A Dataflow Description of ACC-JPEG Codec

Khaled Jerbi[1,2], Tarek Ouni[1] and Mohamed Abid[1]
[1]*CES Lab, National Engineering School of Sfax, Sfax, Tunisia*
[2]*IETR/INSA, UMR CNRS 6164, F-35043 Rennes, France*

Keywords: Video Compression, Accordeon, Dataflow, MPEG RVC.

Abstract: Video codec standards evolution raises two major problems. The first one is the design complexity which makes very difficult the video coders implementation. The second is the computing capability demanding which requires complex and advanced architectures. To decline the first problem, MPEG normalized the Reconfigurable Video Coding (RVC) standard which allows the reutilization of some generic image processing modules for advanced video coders. However, the second problem still remains unsolved. Actually, technology development becomes incapable to answer the current standards algorithmic increasing complexity. In this paper, we propose an efficient solution for the two problems by applying the RVC methodology and its associated tools on a new video coding model called Accordion based video coding. The main advantage of this video coding model consists in its capacity of providing high compression efficiency with low complexity which is able to resolve the second video coding problem.

1 INTRODUCTION

During most of two decades, MPEG has produced several video coding standards such as MPEG-2, MPEG-4, AVC and SVC. However, the past monolithic specification of such standards (usually in the form of C/C++ programs) lacks flexibility. Such specification does not allow to use the combination of coding algorithms from different standards enabling to achieve specific design or performance trade-offs and thus fill, case by case, the requirements of specific applications. So as to overcome the intrinsic limitations of specifying codecs algorithms by using monolithic imperative code, Caltrop Actor Language (CAL) (Eker and Janneck, 2003) (ISO/IEC FDIS 23001-4: 2009, 2009), has been chosen by the ISO/IEC standardization organization in the new MPEG standard called Reconfigurable Video Coding (RVC). RVC-CAL standard is developped in the ptolemy2 project (Brooks et al., 2004). It is supported with a complete framework called OpenDF (Bhattacharyya et al., 2008) and a recently developed compiler called Open RVC-CAL Compiler (Orcc) (Janneck et al., 2010) allowing users to define a multitude of codecs, by combining together actors (called coding tools in RVC) from the MPEG standard library written in CAL (Eker and Janneck, 2003) (ISO/IEC FDIS 23001-4: 2009, 2009), that contains video technology from all existing MPEG video past standards (i.e. MPEG- 2, MPEG- 4, etc.). CAL is used to provide the reference software for all coding tools of the entire library. The originality of this paper is the application of the CAL and its associated tools on a new video coding model called Accordion based video coding (Ouni et al., 2009) (Ouni et al., 2010) (Ouni et al., 2011). The main advantage of this video coding model consists in its capacity of providing high compression efficiency with low complexity. Such advantage comes from the original idea that consists in applying a particular 3D scan, called Accordion, on temporal frames generated by a temporal video decomposition which is able to transform a set of video frames sequence to a high correlated spatial representation called IACC. The high correlation of the IACC representation, which is actually originated from the video temporal correlation, is easy and efficiently exploited by any still image coder. It was shown in (Ouni et al., 2011) that such coder could produce a high compression ratio (close to Inter compression models such as MPEG) with low computational requirements (close to Intra compression models such as M-JPEG). Additionally, this model offers the possibility of easy reutilizing different mature image compression components for video compression purpose. The ACC-JPEG coder, as an example of Accordion based video coding model, consists in associated the so called Ac-

Jerbi K., Ouni T. and Abid M..
A Dataflow Description of ACC-JPEG Codec.
DOI: 10.5220/0004076901020107
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications and Wireless Information Networks and Systems* (SIGMAP-2012), pages 102-107
ISBN: 978-989-8565-25-9

cordion process to the JPEG standard image coder (Ouni et al., 2009). We try to prove, via this coder, the efficiency of such model, its low computational requirements and the ease of its implementation which become much easier with adopting the RVC framework. Actually, we propose -via this work- an easy and practical solution for fast designing and implementing an efficient video coder.

In section 2, we present the Accordion based coding principle and we give an overview on the RVC-CAL methodology. Section 3 analyses through the proposed CAL based implementation the ACC-JPEG coder performances (coding and decoding speed, memory consumption). Section 4 concludes.

2 BACKGROUND

In this section we review the Accordion based coding approach. Then we details the data flow implementation and the RVC CAL methodology.

