The Runtime Benchmarking of DCT-II based on Cyclic Convolutions

Ihor Prots’ko and Roman Rykmas
Lviv National Polytechnic University, Lviv, Ukraine
Corresponding author. Tel.:+38-097-968-2778; email: protsko@polynet.lviv.ua

ABSTRACT

DCT_CC program for the automatic generation of algorithms and their computation of DCT-II of type II based on cyclic convolutions are considered. The efficient computation of DCT-II has been performed for the methodological approach which is based on hashing arrays. The subtasks of automatic code generation for computing of DCT-II of arbitrary size N have been determined. The comparison and evaluation of developed DCT_CC program with the program of FFTW library have been performed. As a result of benchmarking, DCT_CC program executes of DCT-II faster than the program of FFTW library for the short sizes. The algorithms of DCT_CC program of short sizes are important for designing the algorithms of the large sizes. The program for the automatic generation of algorithms can be extended to create software systems for other discrete transforms of Fourier class.

Keywords: Discrete cosine transform; Automatic generation; Fast algorithm; FFTW library; Cyclic convolution.

1. Introduction

Modern development of software and hardware technology creates highly effective means of digital signal processing. Among these means the fast algorithm of discrete transform of Fourier Class (FC) are of particular attention. The current general level of the analysis and development of fast algorithms of FC is based on algebra [1], which comprises various versions of the efficient algorithms. This allows developing the software for generation of the efficient algorithms of FC in modern software libraries FFTW [2], SPIRAL [3], CuFFT [4], MKL [5], IPP [6], Nukada FFT [7], which provide a wide choice of fast algorithms to different hardware computing platforms.

The library of FFTW (Fastest Fourier Transform in the West) is freely available [2] and tunes the computation of fast transforms of Fourier class automatically for any particular hardware. With the advent in 1997, FFTW library is improved in the form of new versions (currently available FFTW 3.3.6) in terms of the transform sizes, a set of types of discrete transforms, the parallelization of computing, the including in the library the specific algorithms with new effective solutions. FFTW library uses the self-optimizing approach of the selection among the various developed fast algorithms of discrete transforms of FC. The capabilities FFTW library of adapting to different computer architectures is the result of its runtime empirical search.

The comparison and analysis of the developed software implementation DCT_CC of the automatic generation of the algorithms of DCT-II based on cyclic convolutions (CC) and of FFTW library are considered. This article is organized as follows: Section 2 introduces the level of algorithm and its implementation of the computation of DCT-II based on cyclic convolutions, Section 3 describes the benchmark of DCT_CC based on cyclic convolutions and FFTW for computation of DCT-II. Section 4 presents the analysis and discussion of the average results of performance of DCT_CC and FFTW_DCT programs. Finally, Section 5 underlines the importance of further development of the generation algorithms and their execution of DCT-II based on cyclic convolutions.

2. The generation of algorithms of DCT-II

In the basis of programs of the generation of the algorithms of discrete transform of FC lays the use of advanced approaches for the development of effective algorithms [8]. The modern systems of the code generation perform their selection using evolutionary methods, which take into account the multifactor characteristics [9], FFTW library for the most optimal choice with the collection of algorithms uses the dynamic programming [10].

In modern systems of the generation of the efficient algorithms of discrete transforms of FC excrete the levels of algorithm, implementation and evaluation [11]. Let us consider these levels in the case for the generation and implementation of algorithms of DCT-II in the program system DCT_CC and FFTW library.

2.1. The level of algorithm of the computation of DCT-II

Along with DFT there exist a number of other real transforms, better known as discrete cosine (DCT) and sine transforms (DST) of types I-VIII. DCT-II, which is often called as DCT, has wide practical use and especially for image processing [12].

DCT-II for size \( N \) is defined in the matrix form:

\[
X = W x
\]  
(1)
where \( W \) – a basis square matrix with the real elements \( w(k,n) = c(n) \cos\left((2k+1)\pi/2N\right) \) for \( n, k = 0, 1, \ldots, (N-1) \) and \( c(0) = (1/N)^{1/2} \); \( c(n) = (2/N)^{1/2} \); \( x(N) \) and \( X(N) \) – columns of input and output data.

