Hardware/software 2D-3D backprojection on a SoPC platform

Nicolas Gac, Stéphane Mancini, Michel Desvignes

Hardware/Software 2D-3D backprojection on a SoPC Platform

Nicolas GAC, Stéphane MANCINI and Michel DESVIGNES
Laboratoire des Images et des Signaux
46 av Felix Viallet
38100 Grenoble, France
{nicolas.gac, stephane.mancini, michel.desvignes}@lis.inpg.fr

ABSTRACT

The reduction of image reconstruction time is needed to spread the use of PET for research and routine clinical practice. In this purpose, this article presents a hardware/software architecture for the acceleration of 3D backprojection based upon an efficient 2D backprojection. This architecture has been designed in order to provide a high level of parallelism thanks to an efficient management of the memory accesses which would have been otherwise strongly slowed by the external memory. The reconstruction system is embedded in a SoPC platform (System on Programmable Chip), the new generation of reconfigurable circuit. The originality of this architecture comes from the design of a 2D Adaptative and Predictive Cache (2D-AP Cache) which has proved to be an efficient way to overcome the memory access bottleneck. Thanks to a hierarchical use of this cache, several backprojection operators can run in parallel, accelerating in this manner noteworthy the reconstruction process. This 2D reconstruction system will next be used to speed up 3D image reconstruction.

Categories and Subject Descriptors

CAHC [**Computer Applications in Health Care**]: Computer applications for Medical Imaging

Keywords

3D reconstruction, PET, Adequation Algorithm Architecture

1. INTRODUCTION

Reconstruction of images in tomographic image modalities is cpu intensive and usually postponed. However, real-time reconstruction of the acquired data in positron emission tomography (PET) imaging would facilitate positioning of the subject, detect the potential problem during the acquisition and examine image quality. Real time reconstruction is also needed for large scale diffusion of clinical PET examinations, which are used for early detection of cancer, evaluation of disease spread and treatment response. Then, minimization of examination duration can decrease the cost of PET to make its powerful technology more widely available. The development of new activities such as micro PET, PET mammography, small animal PET imposes flexible, simple and fast system.

Image reconstruction is often partitioned into data acquisition, filtering and backprojection. In the past years, several works [1] on PET data acquisition have increased quality and speed of acquisition, using specific hardware for human PET. DSP and/or FPGA provide modular, scalable, programmable solutions which actually reduce research and development time and costs. On the other side, backprojection is a widely used technique in the field of tomography, including SPECT, CT, multislide CT. Software research and algebraic algorithms have increased the image quality (artifacts, signal to noise ratio) but are really more time consuming techniques than classical Filtered BackProjection reconstruction. Moreover, backprojection is a part of these reconstruction procedures.

There are several implementations of reconstruction system on parallel computers [2, 3], on specific hardware like ASIC or FPGA [4, 5, 6] or with hardware-software architecture [7]. An other strategy is to use the 3D classical processors like the GPU (graphic Processor Unit) to accelerate reconstruction [8]. The main bottleneck of all these kinds of system is the access to the memory storing the sinograms. Indeed the parallelization of the reconstruction process is limited by the bandwidth of the main memory.

In this paper, we present a FPGA-based 2D backprojection operator whose aim is to be used to accelerate 3D backprojection. The original idea is the way we overcome the memory access bottleneck thanks to a 2D Adaptive and Predictive Cache (2D-AP Cache). Afterwards, it has been build a hierarchical architecture of this cache allowing a parallelization of the backprojection. Results in time performance are evaluated and compared to standard procedures.

2. GOALS

2.1 3D Backprojection algorithm

Data acquired by the scanner are the Radon Transform of the body, with distribution function f, and is called the sinogram, S. The backprojection consists in summing up all the projection elements $p(x_r, y_r, \phi, \theta) = S_{x_r,\phi}(y_r, \theta)$ in order to reconstruct one voxel (volume element) $f^*(x, y, z)$. The

SAC'06 April 23-27, 2006, Dijon, France

projection space is a 4D space along (x_r, y_r, ϕ, θ). (x_r, y_r) are the coordinates on the projection planes :

$$\begin{cases} x_r = -x * \sin\phi_k + y * \cos\phi_k \\ y_r = z * \cos\theta_l - (x * \cos\phi_k + y * sin\phi_k) * \sin\theta_l \end{cases} \quad (1)$$

