IADIS International Journal on Computer Science and Information Systems
Vol. 8, No. 2, pp. 72-90
ISSN: 1646-3692

VLSI PLATFORM FOR REAL-WORLD INTELLIGENT INTEGRATED SYSTEMS BASED ON ALGORITHM SELECTION

Martin Lukac. *Graduate School of Information Sciences, Tohoku University, Sendai, Japan.*

Michitaka Kameyama. *Graduate School of Information Sciences, Tohoku University, Sendai, Japan.*

Yoshichika Fujioka. *Department of Engineering, Hachinohe Institute of Technology, Hachinohe, Japan.*

ABSTRACT

A real-world intelligent system consists of three basic modules: environment recognition, prediction (or estimation), and behavior planning. To obtain high quality results in these modules, high speed processing and real time adaptability on a case by case basis are required. In each of the above mentioned modules, many different algorithms and algorithms networks exists and provide various performances on a case by case basis. Thus, a mechanism that for any of the three computational stages selects the best possible algorithm is required. We propose a platform based on the algorithm selection approach to the problem of natural image understanding. This selection mechanism is based on machine learning; a bottom-up algorithm selection from real-world image features and a top-down algorithm selection using information obtained from a high level symbolic world description and algorithm suitability. To accommodate the high-speed processing requirements, the high-frequency of real-time reconfiguration and a low-cost of implementation, we are using present a novel dynamic reconfigurable VLSI processor for real-time adaptation of the algorithm selection. The new architecture includes a fine-grain Digital Reconfigurable Processor, a distributed configuration memory to solve the data transfer bottleneck and an intra-chip packet routing scheme to reduce the size of the configuration memory.

KEYWORDS

Natural Image Processing, Image Understanding, High Level Verification, Algorithm Selection, Reconfigurable VLSI, Memory Reduction.

1. INTRODUCTION

The complexity and the diversity of the environment make the robust real world information processing at most satisfactory and only for some particular applications. Such low reliability is on one hand due to the limited suitability of algorithms used for the processing of real world information (Figure 1); each algorithm provides the best result for a particular set of environmental configurations (conditions). The common approach to solve this problem is an incremental increase of algorithms' functionality that allows taking into account the newly encountered environmental configurations. This approach however often results in algorithms with very high complexity, with limited scalability and with a continuously decreasing performance; a single aggregated algorithm cannot always robustly process the real-world information for all available conditions. But such robust performance is crucial in many real world applications such as intelligent cars or service robots. On the other hand, the processing of real-world information is limited by the hardware resources: bus width for the CPU-Memory transport, size of the cache memory for real-time context reconfiguration. Such restrictions are in particular observable in robotics where high performance and low power resources are required to provide computation necessary. Thus to improve the real-world information processing the improvement must be both done on the software as well as on the hardware level.

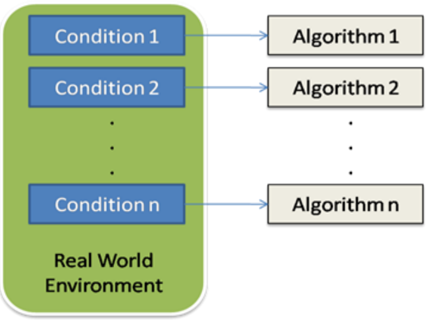


Figure 1. Many algorithms are necessary to process various instances of the real world environment

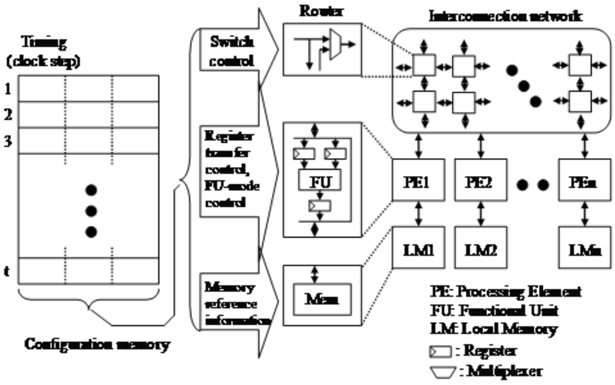


Figure 2. Example of a Conventional Memory for a Dynamically Reconfigurable Processor (DRP)

In this paper we describe a platform that combines the algorithm selection approach to image processing with a parallel architecture using the packet control transfer scheme and the configuration memory size reduction. That is, given a set of algorithms and a set of distinctive features computed on the input information, the selection paradigm allows using the best algorithm on a case by case basis. The main problems of the algorithm selection implementation are the requirement for large number of processing cycles, high speed processing and real time reconfiguration. The algorithm selection platform is not thus suitable for a standard CPU, SIMD or FPGA like architectures. While the CPU provides a relatively high speed of reconfiguration it is too general and thus too slow for the requirements of the platform. SIMD architectures like the GPU is much faster but in general its reconfiguration is much slower and the porting of algorithms to such structure is much more costly. Finally the FPGA provides a very high speed reconfiguration but it processing speed is even slower of that of a CPU.

As an implementation, a VLSI Platform is investigated with the target being the implementation of intelligent systems for real-world applications such as a highly-safe intelligent system, an intelligent robot system and an information appliance system. A real-world intelligent system consists of three basic modules: environment recognition, prediction or estimation, and behavior planning.

The best choice for the algorithm selection platform is thus such hardware such as the Dynamically Reconfigurable Processors(DRP) (Figure 2). Recent coarse grain (DRPs) have been developed to achieve high performance and flexibility for a fraction of the cost of an Field Programmable Gate Array (FPGA) [Amano2006]. A dynamic reconfiguration method is called "multi-context" reconfiguration [Tang2000, Sugawara2004, Kodama2006, Saito2005, Motomura2002, Levine2005, Veredas2005]. In this method, each Processing element (PE) provides a memory module that stores the configuration data sets for the corresponding PE and interconnection of surrounding buses. Although context switching can be done with a clock cycle in multi-context devices, the area of each PE is increased with the distributed context memory. The area of configuration memory provides 32 contexts is almost the same as that of a PE itself [Veredas2005].

