad (8)$$

where $\overline{W}(n,0) = x(n)$ and $\overline{w}(n)$ and $w(n)$ are Quadrature Mirror Filters derived from the wavelet. This can be written in a matrix form as,

$$\mathbf{y} = \mathbf{Mx} \quad (9)$$

where $\mathbf{y}$ and $\mathbf{x}$ are the N-point output and input vectors, respectively. And $\mathbf{M}$ is the $N \times N$ filter matrix. $\mathbf{M}$ is a block matrix with P blocks, the first block (topmost) is $\frac{N}{2} \times N$ and corresponds to the $N/2$ highest resolution outputs. The second block is $\frac{N}{4} \times N$ and corresponds to the next $N/4$ outputs and so on. $\mathbf{M}$ is, obviously, derived from the QMFs $\overline{w}(n)$ and $w(n)$.

Theorem 2 *For the DWT,* $AT^2 = \Omega(N^2 \log^2 N)$.

Proof: There are two possibilities, either the filter matrix is part of the problem definition (ie., it is a constant) or it is part of the input (ie., any wavelet can be used). For the latter case the problem degenerates to one of computing a matrix-vector multiplication. The bound for this is known and is given by $AT^2 = \Omega(N^2 \log^2 N)$[12]. Consider the case when the filter matrix is constant. If we assume that the same wavelet has been used to derive all the blocks of $\mathbf{M}$ then the density of non-zero elements in the bottom half of this matrix is atleast as much as that of the top half. This is because the number of vanishing moments of the wavelet reflects itself in the number of non-zero elements in each row of $\mathbf{M}$. Thus, to get an asymptotic bound we can discard the last $N/2$ outputs and consider only the first half of the outputs. But, this is equivalent to a FIR filtering operation with $N/2$ outputs, the lower bound for which is known to be $AT^2 = \Omega(N^2 \log^2 N)$[12]. □

4. THE MULTIPLICATIVE COMPLEXITY

Consider (2), keeping in mind that we are interested in an asymptotic lower bound on multiplicative complexity, we can rewrite it as,

$$\mathbf{Y} = \mathbf{WX}' \quad (10)$$

$$\mathbf{y} = \mathbf{Zx} \quad (11)$$

$\mathbf{y}$ is the MN-point output vector, $\mathbf{x}$ is the MN_w-point input vector and $\mathbf{Z}$ is a block diagonal $MN \times MN_w$ matrix, each of the blocks being $\mathbf{W}$. For the sake of completeness, we state a theorem here which is proved in [8][9] . Its called the row rank theorem.

Theorem 3 *Let $z = \Phi y$ be a semilinear system and r be the row rank of Φ over G, then the minimum number of multiplications needed to compute z over B, μ, is bounded from below as $\mu \geq r$.*

G is the ground set or field of constants and multiplication by an element of G is not counted towards the multiplicative complexity. B is the base set and includes all the elements of G plus a set of indeterminate values that are not in G.

Theorem 4 *The multiplicative complexity of the DSTFT is given by* $\mu = \Omega(MN)$.

Proof: Let us take our ground set to be the field of rational numbers, Q , ie., we are not interested in scalar multiplications. Let the base set be the field of complex numbers, C. Consider (11), the matrix $\mathbf{Z}$ is made up of column disjoint blocks (ie., non-overlapping) of **W**. As mentioned earlier the matrix **W** has a row rank of N over C. Since there are M non-overlapping blocks in **Z**, it has a row rank of MN over C. Since Q is a subfield of C, the row rank of **Z** over Q is atleast MN. Thus by the row rank theorem, $\mu = \Omega(MN)$. □

As mentioned earlier, the DWVD, the Ambiguity function and the Gabor transform have the same form as the DSTFT, namely, multiplying a roots of unity matrix by an input matrix. Thus, the result proved above holds for them too.

Corollary 2 $\mu = \Omega(N^2)$, *for the discrete versions of the Wigner-Ville distribution, the Ambiguity function and the Gabor transform.*

Theorem 5 *The multiplicative complexity of the DWT is given by* $\mu = \Omega(N)$.

Proof: Consider the DWT defined in (11), it has N inputs and outputs. As before, there are two cases. If the filter matrix is considered as part of the input ie., indeterminate, then it belongs to the base set, B. This implies that its row rank over G is N. Thus by the row rank theorem we have the lower bound as $\mu = \Omega(N)$.