The sample values of the two angles of projection are :

$$\begin{cases} \phi_k = \dfrac{k\pi}{K} \text{ with } 0 \leq k < K \\ \theta_l = \dfrac{l * \theta_{Max}}{L} \text{ with } -L \leq l \leq L \end{cases} \quad (2)$$

One way to organize the data is to store the projections elements along (y_r, θ) in 2D sinograms S_{x_r,ϕ_k} defined by (x_r, ϕ_k). Computation of one voxel $f^*(x, y, z)$ becomes :

$$f^*(x, y, z) = \sum_{k=0}^{K} \sum_{l=-L}^{L} S_{x_r,\phi_k}(y_r, \theta_l) \quad (3)$$

In this manner, 3D backprojection is made up of a selection of sinograms along (x_r, ϕ_k) and a course through the selected sinograms, as shown on figure 1. The sinograms selection can be done in software and the course through sinograms accelerated thanks to a dedicated hardware operator as the one presented below. In this way, 3D PET reconstruction made with the 3D-RP algorithm mainly based on backprojections can be notably accelerated.

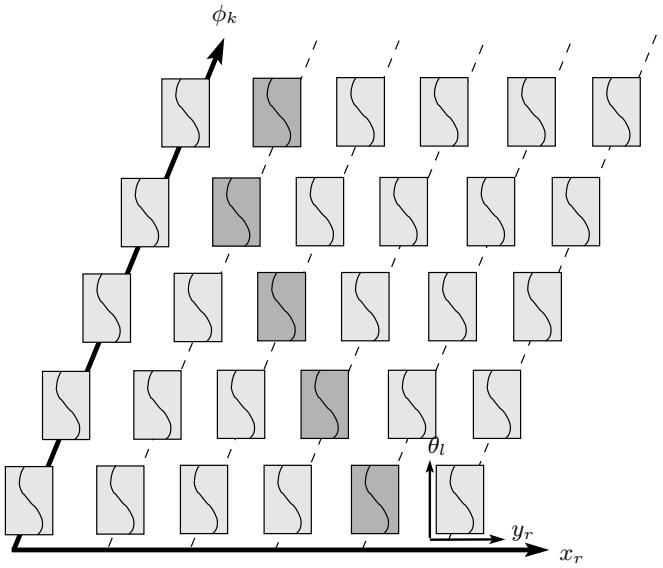

Figure 1: Selection of sinograms along (x_r, ϕ_k)

2.2 2D backprojection operator

This operator computes the backprojection of 2D data. It can be seen as a basic hardware operator for the 3D backprojection or as a system of reconstruction itself in 2D mode. In that case, this module reconstructs the pixel (picture element) $f(x, y)$ from data acquired from the scanner and stored on one sinogram S. Each column k of the sinogram corresponds to the orthogonal projection of the body on a line of r detectors orthogonal to the direction $\phi_k = \frac{k*\pi}{K}$, from the object x axis. Therefore, the sinogram's pixel $S(x_r, \phi_k)$

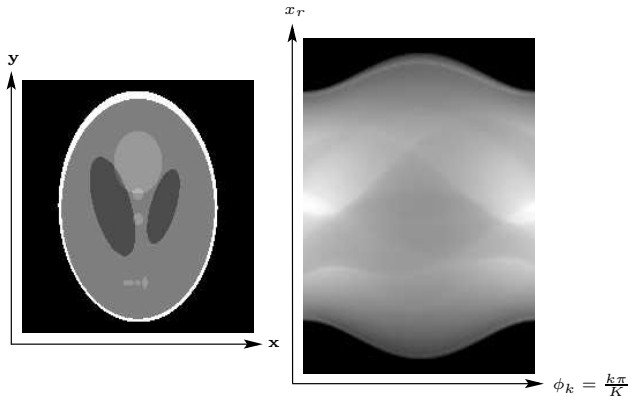

Figure 2: 2D Phantom image and its sinogram

is the sum of the image pixels on the Line Of Response (LOR) of the scanner which is perpendicular at the detector axis and passing by the detector of coordinate x_r :

$$S(x_r, \phi_k) = \sum_{(x,y)\in \text{LOR}} f(x, y) \quad (4)$$

From this sinogram, the algorithm rebuilds the image f^* by a backprojection of the K lines of the sinogram on the reconstructed object :

$$f^*(x, y) = \sum_{k=0}^{K} S(x_r, \phi_k) \quad (5)$$

$$x_r = x * \cos\phi_k + y * \sin\phi_k + \text{offset} \quad (6)$$

This method is not the exact inverse of the Radon transform. Indeed, it produces star-shaped artifacts and the reconstructed image is blurred, as shown on figure 3. To improve the reconstruction, one can filter the sinogram but this is out of the scope of this paper.