To realize a low-cost, high-performance and low-power VLSI platform, reconfigurable VLSIs (Figure 2) is superior to the conventional FPGAs are designed and implemented using the intra-chip packet routing scheme [Fujioka2012]. In the conventional DRPs, control words required in the processing operations must be provided clock by clock. To reduce the configuration memory size, we propose a register-transfer-level packet data transfer scheme based on the fact that every control operations can be replaced by register transfer control [Honma2005, Fujioka2006]. The control of the data-path components at each clock cycle can be done by register transfer control realized by the packet transfer control scheme. A larger number of PEs can be provided to perform ultra-highly parallel processing in the same chip size, if we can reduce the configuration memory size.

For ultra-highly parallel processing, a router size must be designed as small as possible, because a large number of routers are required in the fine-grain packet data transfer. The semi-autonomous packet routing scheme is proposed based on hybrid utilization of packet data transfer and scheduling/allocation which has completed prior to execution. The semi-autonomous packet routing scheme is effectively employed for making a router as simple as possible, because arbiters or buffer memories in a router is not required to make packet collision free.

This paper is organized as follows. Section 2 introduces the concept of algorithm selection and describes the software platform for image understanding. Section 3 shows the method for reducing memory by manipulating the Bayesian Network used for algorithm selection. Section 4 describes the high level mapping of the algorithm selection platform onto the hardware and Section 5 describes the details of the Dynamically Reconfigurable Processor, the packet routing, local memory and shows some experimental results in the local memory reduction by our approach. Finally Section 6 concludes the paper.

2. ALGORITHM SELECTION PLATFORM

2.1 Algorithms Selection

The algorithm selection paradigm was originally introduced by Rice [Rice1976] and since various but only a relatively small amount of applications and studies have been made. Previous works related to image processing includes mainly the work of Yong [Yong2003] that used algorithm selection for segmentation in noisy artificial images and by Takemoto [Takemoto2009] that used algorithm selection to determine best edge algorithm for edge detection in biological images. With respect to general robotic processing [Lukac2012, Lukac2013] introduces the concept of algorithm selection into middle and high level processing of natural image segmentation and understanding. In particular, in [Lukac2012\ the algorithm selection was used to improve the segmentation of natural images.

The algorithm-selection framework is shown in Figure 3.

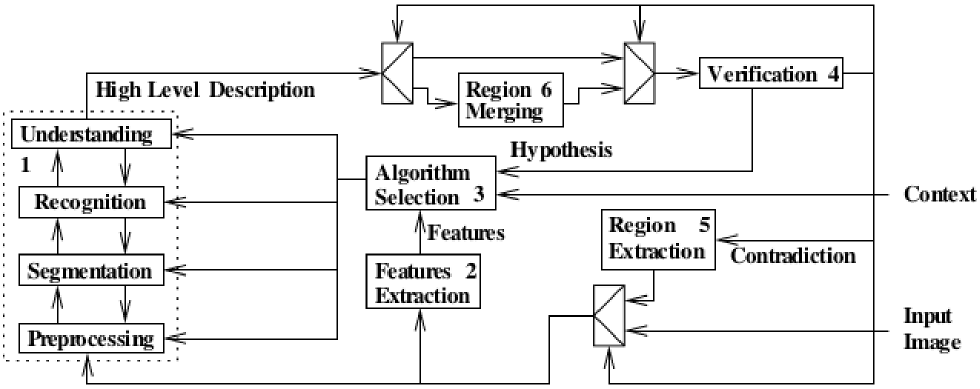


Figure 3. Algorithm Selection Based Platform for Image Understanding

The system operates as follows:

- A first cycle of processing starts with the features of the whole input image are extracted (Box 2) and are used as input to the algorithm selection mechanism (Box 3) that determines what algorithms should be used for processing the image (Box 1). The result of the processing is a high level description and thus the selection process select algorithms in all level of processing. The high level description is then verified(Box 4) for the correctness of the symbolic content and is analyzed for the existence of a logic contradiction. If contradiction was not detected the processing stops.

- A second cycle starts when a contradiction was detected in the high level description. The contradiction is used to extract the region where the contradiction is located (Box 5). The features of the region are extracted again (Box 1). At the same time, the contradiction and the high level description are used to generate hypothesis resolving the contradiction in the high level description. The resulting information: region features, high level description and user specified context information are used as input nodes in a Bayesian Network. The output of the BN is a set of different algorithms that are applied only to the extracted region. The result of processing of the contradiction region is merged back into the whole image (Box 6). The new high level description is verified again and if contradiction occurs the process is repeated. This loop is repeated until there is no more contradiction in any region of the image.

Notice, that the high-level representation is \em validated \em by the correct set of algorithms: only the correct algorithms will generate description without contradiction. Such processing however requires many processing cycles to obtain the desired result. This is due to the fact that each of the mechanism in the loop is in generally inaccurate and various combinations of algorithms needs to be tested to obtain a high level representation without contradiction [Lukac2013]. Such requirements - high speed processing and many various algorithms - are an ideal application for high speed reconfigurable VLSI implementation.

The algorithms used in this platform are highly heterogeneous. In the preprocessing level edge detection, noise removal, smoothing and color transformations are available. Some examples are of algorithms are Canny or Prewitt edge detection. The segmentation contains various algorithms and various features extraction that these algorithms require. Examples of features required for segmentation are salience, histogram of oriented gradients [Malik1999] or simply brightness. Segmentation algorithms such as Global probability of boundary [Maire2008], maximally stable extremal regions, Normalized Cut [Shi2000], Salience Based Segmentation [Donoser2009] can be used. The recognition processing contains only two distinct algorithms because recognition algorithms are a special case of pattern matching. The available algorithms are SVM matching [Carreira2012] and components matching using SVM [Felzenswalb2010], and a Bayesian network approach. In the final level of processing - the understanding - algorithms in general can be divided into two categories: logic and probabilistic. However in this work our focus is mainly on the diversity of algorithms in the first three level of processing [Lukac2013].

The most important fact is that the various algorithms in the proposed processing levels are highly heterogeneous and even within a same level the variety is relatively high. Moreover because the processing is adaptively changed not only for every image but also for various regions in the image, a real-time platform requires a high-speed platform permitting both the high speed of processing and of reconfiguration.