The designing and implementing of the efficient algorithms of DCT-II based on CC was described in the paper [13]. The technique using CC for efficient computation of discrete transforms of FC is based on the reformation of discrete basis matrix into the matrix with cyclic submatrices. Respectively for the methodological approach [13], the matrix with cyclic structure of the basis matrix is determined by hashing array \( P(m) \):

\[
P(m) = P_1(m_1) P_2(m_2), \ldots, P_t(m_t)
\]

where \( m \) – a size of array; \( t \) – a number of subarrays \( P_i(m) \); \( m_i \) – a size of subarray, \( i=1,2,\ldots,t \).

The number of hashing arrays \( P(m) \) is more than one and determined by the value \( N \) of the size of DCT-II. The hashing array \( P(m) \) can be defined by the simplified hashing array \( P'(m) \) with the addition of a respective array of cosine signs \( Sc(m) \):

\[
P'(m) = P'_1(m_1) P'_2(m_2), \ldots, P'_t(m_t)
\]

\[
Sc(m) = S_1(m_1) S_2(m_2), \ldots, S_t(m_t)
\]

The array of cosine signs \( Sc(m) \) consists the elements what equal to the values \( +1, -1, 0 \). Then using the simplified hashing array \( P'(m) \) and the array of signs \( Sc(m) \), the structure of the discrete basis matrix \( W \) as a set of cyclic submatrices with simplified elements is reformed.

The specific of the technique [13] for DCT-II is in the use of two hashing arrays of the rows \( Pr(m) \), \( Pr'(m) \), \( Sr(m) \) and columns \( Pc(m) \), \( Pc'(m) \), \( Sc(m) \) for the reformation of the structure of basis matrix \( W \) into the cyclic submatrices cyclic. The structure of basis matrix \( W \) might contains identical cyclic submatrices for the condition of the equality of the first simplified elements of submatrices with signs or opposite signs (quasi identical). Due to the presence of horizontally / vertically identical cyclic submatrices in the structure of the matrix \( W \), the number of the computations of CC decreases. The combination of the results of CC is performed in accordance with the specific of the structure with cyclic submatrices of the basis matrix \( W \) for computation of output data of DCT-II.

FFTW library applies two kinds of plans of DCT-II: the direct plan, the general-length plan. The general-length plan performs the re-expression of the transform of length \( N \) in terms of real-input DFT with the pre/post-processing. In most cases, FFTW library use the transition to the computation of the size \( 2N \) of real-input DFT for computation of DCT of the type II/III of arbitrary sizes \( N \). As a result, the algorithm of DCT-II is replaced by a real DFT of larger size, which is simplified taking into account the symmetry of resulting DFT [10]. The direct plan uses a dynamic programming algorithm to prune the search space with purpose to select the best execution time.

2.2. The level of the implementation of the algorithms of DCT-II

The program implementation on the C++ language for automatic generation of efficient algorithms for computation of DCT-II of arbitrary sizes \( N \) using CC corresponds to the developed methodological approach which is based on hashing arrays [13]. The schematic model of the automatic generation of algorithm and computation of DCT-II based on CC for the size \( N \) is divided into subtasks that correspond to nodes of the graph (Figure 1).

The schematic model includes the input \( N \) – a size of transform, \( x(N) \) – input and \( X(N) \) – output data of DCT-II. The unidirectional arcs of relationship between the specific nodes of the subtasks, responsible for serial transmission of relevant data, provide automatic generation of algorithm and computation of DCT-II. The view of schematic model of the relationship of the subtasks reflects the possibility of concurrent organization the generation of algorithm and the computation of DCT-II.

**Figure 1.** The schematic model of the automatic generation of DCT-II algorithm and their computation using cyclic convolutions.