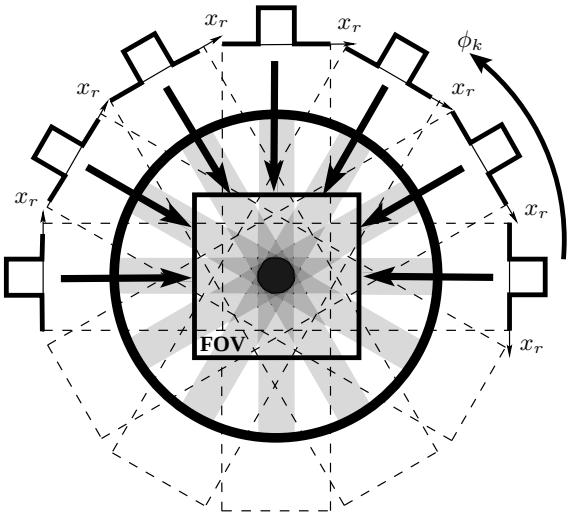

Figure 3: 2D backprojection and star-shaped artifacts

2.3 The challenge

As the sinogram is kept in a SDRAM like external memory, we need an efficient memory management to overcome its latency and allow a high level of parallelism. Standard cache based on a linear accesses to memory can't be a satisfactory solution because both of their complexity and their weakness to load the needed data. Indeed, memory accesses needed to reconstruct one single pixel $f^*(x, y)$ follow a sinusoid in the sinogram which is of poor address locality. To speed-up these memory accesses, a new cache mechanism is needed. Predicting which pixels the processing unit would use, will help the cache to download the needed pixels during the computing process.

3. ARCHITECTURE

3.1 System architecture

A low cost and efficient strategy is to fit the 2D backprojection algorithm on a SoPC (System on Programmable Chip). Because of its PCI interface between a PC and the system, we chose to evaluate the architecture thanks to an Avnet development board which contains:

- a Xilinx V2Pro with 2 PowerPC processor and 1 Million equivalent gate
- 32 MBytes SDRAM
- 2 MBytes SRAM
- 128 MBytes DDR-SDRAM

As one can see on figure 4, the SRAM is shared between the V2Pro and the host PC through a PCI interface. Thanks to a synchronisation mechanism the sinogram is transferred from the PC to the SDRAM and the reconstructed image from SDRAM to PC.

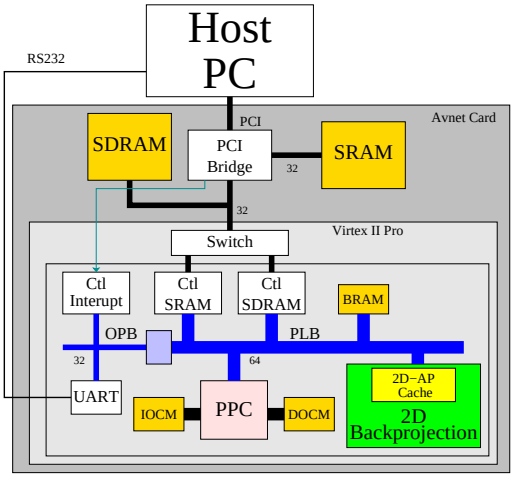

Figure 4: System architecture

The 2D backprojection system architecture, illustrated on figure 4 have three main components :

- a PowerPC 405 which manages the global process
- a hardware 2D backprojection unit
- the external SDRAM which stores the sinogram

The whole system is built around a 64 bits PLB Coreconnect system bus.

The 2D backprojection unit performs the reconstruction of several pixels of f^* of an arbitrary pattern. As shown on figure 5, this unit is master on the PLB bus and downloads the needed data from the SDRAM along the reconstruction process. The 2D-AP Cache module fully exploits the 2D spatial locality which appears when the processing unit performs the backprojection on a set of neighbouring pixels and allows an almost non-stopping computing. These mechanism are described in section 4 and justified on the next one.