The selection mechanism initially used in [Lukac2011] was a machine learning based Features to Algorithms mapping similar to those used in [Yong2003, Takemoto2009]. However, using only features for algorithm selection was shown to be only partially successful and the best recorded selection precision was up to 60% [Lukac2012]. To improve the quality of algorithm selection [Lukac2013] proposed reduce the amount of information required for the algorithm selection, the proposed platform uses a hierarchy of information that is learned using a Bayesian Network as well as meta-information about algorithm suitability that permitting to determine exactly the information required on an algorithm-by-algorithm basis [Lukac2013].

The target applications of this approach is a Live-Feeling Communication Platform, where the robot controls remote live-event content delivery mechanism and modifies it based on user's emotional feedback. A different type of application is 3D image matching where using %This is possible because a mechanism called the Human Emotional-State Data Mining (HESDM) is designed that allows to estimate the expected robot behavior w.r.t. to the user’s intentions.

3. BAYESIAN INFERENCE AND HIERARCHY BASED NODE REDUCTION

The algorithm selection using features (loop 1 in Figure 3) improves the result of processing where only algorithms that do not require directly reasoning about the content of the image are used. The image understanding, object recognition and content/scene interpretation however might require algorithm selection based on symbolic information. More precisely, the content information can be used to provide information about where to apply particular algorithm or to determine the real-world environment cause of the algorithm failure (loop 2 Figure 3). Thus, finding the high-level cause of the algorithm failure provides additional information not obtainable directly from image features.

For instance assume Figure 4 shows a graph of relations between detected objects in a high level hierarchy. Notice that knowing there is a contradiction (as there is very rarely a TV on Sofa) resulting from understanding of the content, a hypothesis can be generated. The additional information provided by the hypothesis permits to select algorithm that fits both the properties of the image (features from the region contradiction) and the *virtual* properties of the hypothesis.

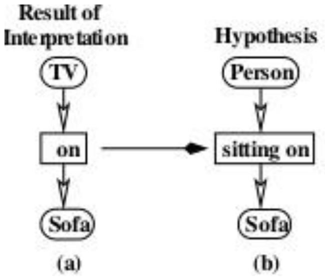


Figure 4. Example of graph obtained as a (a) result of interpretation of the image content, (b) result of hypothesis generation

However, estimation of the real world environment, properties and error causes is necessary because a reliable high-level interpretation resulting is rarely available and contains many errors. To solve this problem we use a hierarchy of knowledge built on measure for each algorithm. This means that for each algorithm the feature suitability as well as the algorithm’s performance is analyzed from the point of view of the environment and high-level description. Such analysis results in a set of advantages and disadvantages for each algorithm. The resulting information is built into a hierarchy that is used to minimize the amount of information by combining the various advantages and disadvantages of the various algorithms. The hierarchy is also used to determine how detailed the high level description must be.

Consider the two Figures 5 and 6. These Bayesian-like graphs represent the suitability of ALE [Ladicky2010] and CPMC [Carreira2012] algorithms respectively. These properties were obtained by the analysis of the generated data from the VOC 2010 set of selected 300 images. Each image was processed by both of the algorithms resulting in a set of detected objects and object contours.

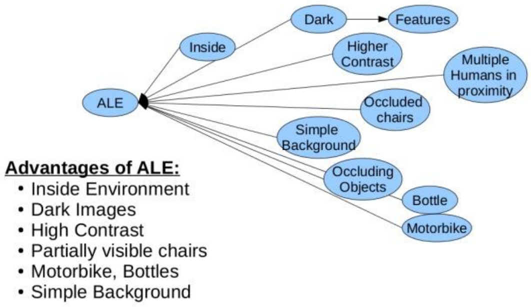


Figure 5. Advantages of the ALE algorithm

Each of the Bayesian Networks in Figure 5 and 6 selects the best algorithm very accurately if all the required information about the suitability would be available. But this is not the case because some of the suitability properties cannot always be obtained by the available algorithms directly. In the present case the high level image description contains a set of region labels (detected and recognized objects) and the relative position and size of the objects. Thus determining suitability conditions such as *complex background*, scenery location or *occluded chairs* is not directly possible. Moreover each algorithm's specialization to particular detection of object category must be minimized because for a general framework, the objects, their variation in space and features is too large to be effectively contained in a BN.

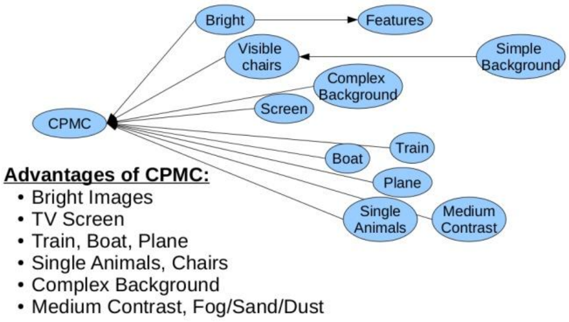


Figure 6. Advantages of the CPMC algorithm

The main problem with high specificity of the suitability properties is that in the BN it will result in at least one node that will estimate the algorithm suitability and that has as many input connections as there are advantages and disadvantages for all considered algorithms. The algorithm selection node in a BN deciding if the ALE of CPMC should be used will thus have 19 inputs. This number is expected to grow much faster than linearly with the number of algorithms because each used algorithm has in general more than one advantages and disadvantages.

To reduce these particular advantages and disadvantages and preserve the ability of the BN to distinguish algorithms we build a hierarchy of structured cause-effect real-world properties. Naturally to successfully however integrate algorithms' advantages and disadvantages into the real-world hierarchy, common properties of these advantages must be extracted.

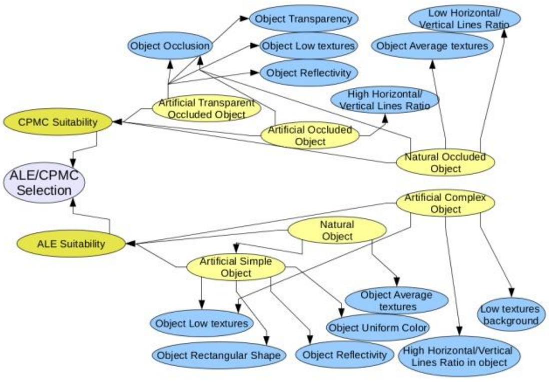


Figure 7. Bayesian Network using hierarchy to minimize the object properties

Building of the hierarchy starts by categorizing the possible sources of information. The categories of nodes that are used in this hierarchy are: entire image features, context advantages, background features, object advantages, interaction advantages, natural class group, action group and behavior group. The hierarchy used to minimize the objects advantages in two BN for both the ALE and CPMC algorithms is shown in Figure 7. Notice that the individual object categories disappeared and were replaced by both properties and a hierarchy of generalized object entities. Once all the objects are transformed into properties, new super nodes are created using common properties and representing super categories of the objects. Similar approach is used to minimize the other advantages such as the background or light properties.

