ON THE SIMULATION AND DEVELOPMENT OF MASSIVE PARALLEL DIGITAL ARCHITECTURES FOR MARKOV RANDOM FIELDS

Stephan Stilkerich

EADS Corporate Research Center
Image and Signal Processing Groupe
81663 Munich, Germany
e-mail: stephan.stilkerich@eads.net

Rupert Reiger

EADS Corporate Research Center
Algorithm and Software Groupe
81663 Munich, Germany
e-mail: rupert.reiger@eads.net

ABSTRACT

Markov Random Field modeling is a powerful parallel processing paradigm which can appropriately deal with the huge amount of data in the domain of low-level image processing problems. This paper describes a novel combined Simulation and semiconductor-technology independent VLSI design environment for Markov Random Field based processing models and systems. The concepts of this novel combined Simulation- and VLSI Design-Environment are experimentally demonstrated and proved by simulation results and detailed chip-layouts of a special Markov Random Field, which simultaneously solves the image processing problem of noise removing, intensity-level preserving and intensity histogram based segmentation.

1. INTRODUCTION

With the influential paper of D. Geman and S. Geman from 1984 [2] with the title *"Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images"* a probabilistic approach to signal- and image-analysis has been popularized. This approach adopts the Bayesian paradigm and uses discrete Markov Random Fields (MRF) as *a priori* models. Since this paper from 1984 numerous publications have proven that the Bayesian approach to image-analysis provides a very general and powerful framework encompassing various problems concerning image-processing - especially in the domain of low-level image processing. But up to now no industrial-relevant digital MRF hardware realizations and MRF hardware development approaches have been reported. This holds true for MRF hardware realizations because of their oversimplified MRF-models respectively algorithms for tractability, hardware architecture inflexibility, improper system scalability, analogue technologies and overall processing speed; regarding MRF hardware development approaches - if so far addressed as a problem at all - by now only few academic and industrial-intern studies exist, typically exploring just one specific problem setting at a time. Until now an overall concept and approach to tackle the MRF development problem in general is still lacking.

2. MRF MODEL-CLASS

This approach formally adopts the Bayesian framework and states the signal- and image-processing problems under consideration with probabilities and Bayes law in the following way:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{1}$$

The probability $P(x|y)$ is called the *a posteriori* probability, $P(x)$ the *a priori* probability and $P(y|x)$ the *likelihood* probability. Where y denotes some observed data and x denotes one possible solution of the signal- or image-processing problem. As we are dealing with an ill-posed problem class, we need a regularization. In this Bayesian approach regularization is realized by means of the *a priori* probability and the *likelihood* probability in order to provide solutions with a probability measure. So we are looking for an estimated solution x with high probability; that is essentially to maximize the *a posteriori*(MAP) probability:

$$\arg\max_x P(x|y) = \arg\max_x P(y|x)P(x) \tag{2}$$

From above equations it becomes obvious that this approach requires the modeling and representation of the probabilities $P(x)$ and $P(y|x)$ in a mathematical exact, flexible and tractable way as well as optimization algorithms to solve the MAP (maximum a posteriori) formulation. With the help of Markov Random Field theory, the properties of Gibbs distributions and the so called Hammersley-Clifford Theorem [3, which establishes the link between Markov Random Fields and Gibbs distributions, we can model, represent and calculate the two probabilities above mentioned by a sum of local interactions of the Sites with each other. We are especially interested in low-level image processing problems and therefore may restrict ourselves to focusing on site interactions in neighborhoods up to the 5th order (Figure 1).

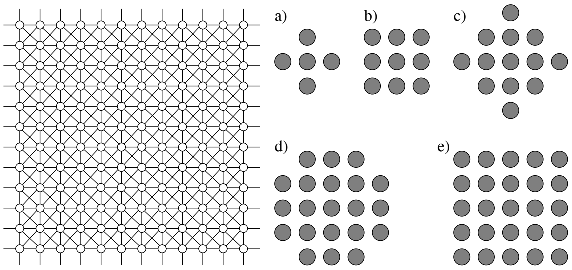


Fig. 1. Markov Random Field on a regular lattice. MRF neighborhood system of 1.-5. order (a-e).

The interested reader may consult the literature [7] for an in-depth presentation, proofs and discussion of this topic. Thus we may represent the *a posteriori* probability by an energy-functional $U(x, y)$, which is essentially just the summing of local computations:

$$U(x, y) = U(y, x) + U(x) \tag{3}$$

with

$$U(x) = \sum_{C_1} V_1(C_1) + \sum_{C_2} V_2(C_2) + \sum_{C_3} V_3(C_3) + \tag{4}$$

$$\sum_{C_4} V_4(C_4) + \sum_{C_5} V_5(C_5) + ... \tag{5}$$

and

$$U(y, x) = \sum_{s} F((x_s, y_s)) \tag{6}$$

C_i representing the cliques of the ith neighborhood system (Figure 1) and V_i respectively F denoting any function. s is a finite index set on the MRF-lattice.