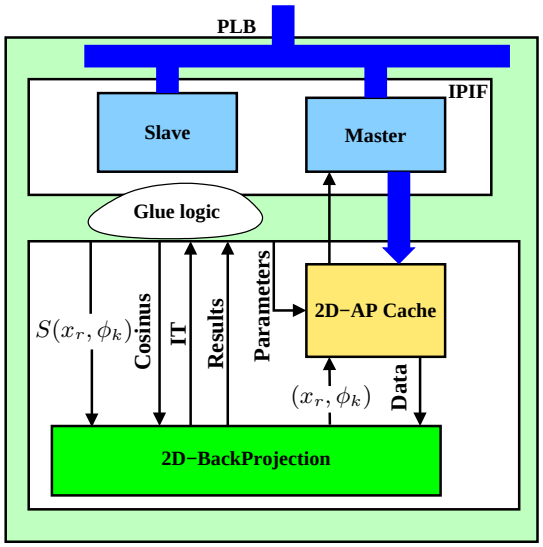

Figure 5: 2D backprojection unit architecture

3.2 Increasing temporal and spatial locality

As a single point reconstruction needs to sum the pixel's sinogram along a sine curve, memory accesses become a bottleneck if we use a SDRAM to store the sinogram. Furthermore, memory access latency is increased if the 2D backprojection unit is inserted in a SoPC as data has to move from the SDRAM controller to the processing unit through system buses. Indeed, the memory accesses have been a crucial point for several designs [9]. Spatial locality has to be improved "artificially" because the sine curve one is poor even considering 2D locality or address locality.

To increase spatial locality, the 2D backprojection unit performs the reconstruction of a set of neighbouring f^* pixels which could form any arbitrary pattern. For a square pattern to be reconstructed, 2D locality of the sinogram pixels appears :

- x_r locality comes from the fact that, for a given angle ϕ_k, the n^2 needed sinogram's pixels are in a segment of maximum length $\sqrt{2} * n$ pixels, which is the projection of the $n * n$ square pattern at angle $\frac{\pi}{4}$. A pixel of this segment is used $\frac{n^2}{\sqrt{2} * n} = \frac{n}{\sqrt{2}}$ times on average.
- ϕ_k locality, for a given position x_r, is due to the continuity of the projection operator. If a pixel (ϕ_k, x_r) has

been used then we are likely to use the pixel (ϕ_{k+1}, x_r).

The sinogram pixels used to reconstruct a set of f^* pixels are in the union of all their corresponding projection curve, which forms a kind of "tube" for adjacent f^* pixels. This "tube" itself looks like a sine curve and its height (in x_r axis) depends on ϕ_k.

To speed-up the reconstruction process and parallelize it, we are likely to design a memory hierarchy and store parts of the sinogram in fast embedded memories to fully exploit the 2D locality of accessed pixels. Temporal locality is obtained by updating the reconstructed pixel pattern for each ϕ_k and sinogram pixels used are in a x_r segment. An efficient strategy would be to compute the "tube" border and download data before we actually need them but it would be difficult to design because of this border complex equation.

The 2D-AP Cache is used as the memory hierarchy and frees us from the design of a complex "tube" border estimation. Indeed, this cache, which behaviour is described in section 4, dynamically estimates the 2D zone of accessed pixels and tries to predict future pixels.

3.3 Trade-off between memory and speed

Spatial locality, as demonstrated in previous section, increases with n, the reconstructed block size, and, at an extremum, would be the most efficient if the whole sinogram was stored in embedded memory, which is not realistic. Furthermore, as n increases the more we need embedded memories to store the intermediate data necessary for the reconstruction process. Indeed, we need memories to store:

- the (x, y) coordinate of each reconstructed pixel, to compute (6)
- the current sum for angle ϕ_k (5)

Their size increases with n^2 and may be limited by the amount of available embedded memory. One has to find a trade-off between the cache efficiency and the available memory.

3.4 Parallelism thanks to a hierarchical cache

To reduce computation time, the backprojection operator is parallelized and a hierarchical memory provides the data to the modules. In this way, every module gets its data from one 2D-AP Cache which itself updates its data from a 2D-AP Cache of a higher level. In our architecture, we have a two level cache, as shown on figure 6.