The final BN is shown Figure 8. It was built using advantages of both the ALE's and CPMC suitability properties (Figure 5 and 6). For instance, the occlusion between two objects is estimated and used to select an algorithm. The selection quality with respect to the original BN's from Figure 5 and 6 was verified on a subset of the VOC 2010 database.

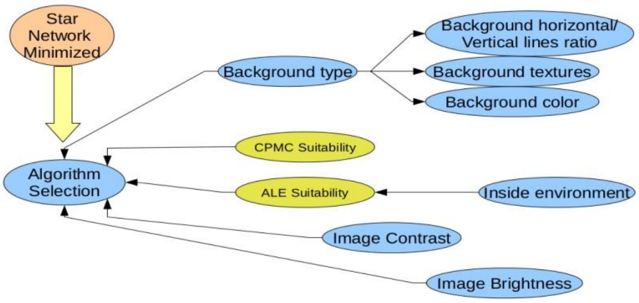


Figure 8. Bayesian Network used to select the most appropriate recognition algorithm

To verify the hierarchy introduced in this paper we used a set of 300 images from the VOC 2010 database for which the recognition results have been generated using the features selection from the loop 1 of the algorithm selection framework shown in Figure 3. Additionally set of contradiction have been generated for the images based on the rules shown introduced previously. For each of the images that contained a contradiction, the information necessary as input was also generated. In this manner, the network was first trained on 200 images and tested on 100 different images. The Bayesian Network used in the experimentation is shown in Figure 9. It was obtained by combining Figure 5-8 and minimizing by further analysis.

Table 1. Comparison of Results

Selection Method	Selection Accuracy
Feature based selection	70%
Non hierarchical BN selection	75%
Hierarchical BN selection	65%

Table 1 shows the precision of selecting algorithms using the BN and hierarchical BN. For comparison we also show the result compared with the feature based algorithm selection. Notice that as described in the method in this section, a correct determination of hierarchy to represent meta-level information is successful in both minimizing the amount of required knowledge and relative preservation the quality of selection.

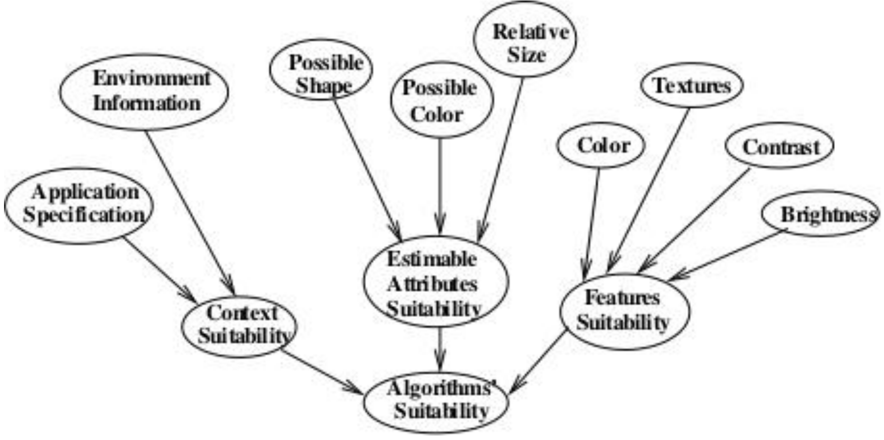


Figure 4. A BN network using hierarchy based on type of suitability of algorithms

4. MAPPING TO HARDWARE

The mapping of the algorithm selection platform requires a heterogeneous architecture that can efficiently accommodate all requirements of the platform. Figure 10 shows the various levels of the platform. In the previous section the block (a) was explained and the argument for high speed VLSI was given. Block (b) shows a high level schema of a heterogeneous system containing a I/O interface, a CPU, main memory and a set of reconfigurable processors all interconnected by a bus. Block (c) shows a detail of the reconfigurable processor showing a low speed but large capacity external memory and a high speed, low capacity local memory (configuration memory).

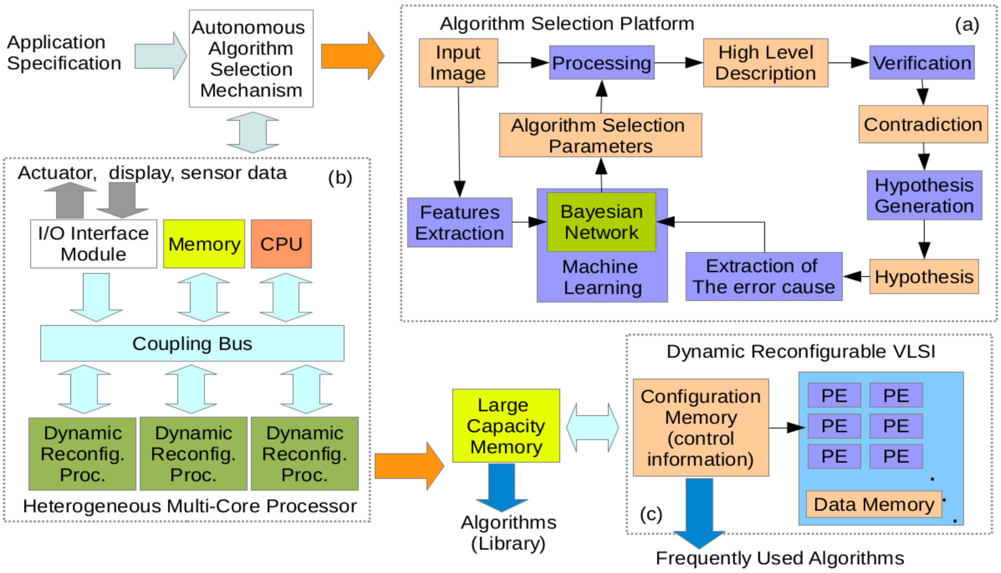


Figure 50. High level representation of the mapping from algorithm selection to hardware. Box (a) represents the algorithm selection platform for real world image understanding, Box (b) represents the high level system schema of the hardware and Block (c) shows the configuration memory within the reconfigurable VLSI.