2.1 Accordion

The idea behind the Accordion approach relies on the hypothesis saying that the video stream contains more temporal redundancies than spatial ones (Yun Q . Shi, 1999), (Gokturk and Aaron, 2002). In order to take advantages from this hypothesis, the idea consists in trying to put pixels -which have a very high temporal correlation - in spatial adjacency. Thus, video data will be presented with high correlated form which exploits both temporal and spatial redundancies in video signal with appropriate portion that put in priority the temporal redundancy exploitation.

2.1.1 Accordion Representation

The input of our encoder is the so called video cube (GoF), which is made up of a number of frames. This cube will be decomposed into temporal frames which will be gathered into one 2D representation. Temporal frames are formed by gathering the video cube pixels which have the same column index. These frames will be scanned while reversing the direction of odd frames in order to more exploit the spatial correlation of the video cube frames extremities. This representation transforms temporal correlation of the 3D original video source into a high spatial correlation in the 2D representation (”IACC”) (Ouni et al., 2009) (Ouni et al., 2010) (Ouni et al., 2011). Figure 1 illustrates the principle of this representation.

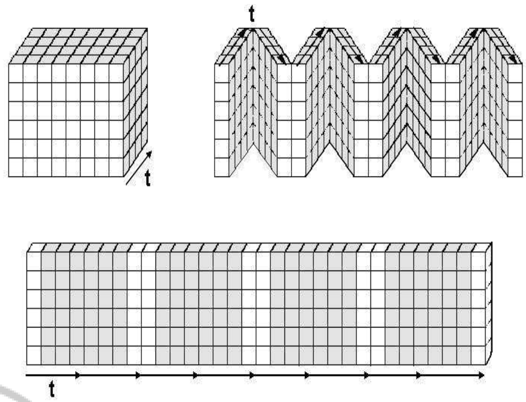

Figure 1: Accordion principle.

2.1.2 Accordion Algorithm

In the following we present the algorithms corresponding to Accordion representation of a sequence of video frames. The input of this algorithm, called ACC, has as input a group of pictures (I_K, for $0 \leq K \leq N-1$) called GOP (Group of Pictures) and as output the resulting IACC image.

```
 Inputs : I0,, IN-1

Outputs : IACC
  For x from 0 to (L x N)-1
    For y from 0 to H-1
      If (x div N mod 2 !=0) Then
        n=(N-1)  (x mod N)
      Else
        n= x mod N
      End
      IACC (x,y)= In (x div N,y)
    End
  End
```

The inverse algorithm, denoted ACC-1, has as input the image IACC and as output the set of images I_K, for $0 \leq K \leq N-1$.

```
 Inputs : IACC
 Outputs : I0,, IN-1

 For n from 0 to N -1
    For x from 0 to L -1
       For y from 0 to H -1
         If (x mod 2=0) Then
           XACC= (N -1)  n +(x*N)
         Else
           XACC= n+ (x*N)
         End
       In(x,y)=IACC (XACC,y)
       End
    End
 End
```

Let us note that :

- L and H are respectively the length and the height of the video source frames.

- N is the number of frames of a GOF.
- IACC(x, y) is the pixel intensity with the coordinates x, y according to accordion representation repair.
- In(x, y) is the intensity of pixel situated in the Nth frame in the original video source.

2.1.3 Video Coding Model

In this part, we present the coding diagram based on the Accordion representation. First, the video encoder takes a video sequence and passes it to a frame buffer in order to construct volumetric images by combining N frames into a stack. Then, the obtained stack will be transformed to form the accordion representation (IACC). Here N is the constructed stack depth (N is 8 in our experiments). Next, each IACC will be divided into N x N blocks to be processed furthermore by the eventual used 2D transform. The encoder block diagram of the Accordion based compression algorithm is presented above.

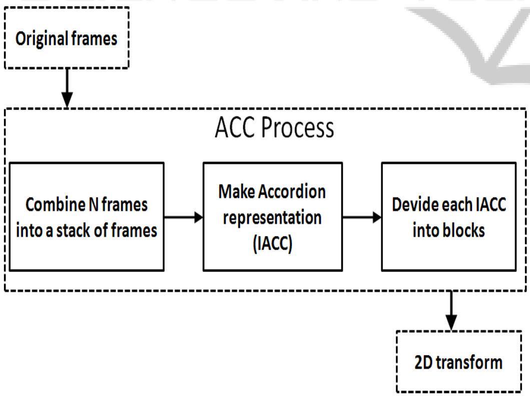

Figure 2: Accordion based video coding model.