This original memory architecture needs only one external memory storing the whole sinogram unlike the solution proposed by [6]. It allows :

- recovery of the external memory and system bus latency
- reduction of data flow to the external memory

The more bus latency is reduced, the more efficient is this hierarchical memory. A direct connection of the memory to the backprojection unit would greatly ease a massive parallelization.

4. THE 2D-AP CACHE

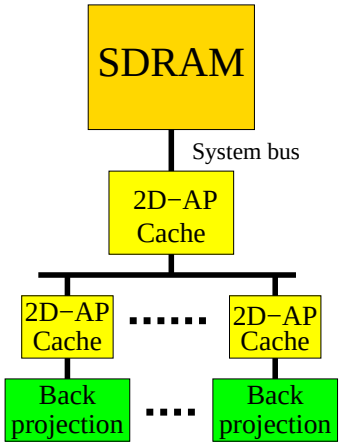

Figure 6: Hierarchical 2D-AP Cache

4.1 The 2D-AP Cache behaviour

Basics of the 2D-AP Cache is to copy 2D zones of the main image in cache memory to speed-up memory accesses, as shown on figure 7. It aims at predicting which pixels the processing unit would use. Doing so we reduce cache misses while the needed pixels are moving vertically and horizontally. To predict the cached zone position and geometry, we dynamically measure the mean and pseudo-standard deviation (PSD) of 2D coordinates issued by the processing unit. Low-pass IIR filters are applied to these coordinates to obtain the mean (filtered coordinates) and the PSD which is the differential between the current and filtered coordinate. Assuming an uniform distribution of the points around the mean, we are sure that most of the points would be included in a square equal to two times the PSD around the mean for a short period of time. We can load this zone in cache to reduce cache misses.

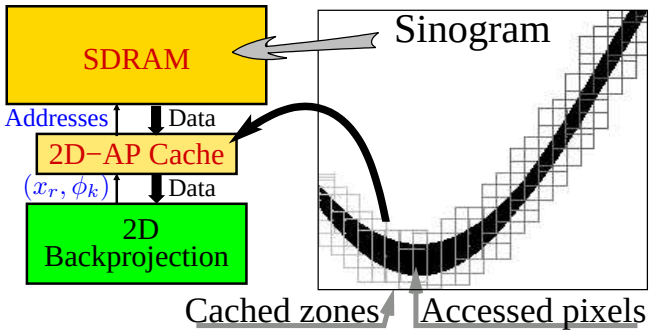

Figure 7: The 2D-AP Cache concept

A tracking mechanism is needed if the processing unit issues coordinates evolving along complex paths. The cached zone moves each time the computed mean is too different from the current cache center. Therefore we define a guard zone around the current cache center as shown in figure 8. The cache doesn't move while the mean is inside this guard zone and moves when it passes it. As we extrapolate a first order system, at constant speed, the guard zone size is dependent the PSD. Because the mean may be close to the guard border, the actual cache size is three times the PSD to keep a security zone around the mean position.

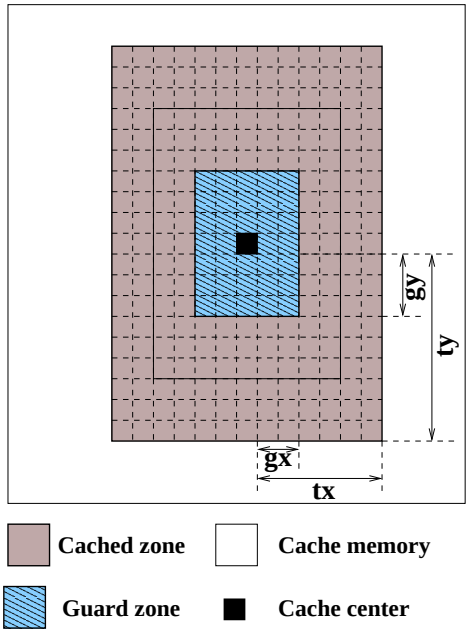

Figure 8: The 2D-AP Cache zones

To reduce the chip size and to get good performances, IIR filtering is performed with a single addition. Indeed, the filters are low-pass first order IIR filters which recursive equations are $s_n = at_n + (1-a)s_{n-1}$. We note $s = \mathrm{f}^a(t)$ and the parameter $1-c$ is equivalent to the time constant of an RC filter. Choosing c a negative power of 2 and the series s and t in fixed point arithmetic, the product becomes a simple shift. To keep the cache flexibility, the shift can be programmable and the issued coordinates can be down sampled.