The block (b) in Figure 10 shows the system level description of a heterogeneous platform required for the implementation of the algorithm selection approach. Elements of the boxes 3, 4, 5 and 6 (Figure 3) are static elements that do not need reconfiguration. The region extraction and merging (boxes 5 and 6 Figure 3) use multiple manners of extracting and merging regions but both extraction and merging is done by a single algorithm respectively. These two algorithms require a unified processing as the extraction and merging of regions cannot be efficiently split into a set of parallel tasks.

Box 3 (Figure 10) is the process that selects algorithms. It was initially developed as a LUT algorithm selector using machine learning methods (SVM, Adaboost or EM). However, in [Lukac2013] introduced a Bayesian Network for the algorithm selection that minimizes the amount of required resources for algorithm selection. Despite this, the BN for algorithm selection is a one-time configured structure that does not need reconfiguration in real time once the algorithms are known. Finally the block 4 is the verification process that also does not use reconfiguration in real time; it only operates on the high level description of the image as a logic verifier using first order logic rules. Thus, blocks 3, 4, 5 and 6 are mapped into a CPU and these elements are not in the main focus of this paper.

Blocks 1 and 2 in Figure 3 require the capability of high speed reconfiguration. Block 1 represents all available processing that is used in the platform. Algorithms must be configured for each input image and for each level of processing. Not only that, but when the loop 2 of the processing is entered, a single image is processed by multiple algorithms on a region-by-region case. Block 2 represents the algorithms used to extract image features. Because the algorithm selection platform uses various features and the algorithms are selected using a hierarchy of suitability of properties obtained from various features, the computation is done one feature at a time. This means that based on the input image and the high level feedback features are calculated on demand. Thus high speed processing and real-time reconfiguration is required in both block 1 and 2.

5. DYNAMICALLY RECONFIGURABLE PROCESSOR BASED ON A REGISTER-TRANSFER-LEVEL PACKET DATA TRANSFER SCHEME

High speed reconfiguration is only possible if the desired programming data is stored in a local, high speed configuration memory or cache. The problem is that such local memory is expensive and a large local memory requires also a larger address space slowing down the reconfiguration process. This is shown in Figure 10 (block c); a large and slow external memory contains all the available programs and the local high speed configuration memory serves as configuration cache. Consequently, to obtain the highest possible speed of reconfiguration, the configuration memory must contain the most often used algorithms and its size should be as small as possible. Such minimization is achieved by configuring the memory to match exactly the size of the stored program and by reducing the size of the address data.

To accommodate the variable size of programming data and minimal memory size we use the previously introduced dynamic reconfiguration of Local Memories (LM), where the number of LMs for each PE can be changed according to the data storage requirement [Fujioka2006, Fujioka2012]. Using these approaches, the hardware resource of memories can be sufficiently utilized for different algorithms in the algorithm selection scheme.

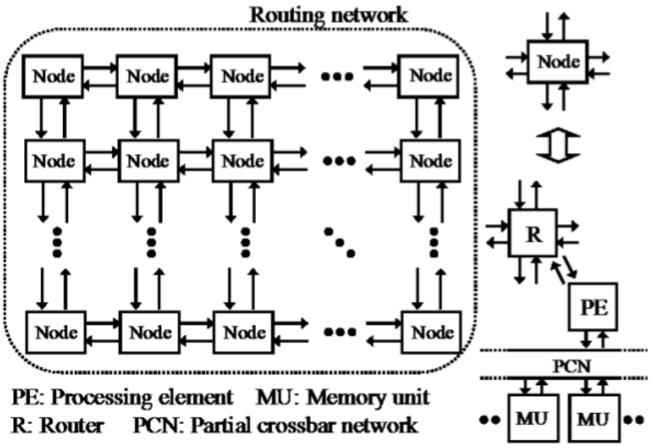


Figure 6. Two-dimensional mesh array structure for a DRP

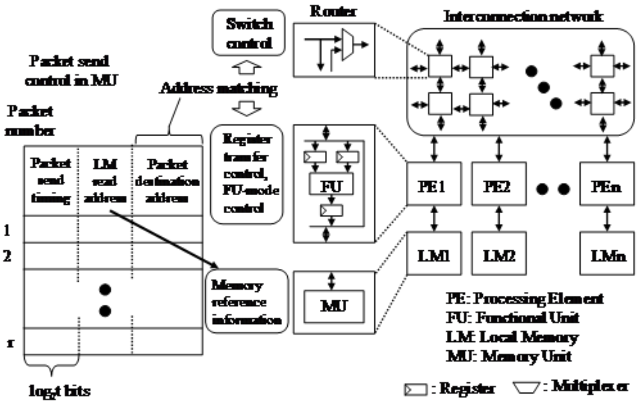


Figure 7. Configuration memory in the proposed DRP

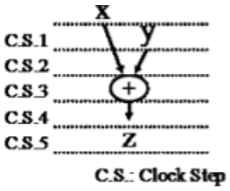


Figure 8. Example of a SDFG

5.1 Parallel VLSI Processor Architecture Based on Packet Data Transfer}

5.1.1 Packet-Data-Transfer-Based Routing Network Structure

Two-dimensional mesh array structure is typical parallel processor architecture. Figure 11 shows the DRP composed of multiple nodes. The node consists of a router, a PE and a set of memory units (MUs).

If an MU is connected directly with a PE without relaying another PE, the MU is called a local memory. For dynamic reconfiguration of the LMs, a partial crossbar network (PCN) is provided in the node. If the PCN switch between a PE and an MU turns ON, the MU becomes the LM of the PE. By the dynamic reconfiguration of LMs, the hardware resource of memories can be sufficiently utilized for different selected algorithms [Fujioka2006, Fujioka2012].