This Accordion based video coding model as it is illustrated in figure 2 shows a great flexibility with different possibilities of extensions and reutilization of existing image processing tools leading to designing various versions. In this paper, as it was introduced above, we are interested by the JPEG version known as ACC-JPEG.

2.2 Dataflow Implementation

In the following we present the dataflow programming based on the Caltrop Actor Language and the implementation generation using the back-ends of Open RVC CAL compiler (Janneck et al., 2010).

2.2.1 CAL Programming

The execution of an RVC-CAL code is based on the exchange of data tokens between computational entities called *actors*. Each actor is independent from the others since it has its own parameters and finite state machine if needed. Actors are connected to form an application or a design, this connection is insured by FIFO channels. This connection is modeled using XML based dialects as Xml Dataflow Format (XDF). These languages also provide the possibility to include parameters when instantiating an actor. Consequently, the same actor may be instantiated several times with different parameters. We currently use the *Graphiti*[1] tool to manage XDF and NL graphs.

Executing an actor is based on firing elementary functions called *actions*. This action firing may change the state of the actor. An action may be included in a finite state machine or untagged. An untagged action is higher priority than FSM actions. An RVC-CAL dataflow model is shown in the network of figure 3.

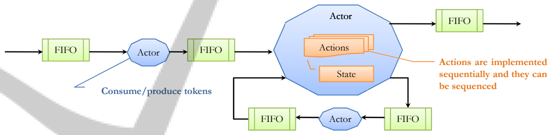

Figure 3: CAL actor model.

When an action is fired, it consumes token streams from input ports and produces token streams to output ports.

We consider a simple *clip* actor that clips the consumed tokens values greater than 255 to 255 and those less than 0 to 0. This algorithm is used in the IDCT2D process of MPEG4 decoders and it is applied on the image 8x8 macro blocks. The associated RVC-CAL code is shown in figure 4 and thus the input streams are:

INPUT1: $INPUT_0$, $INPUT_1$ ... $INPUT_{63}$

the output stream is:

OUTPUT: $OUTPUT_1$, $OUTPUT_2$ ... $OUTPUT_{63}$

such as, for an integer k, $OUTPUT_k$ = clip($INPUT_k$)

In this case, the firing rule is the presence of 64 tokens at least in the input FIFO and the availability of 64 memory cells at least in the output FIFO.

2.2.2 CAL Compiling

Orcc is a CAL compiler that takes in the front-end a set of actors and a graph (see example of graphiti gen-

[1] http://sourceforge.net/projects/graphiti-editor

```
actor clipActor ()
(int size=8) INPUT1 ==> int(size=8) OUTPUT:

 clip: action INPUT1:[ x ] repeat 64 ==>
 OUTPUT:[[
  if x[i] > 255 then
   255
  else
   if x[i] < 0 then
    0
   else
    x[i]
   end
  end : for int i in 0 .. 63 ] ] repeat 64
 end
end
```

Figure 4: RVC-CAL example algorithm of clip actor.

erated graph in Figure 5) that specifies the connexion between theses actors. After some modifications in the middle-end, Orcc applies a string template on the intermediate representation (IR) to generate a chosen back-end this principle is detailed in Figure 6.

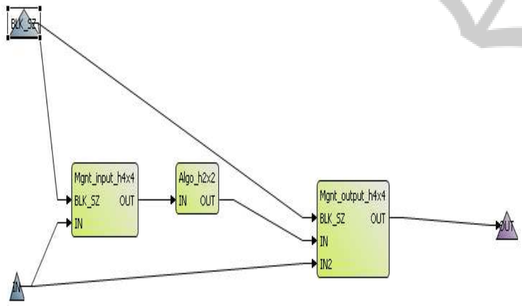

Figure 5: XDF Graph example.

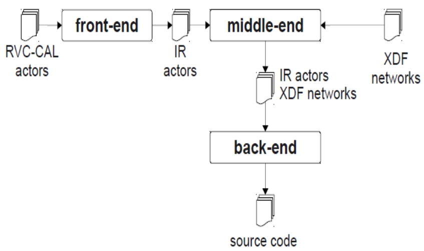

Figure 6: Orcc compilation principle.