The energy function $U(x, y)$ is usually non-convex and therefore we need relaxation methods to reach an optimum. The first group comprises stochastic relaxation methods, which are based on Simulated Annealing (SA) [4]. These methods converge asymptotically towards the global optimum but have the drawback of exhaustive computation time. The second group consists of mixed-mode (stochastic and deterministic) methods, Graduated Non-Convexity, Mean Field Annealing (MFA) [7], Modified Metropolis Dynamic (MMD) and the last group contains deterministic methods for instance Iterated Condition Modes (ICM) [1]. The methods forming the last two groups require significantly less computation time.

3. MARKOV RANDOM FIELD SIMULATION- AND DESIGN-ENVIRONMENT

Each Markov Random Field can be divided in the following four main architectural building blocks, which are essential and generic for each Markov Random Field: (1) Empty MRF-Cell with neighborhood connections. (2) MRF Memory Hierarchy to distribute and collect the data in the topology. (3) Energy Functional $U(x, y)$. (4) Optimization methods. All of the main architectural building blocks of Markov Random Fields earlier mentioned are mirrored in the Simulation-Environment as well as in the Design-Environment. The four main architectural building blocks may automatically be generated for simulation and VLSI implementation purposes.

3.1. MRF Simulation System

During then conception of our Simulation-Environment we had to keep several essential requirements in mind, so that our Simulation-Environment is extendable with respect to MRF-net-topologies, neighborhood-systems, Energy Functions, optimization algorithms and multi-scale processing approaches. The overall system also has to be flexible enough to cover different abstraction-levels of modeling and simulation. The spectrum reaches from pure serial simulation with float point arithmetic to pure massive parallel hardware-relevant simulation with bit-length limited fixed-point arithmetic. The Simulation-Environment (Figure 2) comprises the following eight main building blocks: (1) Simulation

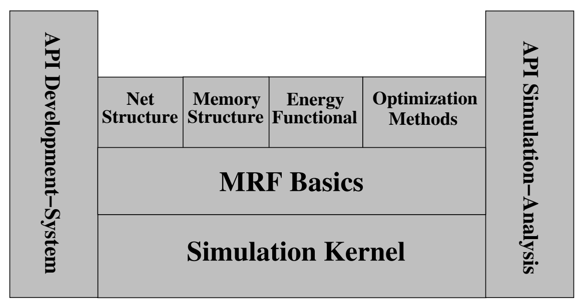


Fig. 2. Structure and components of MRF Simulation-Environment.

Kernel. (2) MRF Basics. Elementary infrastructure to support the Simulation of Markov Random Field based systems. (3) Net Structure. Building block to automatically set-up/initialize and generate the Net-Structure within the Simulation-Environment. (4) Memory Structure. Building block to automatically set-up/initialize and generate the Memory-Structure within the Simulation-Environment. (5) Energy Functional. Library of already modeled Markov Random Fields and reusable parts of different energy functionals. (6) Optimization Methods. Library of optimization algorithms ranging from pure deterministic methods to pure stochastic methods and methods in between. (7) API Simulation Analysis. Allows all kind of analysis: From visualization, data collecting and transfer to other analysis tools. (8) API Development Environment. Transfer of required data to the Development-Environment.

3.2. MRF VLSI Design-Environment

The design and VLSI-Implementation - as System-on-Chip realization - of massive parallel Markov Random Field based signal- and image-processing systems is a challenging, time consuming, error prone and expensive task which until now is not addressed by any approach of the academic and industrial community. In order to set up a MRF Design- Environment we have developed our own high-level design tools to support the critical parts of the design flow. Our tool-set is smoothly integrated in standard industrial FPGA/ASIC design flows by means of two interfaces: A specification language front-end interface for Markov Random Field definition and a HDL & Scripting back-end interface to standard 3rd party HDL-simulation, synthesis and place & route tools. The generators for the main building blocks of each Markov Random Field are the essential parts of our tool-set.

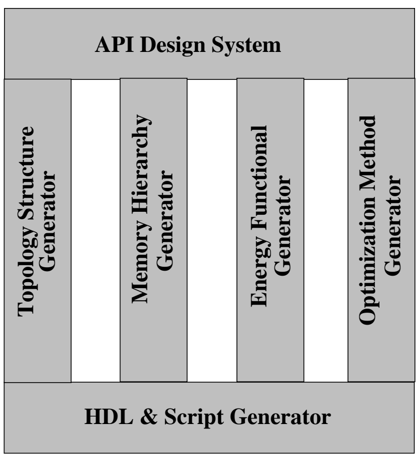


Fig. 3. Structure and components of MRF Design-Environment.