4.2 The 2D-AP Cache parameters control

The 2D-AP Cache state is fixed by the following set of variables:

- $P = (x_n, y_n)$, pixel coordinate of the last used pixel
- $P_f = \mathrm{f}^a(P) = (x_{f_n}, y_{f_n})$, mean of P
- $d = |P - P^f|$, current computed PSD
- $d_f = \mathrm{f}^b(d)$, mean PSD
- c, current cache center
- v, current cache PSD, which defines:
  - $g = \gamma v$, guard zone size
  - $t = \tau v$, cached zone size
  - $s = \alpha v$, cache speed

The cache parameters are:

- a, b, the cut-off frequencies
- γ, τ to compute the guard and cached zone size
- α which is the cache speed

The α speed factor has a big influence on the cache stability because a high α may move the cache too fast. Indeed, if the cache is too fast, the P_f point can be out of the new guard zone, which makes the cache move backward. To allow fast move, the cache is stabilized by forcing the P_f point to the new cache center. Experiments have proved that such a mechanism is efficient as shown on figure 9.

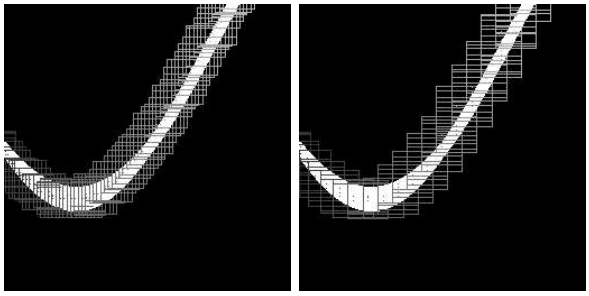

Figure 9: Influence of the α parameter (low α and high α)

4.3 2D-AP Cache architecture

The 2D-AP Cache allows concurrent read accesses from the processing unit and writes from the system bus master interface, as illustrated figure 10. Indeed, the cache initiates transfers on the system bus (CoreConnect bus) when it's internal state causes the downloading of a new zone in the embedded memory. The cache's internal memory is controlled by a central control state machine which drives the two statistical analysis modules on each coordinate's axis. This control unit also performs the address mapping between the bus master, the RAM memory and from the processing unit (here the 2D backprojection unit).

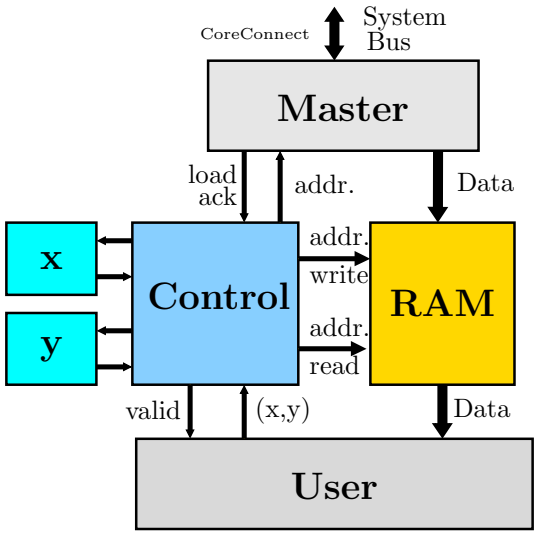

Figure 10: Cache architecture

The cache is none blocking and allows the user to access pixels while a new cache zone is loaded if those pixels are in the common zone between the previous cached zone and the new. To make the cache stable, cache misses are not performed while loading a new zone.

5. RESULTS

5.1 Time performance

Simulations and measures on the platform show us that the hierarchical memory chosen is efficient and allows us to accelerate noteworthy the 2D backprojection thanks to the operator parallelization. This is done with only one external memory storing all data. The results are presented in the table 1 and show an almost linear acceleration along the number of operators. These reconstructions are done for a $x_{max} * y_{max} = 320 * 320$ image (400 blocks of $16 * 16$ pixels) from a sinogram with an angular resolution of $K = 512$.