The nodes (1, 2, 3, 4, 5) correspond to the subtasks of the automatic generation of the algorithm for computing of DCT-II for sizes adapted to an arbitrary integer \( N \), which perform:

1 – the determination, if \( N \) is a prime or multiple of 2, multiple of 4, multiple of 8, or \( N = 2^i \), that belongs to one of five subsets of integers;

2 – the decomposition of the value \( N \) into prime factors [3] and definition of hashing array \( P(m) \) using an appropriate row of arguments of the basic matrix of DCT-II;

3 – the simplification of the hashing array \( P(m) \) to \( P'(m) \) based on the property of symmetry of the basis functions;

4 – the complementation of simplified hashing array \( P'(m) \) by the signs array \( Sc(m) \);

5 – the definition of the structure of basis matrix of DCT-II via the first simplified elements with signs of cyclic submatrices using \( Pr(m) \), \( Pc(m) \).

The nodes (6, 7, 8) correspond to the subtasks of determination of the cosine coefficients, which include the computation of:

6 – the value \( \Delta \phi = (\pi/2N) \) of interval between the arguments of DCT-II;

7 – the value of arguments \( (i \Delta \phi) \) of \( \cos(\ ) \) function, \( i = 1,2,\ldots, m/2 \);

8 – the absolute values of the cosine coefficients \( |\cos(i \Delta \phi)| \), \( i = 1,2,\ldots, m/2 \).
The nodes (9, 10, 11,...,11', 12) correspond to the subtasks of computation of the output data of DCT-II, in accordance with the specific of the structure of the basis matrix \( W \), which perform:

9 – the combinations of defined input data \( x(n) \);

10 – the selections and grouping of the combined input data and cosine coefficients for the perform of cyclic convolutions;

11,...,11’ – the execution of fast CC between the grouped data and the corresponding values of cosine coefficients of the basis functions \( Sc(i)\cos(P(i)\pi/2N) \) and combined input data;

12 – the combinations of the results of the CC for computation of the output data \( X(n) \) of DCT-II.

The DCT_CC program, due to the schematic model (Figure 1), automatically generates algorithms and computes of the DCT-II for an arbitrary size \( N \) of transforms [14]. For automatic generation of algorithm of DCT-II the integer arithmetic is used [15]. The automatic generation of the algorithm for computing of DCT-II for sizes \( p \) of integer power of prime excludes the implementation of the subtask 5, because, for each increment of power the structure of basis matrix with cyclic submatrices of DCT-II is expanded regularly.

The existence of several hashing arrays \( P(m) \) for DCT-II of size \( N \) are possible allowing to forming the various structures of basis matrix. According to the structures, the versions of generating algorithms might use the CC of other values of the length. The evaluation and selection of algorithms for computing the DCT-II generated by one of the hashing arrays is produced for the select the maximal speed of execution on a given hardware platform. As an example of the choice via hashing arrays, the best version for the amount of cycles of the algorithm of DCT-II for size \( N = 103 \) is presented in Table 1.

Table 1. The selection of algorithms of DCT-II for the hashing arrays.

<table>
<thead>
<tr>
<th>Versions of ( P(m) )</th>
<th>Amount of cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>row1 - row2</td>
<td></td>
</tr>
<tr>
<td>0 - 28</td>
<td>1156</td>
</tr>
<tr>
<td>0 - 79</td>
<td>1143</td>
</tr>
<tr>
<td>0 - 102</td>
<td>1490</td>
</tr>
</tbody>
</table>

FFTW library automatically tunes the algorithm DCT-II to specific hardware. As a result, the generation the program code on the language C computes the DFT instead of DCT-II of the size \( N \). FFTW does not create the whole algorithm of DFT because it is structured as a library of program fragments – codelets. The sequential part of code C can combine different codelets, where some codelets execute more software tasks. According to the concept of FFTW – the composition of codelets is called a plan [10]. The exact plan that is used by FFTW depends on the size of input data and speed of execution on the basis of hardware platform. In FFTW the most productive-critical code is generated automatically using specialized compiler – genfft, which produces the source code on the language C. The source code of the abstract algorithm DCT-II is expressed as a code for computation of real DFT for a size greater than \( N \)16 with the relevant symmetries and uses an optimized FFT algorithm of FFTW library. For large sizes of transforms, FFTW library uses a chain of optimizations, which are specially installed in FFTW 3.0 release. Generator FFTW is not getting clearly the general formula for counting the number of floating-point operations for algorithms of DCT (excluding of DCT-I).