Several existing language generators more or less mature are developped and maintained. We cite: C, C++, Xlim, Verilog, VHDL, Promela, Java, LLVM etc. In this paper we use the C back-end because it is the most efficient and also because a C implementation is the easiest to debug and validate.

3 IMPLEMENTATION AND RESULTS

To implement the ACC-JPEG coder using RVC methodology, we start from an RVC-CAL design of the JPEG codec (see http://orc-apps.sourceforge.net/). The design is shown in Figure 7.

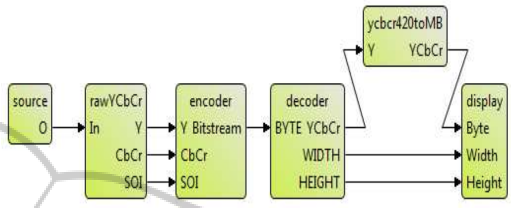

Figure 7: JPEG reference design.

We notice that the design of Figure 7 represents the highest granularity level of RVC-CAL JPEG design. For example, Figure 8 represents a finer granularity of the decoder block.

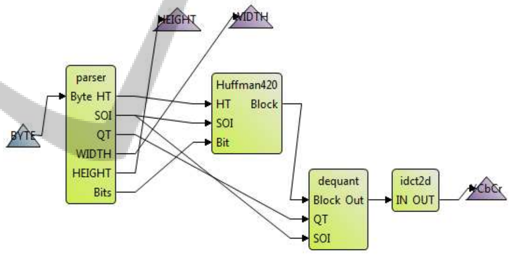

Figure 8: Finer granularity of JPEG decoder.

The next step consists in integrating the accordion algorithm in the *RawYCbCr* actor and the inverse accordion in the *YCbCrToMB* actor as presented in Figure 9.

We used Orcc compiler to generate the software implementation of the JPEG design and we obtained a set of .C files corresponding to every actor of the design and a top file that manages the actors scheduling and the FIFO data exchange.

Table 1 gives information about the proposed design cost in comparison with M-JPEG and MPEG 4 part 2 simple profile standards. Table 1 shows that the ACC-JPEG algorithmic decoding complexity is close to the M-JPEG one and it is much less than MPEG 4 part 2 one.

The design is evaluated with CIF and Q-CIF video sequences. Originally, the video frames look like Figure 10. By adding the accordion algorithm, we obtained the frames of Figure 11. Figure 12 shows a

* LOC:Line Of Code

Table 1: Complexity comparison between CAL and C descriptions.

decoders	level	actors	Parser LOC* CAL	Parser LOC* C
M-JPEG	2	6	184	979
ACC-JPEG	2	7	184	979
MPEG 4 SP	3	27	1285	4720

comparison in terms of rate-distortion between the ACC-JPEG, MJPEG and MPEG-4. The results shown are obtained with the sequence 'Hall Monitor' (CIF, 25Hz). We considered a general processor as an implementation target. The main configuration features of this target are presented in table 2

Table 2: Implementation target features.

processor	Intel dual core CPU
frequency	2,5 GHz
RAM	2 Go
operating system	Microsoft windows XP

Some results about the architecture performances (processing speed) and requirements in term of (memory and resources) are given in tables 3, 4, 5. All results are recorded in the same experimental conditions given in table 2.

Tables 3, 4 shows the coding frames frequencies recorded for CIF and QCIF videos. We can notice that the ACC-JPEG has almost the same processing (encoding and decoding) speed than M-JPEG coder.

Table 3: Throughput frequency ACC-JPEG VS M-JPEG encoders.

	encoding FPS	
resolution	ACC-JPEG	M-JPEG
CIF (288x352)	178	182
QCIF (144x176)	748	760

Table refACCJPEGMem illustrates the ACC-JPEG memory requirements in comparison with other standards coders.

The memory requirements can be divided into two by eliminating intermediate tables reserved for the construction of the Accordion Presentation and thus it will match the theoretical memory requirements as it was estimated in (Ouni et al., 2011). Even processing speed can also be improved and tends perfectly to the M-JPEG processing speed by adjusting the state machine which was initially designed for M-JPEG and then not perfectly adapted to the own ACC-JPEG constrains. Such state machine involves large timeouts resulting to the need to buffering of frames for the set up of the Accordion representation.