The second case is when the filter matrix is constant, ie., it belongs to the ground set, G. In this case the proof is as follows: From (8), (9) and (11) we see that the DWT is equivalent to computing a series of $P = \log N$ convolutions. At each succeeding stage the outputs get decimated by a factor of 2. Let the number of operations required to compute the first convolution of $N/2$ points be t, we know that $t = \Omega(N)$[8]. Since at each step the size of the convolution is halved we see that the total number of operations is given by,

$$\mu \propto t + \frac{t}{2} + \frac{t}{4} + \frac{t}{8} + \cdots + \frac{t}{N/2} = 2t(1 - \frac{1}{N})$$

Therefore, $\mu = \Omega(N)$. □

This matches the result shown in [13] where the number of operations per output sample, of the DWT, is shown to be of the same order as that of the first stage of the DWT. This can be observed by taking the filter bank (ie.,multirate) interpretation of the DWT. The result for filter banks has been proved in [14]. Also note that this result is not restricted to non-scalar multiplications, unlike the result of Theorem 4.

5. CONCLUSIONS

We have shown that for the Discrete Short Time Fourier Transform.

$$AT^2 = \Omega(MN^2 \log^2 N) \quad \text{if } N_w \geq N \quad (12)$$
$$\text{and } AT^2 = \Omega(MN_w^2 \log^2 N_w) \quad \text{if } N \geq N_w$$

$$\text{Multiplicative Complexity (non-scalar)} = \mu = \Omega(MN) \quad (13)$$

Discrete versions of the Wigner-Ville Distribution, the Ambiguity function and the Gabor Transform are shown to have similar (as above) lower bounds. We also showed that for the Discrete Wavelet Transform,

$$AT^2 = \Omega(N^2 \log^2(N)) \quad (14)$$

$$\text{Multiplicative Complexity} = \mu = \Omega(N) \quad (15)$$

Note that the N in (12) and (13) refers to the number of frequencies we want to consider, while in (14) and (15) it refers to the length of the input (output) sequence.

These bounds are tight, in the sense that there exist circuits and algorithms which achieve these bounds.

REFERENCES

[1] Thompson C.D., "Fourier Transforms in VLSI," *IEEE Trans. on Comp.*, vol c-32, No.11, pp. 1047-1057, Nov. 1983.

[2] Lim J.S. and Oppenheim A.V., *Advanced Topics in Signal Processing*, New Jersey, USA: Prentice Hall, 1988.

[3] Peyrin F., Prost R., "A Unified Definition for the Discrete-Time, Discrete-Frequency and Discrete-Time/Frequency Wigner Distributions," *IEEE Trans. ASSP*, Vol. ASSP-34, No.4, pp. 858-867, Aug. 1986.

[4] Tolimieri R., Winograd S., "Computing the Ambiguity Surface," *IEEE Trans. ASSP*, Vol. ASSP-33, No.4, pp. 1239-1245, Oct. 1985.

[5] Dellomo M.R. and Jacyna G.M., "Wigner transforms, Gabor coefficients and Weyl-Heisenberg wavelets," *J. Acoust. Soc. Am.*, vol.89, No.5, pp. 2355-2361, May 1991.

[6] Mallat S., "Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Trans. Acoustics Speech and Sig. Proc.*, vol 37, No.12, pp. 2091-2110, Dec. 1989.

[7] Thompson C.D., "Area-Time Complexity for VLSI," *Proc. 11th Annual ACM STOC*, ACM, pp. 81-88, May 1979.

[8] Winograd S., *Arithmetic Complexity of Computations*, CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 33, 1980.

[9] Heideman M.T., Multiplicative Complexity, Convolution, and the DFT, New York, USA: Springer-Verlag Inc., 1988.

[10] Ullman J.D., *Computational Aspects of VLSI*, MD, USA, Computer Science Press, 1984.

[11] Boashash B., Black P.J., "An Efficient Real-Time Implementation of the Wigner-Ville Distribution," *IEEE Trans. ASSP*, Vol. ASSP-35, No. 11, pp.1611-1618, Nov. 1987.

[12] Vuillemin J., "A combinatorial limit to the computing power of VLSI circuits," Proc. 21st Symp. FOCS, IEEE Comp. Soc., Oct. 1980, pp.294-300.

[13] Vetterli M., Herley C., "Wavelets and Filter Banks: Relationships and New Results," *Proc. ICASSP*, vol.3, pp.1723-1726, 1990.

[14] Ramstad T.A., Saramäki T., "Efficient Multirate Realization for Narrow Transition-Band FIR Filters," *Proc. IEEE Intl. Symp. on Ckts. and Systems*, pp. 2019-2022, 1988.