3. The benchmark of DCT-II based on cyclic convolutions and FFTW

The DCT_CC program of the generation of the algorithms of DCT-II based on CC and their execution via the schematic model (Figure 1) is implemented on the language C++. The software implementation of DCT_CC compares with the program FFTW_DCT, which is generated by FFTW3.3.5 library. In FFTW library for definition of the optimized source code a plan is created, which ensures the generation of algorithm for computing of DCT-II for the given \( n \)Size :

\[
\text{fftw_plan p=fftw_plan_r2r_1d(nSize, in, out, FFTW_REDFT10, FFTW_ESTIMATE);}\
\]

The specialized compiler genfft for the generation of algorithms of FFTW program occupies 5.3 MB memory (#include “fft3.3.5/fftw3.h”). FFTW library includes the best known algorithms of FFT of different approaches and prevails in ten times for the volume in comparison with our developed program DCT_CC.

For the evaluation and comparison of generation and execution of DCT-II algorithms the computing platform Intel(R)Core(TM) i-7CPU2600@4.2GHz is used. The runtime benchmarking uses the library function \text{QueryPerformanceTimer()} of the language C++ for Windows and the library function \text{rdtsc()} for determination of the number of cycles. The sequence \( \{0, 1, 2, ..., N-1\} \) of integer numbers is used for the test data. For obtaining the values of the runtime the computation is performed several times (to avoid fluctuations due to system interrupts, cache priming, etcetera) and reporting the minimum average time.

The comparison of the performance of DCT_CC program and FFTW_DCT for randomly selected of the short sizes is represented in the Table 2 in \( \mu \)s and cycles. We use the number of cycles also, because the resolution of the system timer is not as accurate as the cycles.
Table 2. The average results of performance of DCT_CC and FFTW_DCT programs.

<table>
<thead>
<tr>
<th>Program</th>
<th>N</th>
<th>7</th>
<th>8</th>
<th>10</th>
<th>16</th>
<th>20</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFTW_DCT</td>
<td>µs</td>
<td>0.331292</td>
<td>0.301174</td>
<td>0.391526</td>
<td>0.301174</td>
<td>0.722818</td>
<td>0.963757</td>
</tr>
<tr>
<td></td>
<td>cycles</td>
<td>662</td>
<td>300</td>
<td>1051</td>
<td>813</td>
<td>903</td>
<td>2556</td>
</tr>
<tr>
<td>DCT_CC</td>
<td>µs</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
</tr>
<tr>
<td></td>
<td>cycles</td>
<td>513</td>
<td>677</td>
<td>656</td>
<td>872</td>
<td>895</td>
<td>727</td>
</tr>
</tbody>
</table>

29  30  32  38  45  48  51  64

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.692701</td>
<td>0.993875</td>
<td>0.572231</td>
<td>1.295049</td>
<td>1.114344</td>
<td>1.596223</td>
<td>1.385401</td>
</tr>
<tr>
<td></td>
<td>1966</td>
<td>2915</td>
<td>1595</td>
<td>3984</td>
<td>3252</td>
<td>3981</td>
<td>3866</td>
</tr>
<tr>
<td></td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.301174</td>
<td>0.451761</td>
<td>0.331291</td>
</tr>
<tr>
<td></td>
<td>692</td>
<td>956</td>
<td>1264</td>
<td>847</td>
<td>958</td>
<td>1577</td>
<td>1236</td>
</tr>
</tbody>
</table>

67  83  85  91  96  103  114

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.686575</td>
<td>2.138337</td>
<td>1.837162</td>
<td>1.566106</td>
<td>1.566106</td>
<td>2.891272</td>
<td>2.710567</td>
</tr>
<tr>
<td></td>
<td>5029</td>
<td>6661</td>
<td>5523</td>
<td>4569</td>
<td>4457</td>
<td>9320</td>
<td>7962</td>
</tr>
<tr>
<td></td>
<td>0.301174</td>
<td>0.511996</td>
<td>0.391526</td>
<td>0.391526</td>
<td>0.752935</td>
<td>0.301174</td>
<td>0.632465</td>
</tr>
<tr>
<td></td>
<td>961</td>
<td>1335</td>
<td>1125</td>
<td>1110</td>
<td>2455</td>
<td>1193</td>
<td>2104</td>
</tr>
</tbody>
</table>

As a result, DCT_CC program has the runtime of DCT-II less than FFTW_DCT program for the values of short sizes of transforms (Figure 2).