A packet generator (PG) and a router are provided in the node to control every data transfer between registers by packets. The packet data transfer between two PEs can be done through the routers directly connected to the PEs. Because fundamental operations of a processor are attributed to data transfer between registers, automatic timing generation of data-receive can be done if a packet is appropriately sent to a routing network. This means that timing control for data-receive can be autonomously completed, which is different from conventional configuration specification. Therefore, the configuration memory size of a DRP based on the register-transfer-level packet data transfer scheme is greatly reduced in comparison with a conventional DRP.

5.1.2 Principle of Packet Data Transfer Control

In the conventional coarse-grain DRP, all the control signals for every clock cycle such as switch control, register transfer control, Functional Unit (FU)-mode control and memory reference information are stored in the global configuration memory.

For realization of the same processing, a new processor structure based on packet data transfer control% as shown in Figure 12 was introduced in [Fujioka2012]. Configuration memories are distributed module-by-module in the processor. That is, they are provided at all of the routers, PEs and LMs (Figure 12).

All the control signals are generated from the packet control information stored in each configuration memory. It is sufficient to store only the packet control information which is used to generate active control signals, because no packet control is necessary during no-operation clock cycles. Therefore, the number of the effective clock cycles for packet control is limited to r which is smaller than the number of the clock steps t in the global configuration memory of a classical DRP.

The packet control information consists of the packet send timing; the LM read address and the packet destination address, where the bit length of the packet send timing is given by $\log_2 t$ bits for the timing identification.

Let us consider the difference between the global and distributed configuration memory based control methods using a simple example of a scheduled data flow graph (SDFG) shown in Figure 13.

Example: Distributed vs. Centralized Memory for Packet Data Transfer Based DRP

Figure 14 shows a processor hardware model for the processing of a SDFG. The data inputs x and y are stored in LM, and they are transferred to the inputs of the ALU, and latched into registers REG1 and REG2. Then, the addition of x and y is performed at ALU to produce the result z transferred to LM. In the conventional DRP, all the control signals for every clock cycles such as LM address, LM read control, LM write control, the load control of REG1 and REG2, and ALU mode are stored in the configuration memory as shown in Figure 15.

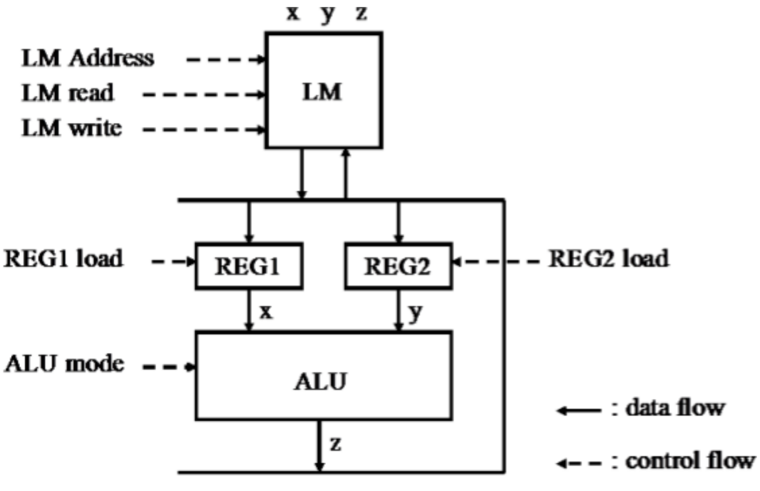
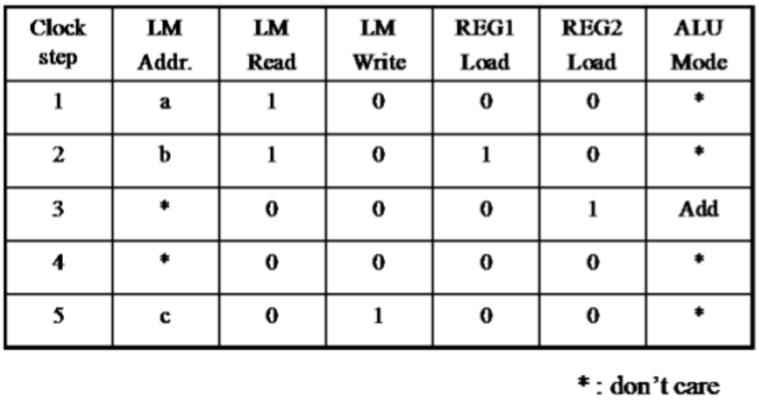


Figure 9. Conventional processor hardware model



Clock step	LM Addr.	LM Read	LM Write	REG1 Load	REG2 Load	ALU Mode
1	a	1	0	0	0	*
2	b	1	0	1	0	*
3	*	0	0	0	1	Add
4	*	0	0	0	0	*
5	c	0	1	0	0	*

* : don't care

Figure 10. Configuration memory for the conventional DRP

The same processing can be executed on the processor hardware model based on the packet data transfer scheme as shown in Figure 16. The data x and y are transferred to a packet generator PG1, where Packet1 is produced from its destination address information "1". Then, the packet destination address is compared with REG1 ID number at the comparator CMP. If the comparison result indicates "Equal" ("1"), the data x is latched into REG1 by the load control. Similarly, the ALU output result z is transferred to LM by the generation of a packet from PG2. Figure 17 shows the packet data control information stored in the modules LM, ALU, PG1 and PG2. At the cycle when the active packet transfer to REG1, REG2 and LM is not necessary for the processor operation, the corresponding packet control information is not required to be stored in the configuration memories. Therefore, great reduction of the control memory size can be achieved.