System	Clock Cycles	Time
Software		
Pentium 3 (1 GHz)		3,5 s
PPC (VirtexII-Pro)		94 s
Hardware		
Ideal	$52 \cdot 10^6$ $(320 * 320 * 512)$	1,04 s
Without cache	$52 \cdot 10^6 * 28$ (Ideal∗Latency)	29,12 s
1 unit	$78 \cdot 10^6$	1,56 s
parallelized hardware		
2 units	$42 \cdot 10^6$	0,84 s
4 units	$21 \cdot 10^6$	0,42 s
9 units	$11 \cdot 10^6$ (simulated)	0,22 s

Table 1: Time acceleration with the backprojection operator @50 Mhz

Ideally we could execute a reconstruction without deadtime, that is to say within $x_{max} \cdot y_{max} \cdot K = 320 \cdot 320 \cdot 512$ clock cycles. These ideal performances are degraded on the one hand by the synchronisation between the hardware operator and the PPC, and on the other hand by the 2D-AP Cache performances on the system bus. The measures show us that the simulations made for one block $(320 \cdot 320)$ reconstruction are reliable and can be extrapolated to the reconstruction of a complete image $(512 \cdot 512)$.

System	Clock Cycles	Time
Ideal	$3 \cdot 10^9$	60 s
1 unit	$4.5 \cdot 10^9$	90 s
2 units	$2.25 \cdot 10^9$	45 s
4 units	$1.125 \cdot 10^9$	22.5 s
9 units	$0.5 \cdot 10^9$	10 s

Table 2: Reconstruction time estimated for 3D backprojection with an overhead of 50%

On the table 2, the computation time for 3D backprojection have been estimated. The final backprojection of 3D-RP algorithm is estimated for the reconstruction of a $x_{max} \cdot y_{max} \cdot z_{max} = 128 \cdot 128 \cdot 63$ volume from Siemens HR+ data with a span of 9, a maximum ring difference of 22 and a mash factor of 0. Each voxel reconstruction needs to access $N_\phi \cdot N_{seg} = 576 \cdot 5$ pixels. So we have to access to about $3 \cdot 10^9$ pixels in memory. Ideally the reconstruction can be done in about $3 \cdot 10^9$ clock cycles. We put an overhead of 50% for the estimation of the reconstruction time by one computation unit using a 2D-AP cache. This 50% are an estimation of the degradation due to synchronisation and cache performance on the system bus. For 2D backprojection, this overhead have been measured and presented on table 1 : its value is about 50%. Then for the parallelized hardware, this reconstruction time is divided by the number of units.

5.2 System complexity

The entire operator has been written in generic and parametrizable VHDL and this to quickly explore the different possible configurations. The cache is entirely modular in order to measure the effectiveness of the different kinds of predictive estimators and to ease the design of a hierarchical cache as well. The table 5.2 gives the complexity of the system in number of LUT for the synthesis on Xilinx VirtexII-Pro target.

Block	CLB	FG
1 unit	1078	2155
2 units	2253	4506
4 units	3877	7753

Table 3: 2D backprojector operator synthesis

5.3 Discussion

Better reconstruction times can be reached by increasing the computational power and reducing the memory latency. Indeed, we measured an average latency of 30 clock cycles on the PLB system bus and the 2D-AP Cache efficiency is increased for lower system bus latency, as shown on figure 11.

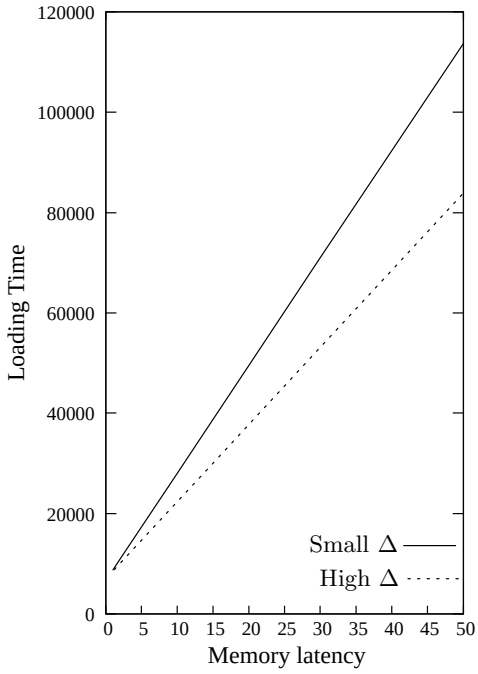

Figure 11: Cache loading time along system bus latency for different amount of displacement (high α or low cut-off frequencies)