Figure 2. The average runtimes FFTW_DCT and DCT_CC.

4. The analysis and discussion

Reasoning is important in the analysis of the benchmarking results. FFTW library uses Raiders algorithms which were described in the book [17]. In the FFTW paper [10] the use of Raiders algorithms for computation of DFT, which reduces the computation using the cyclic convolutions, is described. As an example, the improvement of number of operations for prime \( N = 13 \) of DFT the resulting of performance of the complex operations is given in [10]. The minimum value of runtime FFTW_DCT and DCT_CC for size \( N = 13 \) is equal to 835 and 594 cycles according. In FFTW library the computations of CC are performed via convolution theorem on the basis of direct and inverse FFT. The computation of DCT-II is described using the general-length plan, which converts the computation of DCT-II for the size \( N \) into DFT for size \( 2N \) for real input data [18]. Further the well-optimized of FFT algorithm, especially for sizes equal \( 2^n \), is used.

In accordance with the Table 2, the short sizes of DCT-II based on CC are executed faster than those of DFT with the real symmetric (or anti-symmetric) continuation of input data sequence. The ratios of the runtimes (FFTW_DCT / DCT_CC) from the Table 2 are shown in Figure 3. Only in the case of DCT-II for the sizes \( N = 8, 16 \) the ratio of execution time less to 1 is obtained.

Figure 3. The ratios of the runtimes FFTW_DCT/DCT_CC.

Therefore, the use of algorithms for short sizes of DCT_CC program is important for designing the algorithms of the large sizes. A similar conclusion, after receiving the research of automatic generation of Raider-Winograd algorithms, is described in [19].

The comparison of the generation of DCT-II by DCT_CC and FFTW algorithms is not entirely correct, because the programs use different approaches. The \textit{genfft} for performance of the sequential phases of the generation of algorithm need the corresponding time for creation of the directed acyclic graph (dag), the simplification, the transposition, the description of the relationships and the grammatical synthesis of source code.

The generation of DCT-II algorithm based on CC via program DCT_CC carries out the subtasks (Figure 1) using integer arithmetic only. As a result, of the source code of DCT-II contains
the parts, which includes: the union of input data, computation of cyclic convolutions, combining the results of cyclic convolutions. The ratios of the runtimes for the generation of DCT-II algorithm (FFTW_DCT / DCT_CC) are shown in Figure 4.

Figure 4. The ratios FFTW_DCT/DCT_CC of the runtimes for the generation of DCT-II algorithm.

The comparison shows the relevance of using of DCT_CC program in devices, where the least total time of the automatic generation of algorithm plus their execution and simplicity of their realization is important for DCT-II of arbitrary sizes.

5. Conclusions

In this paper the runtime benchmarking of DCT-II_CC program for the automatic generation of algorithms of DCT-II based on cyclic convolutions and their computation are considered. We use the hashing arrays for the automatic generation of algorithms of DCT of type II based on cyclic convolutions. The comparison and evaluation of DCT-II_CC program have been performed with of FFTW library, which is wide used for benchmarking. The results show the improvement of the execution time for short sizes of DCT-II in comparison with the conventional FFTW program. The advantage of DCT-II_CC program is the total time the automatic generation of algorithm plus their execution of DCT for arbitrary sizes.

The experience of the first version of software implementation of DCT_CC program identifies the ways of further improving the performance of the generation algorithms and their execution of DCT-II based on cyclic convolutions.

The schematic model of generation algorithms of DCT-II based on CC can be extended to create compact software systems for others discrete transforms of Fourier class [20].

References