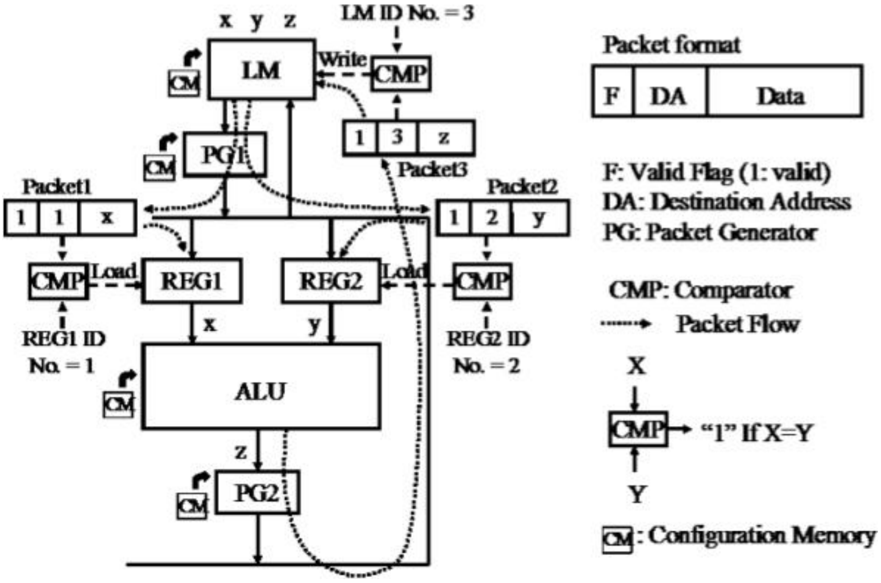
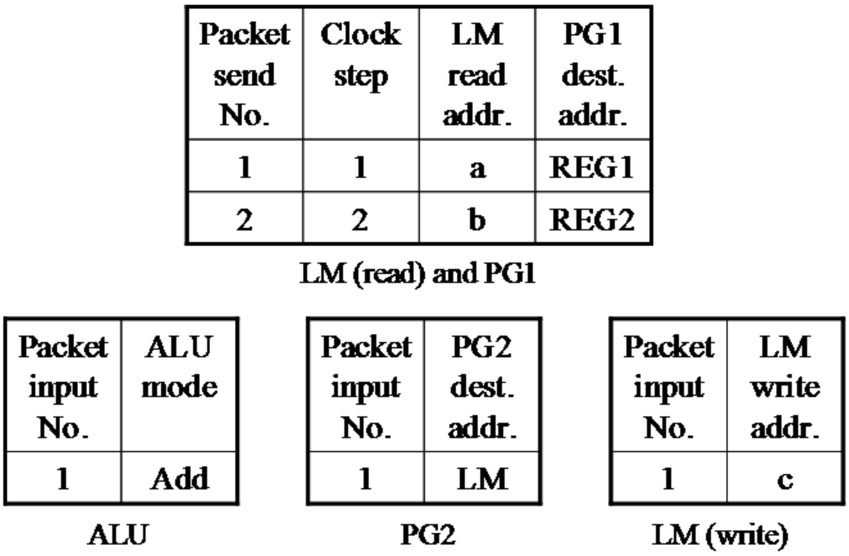


Figure 11. Processor hardware model based on the packet data transfer scheme



Packet send No.	Clock step	LM read addr.	PG1 dest. addr.
1	1	a	REG1
2	2	b	REG2

LM (read) and PG1

Packet input No.	ALU mode
1	Add

ALU

Packet input No.	PG2 dest. addr.
1	LM

PG2

Packet input No.	LM write addr.
1	c

LM (write)

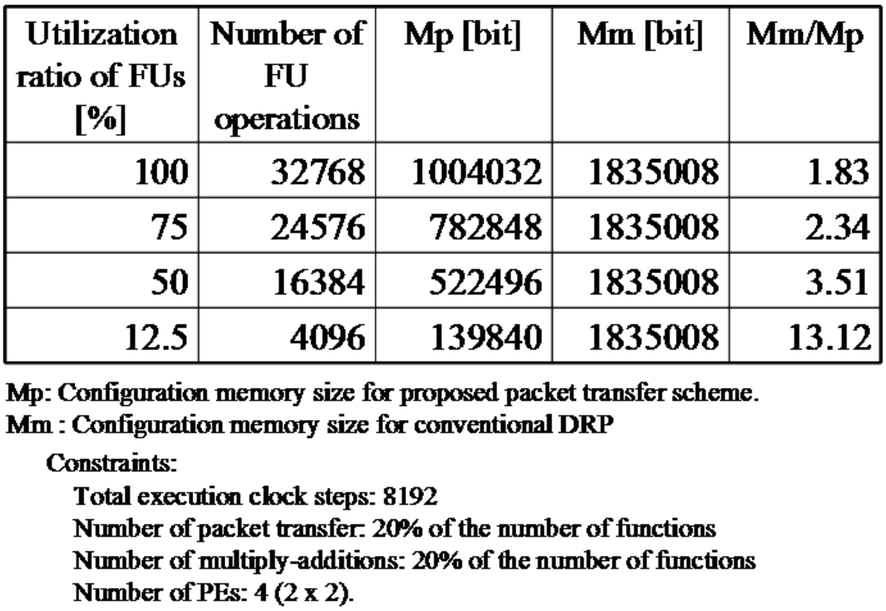
Figure 12. Distributed configuration memory for the packet data transfer scheme

5.2 Evaluation

The configuration memory size is evaluated in a DRP based on the register-transfer-level packet data transfer scheme. As a result, great reduction of the configuration memory size is achieved in comparison with the conventional DRP.

Let us compare the configuration memory size of the DRP based on the proposed packet data transfer scheme with that of the conventional DRP.

Figure 18 gives the configuration memory sizes for various utilization ratios of the FUs under the condition that every processing time is set to 8192 clock cycles. The configuration memory size Mp of the proposed DRP is reduced if the utilization ratio of the FUs becomes smaller.



Utilization ratio of FUs [%]	Number of FU operations	Mp [bit]	Mm [bit]	Mm/Mp
100	32768	1004032	1835008	1.83
75	24576	782848	1835008	2.34
50	16384	522496	1835008	3.51
12.5	4096	139840	1835008	13.12

Mp: Configuration memory size for proposed packet transfer scheme.
Mm : Configuration memory size for conventional DRP

Constraints:
Total execution clock steps: 8192
Number of packet transfer: 20% of the number of functions
Number of multiply-additions: 20% of the number of functions
Number of PEs: 4 (2 x 2).

Figure 13. Comparison of configuration memory size

On the other hand, the configuration memory size Mm of the conventional DRP is constant irrespective of the utilization ratio of the FUs. For an example, Mm becomes about 13 times larger than Mp in the case of the utilization ratio of the FUs is 12.5%.