Table 4: Throughput frequency ACC-JPEG VS M-JPEG decoders.

	decoding FPS	
resolution	ACC-JPEG84	M-JPEG
CIF (288x352)	152	156
QCIF (144x176)	690	702

Table 5: ACC-JPEG memory consumption (GOF = 4).

component	memory consumption in Ko	
Image size	CIF	QCIF
ACC	304,128x4	76,032x4
Encoder	3,614	3,614
Decoder	66,080	33,240
ACC inverse	304,128x4	76,032x4
FIFO	65,536	32,768
Total	2568,254	677,878

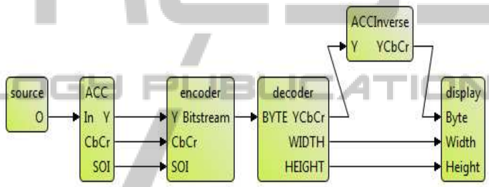

Figure 9: ACC-JPEG design.

Figure 10: Normal frame.

4 CONCLUSIONS

In this paper a CAL description of a new video coder is evaluated. The presented work constitutes a demonstration of an efficient video coder fast implementation. Regarding the proposed video coder tradeoff between performances and complexity of the proposed video coder and the tradeoff between efficiency and cost of the implementation methodology, the presented work could be considered as an economic and efficient solution for many video applications including embedded systems, digital multimedia devices,

Figure 11: ACC-JPEG output frames.

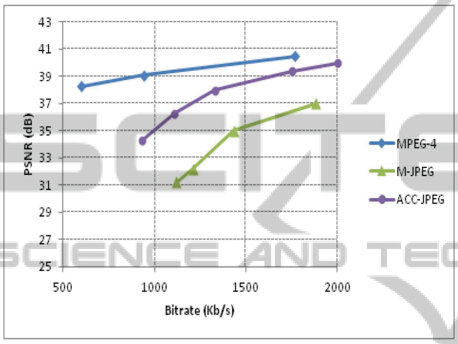

Figure 12: Performance comparison.

networks sensors which requires low computational processing and low time to market. Further orientations will include the implementation of the second Accordion-based-coder called ACC-JPEG 2000- proposed in (Ouni et al., 2010) as a more efficient version than one presented in this paper.

REFERENCES

Bhattacharyya, S., Brebner, G., Eker, J., Janneck, J., Mattavelli, M., von Platen, C., and Raulet, M. (2008). OpenDF - A Dataflow Toolset for Reconfigurable Hardware and Multicore Systems. First Swedish Workshop on Multi-Core Computing, MCC , Ronneby, Sweden, November 27-28, 2008.

Brooks, C., Lee, E., Liu, X., Neuendorffer, S., Zhao, Y., and (eds.), H. Z. (2004). PtolemyII - heterogeneous concurrent modeling and design in java (volume 1: Introduction to ptolemyII). Technical Memorandum UCB/ERL M04/27, University of California, Berkeley, CA USA 94720.

Eker, J. and Janneck, J. (2003). CAL Language Report. Technical Report ERL Technical Memo UCB/ERL M03/48, University of California at Berkeley.

Gokturk, S. B. and Aaron, A. M. (2002). Applying 3d methods to video for compression. In *Digital Video Processing (EE392J) Projects Winter Quarter*.

ISO/IEC FDIS 23001-4: 2009 (2009). Information Technology - MPEG systems technologies - Part 4: Codec Configuration Representation.

Janneck, J. W., Mattavelli, M., Raulet, M., and Wipliez, M. (2010). Reconfigurable video coding a stream programming approach to the specification of new video coding standards. In *MMSys '10: Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, pages 223–234, New York, NY, USA. ACM.

Ouni, T., Ayedi, W., and Abid, M. (2009). New low complexity dct based video compression method. In *Proceedings of the 16th international conference on Telecommunications*, ICT'09, pages 202–207, Piscataway, NJ, USA. IEEE Press.

Ouni, T., Ayedi, W., and Abid, M. (2010). New non predictive wavelet based video coder: Performances analysis. In Campilho, A. and Kamel, M., editors, *Image Analysis and Recognition*, volume 6111 of *Lecture Notes in Computer Science*, pages 344–353. Springer Berlin / Heidelberg.

Ouni, T., Ayedi, W., and Abid, M. (2011). A complete non-predictive video compression scheme based on a 3d to 2d geometric transform. *International Journal of Signal and Imaging Systems Engineering*, 4(3):164–180.

Yun Q . Shi, H. S. (1999). Image and video compression for multimedia engineering fundamentals, algorithms, and standards. In *CRC Press*.