The cache's load is done by horizontal lines therefore the time model in each move direction is the following :

- Horizontal move
  $\text{time}_H = (2 * t_y + 1) * (\text{Latency} + \frac{\Delta_{t_x}}{\text{Bandwidth}})$
- Vertical move
  $\text{time}_V = \Delta_{t_y} * (\text{Latency} + \frac{2 * t_x + 1}{\text{Bandwidth}})$

Small cache's amounts of displacement Δ_{t_x} cause the loading time mainly dependent on the memory latency. Figure 11 shows us that one can reduce the loading time by increasing the 2D-AP Cache movements when the latency is high. It can be done by reducing the cut-off frequencies or increasing the speed factor α. Doing so, the cache size as to be increased and one has to find a trade-off dependent on the available resources.

6. CONCLUSION AND FUTURE WORK

In this paper, a FPGA-based image reconstruction system has been presented, implementing the classical algorithm of 2D backprojection. This implementation is a basic hardware operator used by a 3D reconstruction system. The 2D-AP Cache outperforms the bottleneck of memory accesses. Indeed, the effectiveness of the cache is related to the implementation of the backprojection, which increases the spatial and temporal localities. During the reconstruction, the data needed can be statistically predicted. Then, the cache can load these needed data before they are actually used. The 2D backprojection module has been duplicated for parallelization in a hierarchical way : the spatial locality ensures that several pixels can be reconstructed simultaneously, increasing the reconstruction speed along the number of operators. As it has been shown, this 2D backprojection module can be easily used to accelerate a 3D backprojection as the one used in the 3D-RP algorithm. This accelerated 3D backprojection could be used in the more sophisticated iterative algorithm as well. These algorithms give better image quality but are much more time consuming. This first implementation on FPGA is a first step to build a scalable and flexible reconstruction system.

7. REFERENCES

[1] M.S. Muscrok et al. Performance characteristics of a new generation of processing circuits for pet applications. *IEEE Tr. Nucl. Sci.*, 50(4):974–978, August 2003.

[2] D.W. Shattuck, J. Rapela, E. Asma, A.Chatziioannu, J.Qi, and R.M. Leahy. Internet2-based 3D PET image reconstruction using a PC cluster. *Phys. Med. Biol.*, 47(15):2785–2795, August 2002.

[3] S. Vollmar, C. Michel, J.T. Treffert, D.F. Newport, M. Casey, C. Knoss, K.Wienhard, X. Liu, M.Defrise, and W.-D. Heiss. Heinzelcluster accelerated reconstruction for FORE and OSEM3D. *Phys. Med. Biol.*, 47(15):2651–2658, August 2002.

[4] I. Goddard and M. Trepanier. High-speed cone-beam reconstruction : An embedded systems approach. In *Proc. SPIE Medical Imaging Conf.*, pages 483–491, February 2002.

[5] Nikolay Sorokin. *An FPGA-Based 3D Backprojector.* PhD thesis, Universitat des Saarlandes, Allemagne, 2003.

[6] M. Leeser, S. Coric, E.Miller, H. Yu, and M.Trepanier. Parallel-beam backprojection an FPGA implementation optimized for medical imaging. *J. VLSI Signal Proc.*, 39(3):295–311, March 2005.

[7] J. Muller, D. Fimmel, R. Merker, and R. Schaffer. Hardware- software system for tomographic reconstruction. *J. Circuits Syst. Comp.*, 12(2):203–229, April 2003.

[8] F. Xu and K. Mueller. Ultra fast 3D filtered back projection on commodity graphics hardware. In *Proc. IEEE Int. Symp. Biomedical Imaging (ISBI'04)*, pages 571–574, Arlington, USA, April 2004.

[9] S. Coric, M. Leeser, E. Miller, and M. Trepanier. Parallel-beam backprojection an FPGA implementation optimized for medical imaging. In *Proc. ACM Int. Symp. Field-Programmable Gate Arrays (FPGA'02)*, pages 217–226, February 2002.