The results presented in this section are relevant to the algorithm selection platform because it shows various situations of FU usage. The FUs are the elements that will be reprogrammed and thus will contain the implementation of the various algorithms. The results demonstrate that when the FU utilization ratio is high then a special purpose hardware circuit using pipelines are most appropriate. However when the FUs utilization is ratio is low the application has probably a high complexity of dataflow with many conditional branches. The proposed architecture is thus well suited to such cases where the algorithms are different because the distributed DRP have a very small configuration memory and thus several algorithms can be implemented easily and at a lower cost of resources.

6. CONCLUSION

In this paper we introduced a VLSI platform for the Algorithm Selection based Universal robotic Platform. On one hand we see that the algorithm Selection is a computationally intensive approach that provides robust results in the real-world data processing. On the other hand, the hardware based on the multi-context architecture of reconfigurable VLSI and improved by the advanced packet control makes a suitable candidate for the universal robotic platform.

The timing of a packet send can be programmable in the packet generator, and register-transfer-driven packet send control can be done. That principle makes the configuration memory size in the proposed packet data transfer scheme proportional to the number of packets required for register transfer control. On the other hand, the configuration memory size in the conventional dynamic reconfiguration scheme is proportional to the number of dynamic reconfiguration steps.

The compact router can be constructed based on the semi-autonomous collision-free packet routing scheme, which will be useful for fine-grain packet data transfer, because a large number of the routers must be provided in the ultra-highly parallel processing.

The concept of reduction of the configuration memory size is also useful for fine-grain interconnection network routing such as crossbar switch control. To utilize the advantage, we introduce the architecture of dynamic reconfiguration of LMs, where the number of MUs for each PE can be changed according to the data storage requirement. By the dynamic reconfiguration of LMs, the hardware resource of memories can be sufficiently utilized for different processing applications. Thus, the larger number of PEs in the proposed routing network architecture can be put in a chip with the same size, and the parallel processing capability can be enhanced greatly.

As a future extension of this work, the algorithm selection requires improvements in the quality of the algorithm selection. Moreover, it is important to develop the packet generation mechanism such as an address generation unit which generates a series of packets with regular destination addresses for further reduction of the configuration memory size.

REFERENCES

Amano, H., 2006, *A survey on dynamically reconfigurable processors*, IEICE Trans. Commun., Vol. E89-B, pp. 3179-3187

Tang, X., Aalsma, M., Jou, R., 2000, *A compiler directed approach to hiding configuration latency in chameleon processors,* In: In Proc. FPL (LNCS 1896), pp. 29-38

Sugawara, T., Ide, K., Sato, T., 2004, *Dynamically reconfigurable processor implemented with ipflex's dapdna*

Technology, IEICE Trans. Inf. & Syst, Vol. E87-D, pp. 1997-2003.

Kodama, T., Tsunoda, T., Takada, M., Tanaka, H., Akita, Y., Sato, M., Ito, M., 2006,

Flexible engine: A dynamic reconfigurable accelerator with high

performance and low power consumption, In: In Proc. COOL Chips IX, pp. 393-408

Saito, M., Fujisawa, H.and~Ujiie, N., Yoshizawa, H., 2005, *Cluster architecture for reconfigurable signal processing engine for wireless communication,* In: In Proc. FPL, pp. 353-359

Motomura, M., 2002,*A dynamically reconfigurable processor architecture,* In: Microprocessor Forum.

Levine, B., 2005, *Kilocore: Scalable, high-performance, and power efficient coarse-grained reconfigurable fabrics,* In: In Proc. Int. Symp. on Advanced Reconfigurable Systems, pp. 129-158

Veredas, F.J., Scheppler, M., Moffat, W., Mei, B., 2005, *Custom implementation of the coarse-grained reconfigurable address architecture for multimedia purposes*, In: In Proc. FPL, pp. 106-111

Fujioka, Y., Kameyama, M., 2012, *Configuration memory size reduction of a dynamically reconfigurable processor based on a register-transfer-level packet data transfer scheme,* In: SoC Design Conference (ISOCC), 2012 International, pp. 235-238

Honma, Y., Kameyama, M., Fujioka, Y., Tomabechi, N., 2005, *VLSI architecture based on packet data transfer scheme and its application,* In: In Proc. of 2005 IEEE Int. Symp. on Circuits and Systems, pp. 1786-1789

Fujioka, Y., Tomabechi, N., Kameyama, M., 2006, *Register-transfer-level packet data transfer scheme for a highly parallel VLSI processor*, In: In Proc. of Int. Conf. on Computers and Devices for Communication, pp. 9-13

Rice, J., 1976, *The algorithm selection problem*, Advances in Computers, Vol. 15, pp. 65-118

Yong, X., Feng, D., Rongchun, Z., 2003, *Optimal selection of image segmentation algorithms based on performance prediction*, In: Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing (VIP2003).

Takemoto, S., Yokota, H., 2009, *Algorithm selection for intracellular image segmentation based on region similarity*, Ninth International Conference on Intelligent Systems Design and Applications.

Lukac, M., Tanizawa, R., Kameyama, M., 2012, *Machine learning based adaptive contour detection using algorithm selection and image splitting*, Interdisciplinary Information Sciences, Vol. 18, pp. 123-134

Lukac, M., Kameyama, M., Hiura, K., 2013, *Natural image understanding using algorithm selection and high level feedback*, In: SPIE Intelligent Robots and Computer Vision XXX: algorithms and Techniques.

Malik, J., Belongie, S., Shi, J., Leung, T., 1999, *Textons, contours and regions: Cue combination in image segmentation*, In: International Conference on Computer Vision.

Maire, M., Arbelaez, P., Fowlkes, C., Malik, J., 2008, *Using contours to detect and localize junctions in natural images*, In: Conference on vision and Pattern Recognition.

Shi, J., Malik, J., 2000, *Normalized cuts and image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, pp. 888-905

Donoser, M., Urschler, M., Bischof, H., 2009, *Saliency driven total variational segmentation*, In: Proceedings of International Conference on Computer Vision.

Carreira, J., Li, F., Sminchisescu, C., 2012, *Object recognition by sequential figure-ground ranking*, International Journal of Computer Vision, Vol. 98, pp. 243-262

Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D., 2010, *Object detection with discriminatively trained part based models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, pp. 1627-1645