The Design-Environment (Figure 3) consists of the following six main building blocks: 1) API Design Environment. Specification language front-end to pass over the specification of the Markov Random Field to the Design-Environment. 2) Topology Structure Generator. Mapping of specified MRF topology to an internal abstract graph representation, which can be modified by structural reorganization methods. The topology generator will support MRF neighborhood system up to the 5th order (Figure 1). 3) Memory Hierarchy Generator. Builds up an internal graph representation, which can be modified by structural driven methods 4) Energy Functional Generator. Gets the required information from the AI and builds up an internal graph representation of the energy functional. Structural and technology driven reorganization methods generate the hardware architecture. Currently a High-Level MatLab language design entry and 3rd party tool support is used. 5) Optimization Methods Generator. What is described in the previous item also applies to this generator. 6) HDL & Script Generator. The internal representations of the different building blocks are processed and VHDL-Code and Scripts for 3rd party tools are written out. For an detailed presentation of the VLSI design environment please see [6]

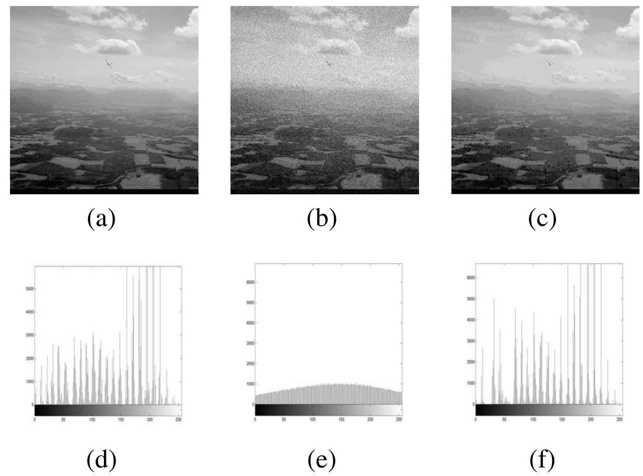


Fig. 4. Images and Histograms. (a)-(c) Original image, noisy image, restored image. (d)-(f) intensity histograms of above images.

4. RESULTS

The novel combined Simulation and design environment for massive parallel Markov Random Field VLSI architectures was intensively and successfully tested over the previous months. The following energy functional was used for the experimental investigations and results presented in this paper. For the simulations the Gibbs Sampler optimization method [2] was used and for the VLSI architectures the deterministic optimization method ICM [1].

$$U(x,y) = \underbrace{\sum_{s} \sum_{i=1,2,3,4} \sum_{<s,t>_i} \beta_i \left(1 - \frac{2}{1 + \frac{|x_s - x_t|}{\kappa}}\right)}_{\text{Intensity level preserving energy term}} + \quad (7)$$

$$\underbrace{\sum_{i=1,2,3,4} \sum_{<s,t>_i} V(\omega_s, \omega_t)}_{\text{segmentation energy term}} + \quad (8)$$

$$\underbrace{\frac{1}{2\alpha^2} \sum (x_s - y_s)^2}_{\text{white Gaussian noise energy term}} \quad (9)$$

where s denotes the finite index set of the MRF lattice, β_i are variables of the form 2^n, the same holds for α and by $< s, t >$ the neighbors are denoted; in this example the direct four neighbors of the 1. order neighborhood system (Figure 1). $V(\omega_s, \omega_t)$ is equal to the parameter $-\varpi$ if $\omega_s = \omega_t$ and otherwise to $+\varpi$. Large values of ϖ will support the generation of homogeneous regions.

(a) (b)

Fig. 5. (a) Image with added white Gaussian noise (Variance: 0.01, Mean:0.0). (b) Restored image

Thus we have a energy functional at hand, which simultaneously removes noise, preserves intensity levels and segments the image. Figure 5a shows the input image for the Markov Random Field with the before introduced energy functional. The image data was degraded with white Gaussian noise. The MRF removes the noise and results in Figure 5b, which is suitable for intensity histogram based segmentation. In Figure 4 the original image, the noise degraded image and the restored image as well as their corresponding histograms are illustrated. The result of the MRF segmentation with 8 classes is shown in Figure 6. The result of the segmentation begins to converge to the final result as soon as the noise removing process has converged to its final result. A prototypical implementation, the floorplan and the place & route result of a Markov Random Field with size 64x64 is illustrated in Figure 7.

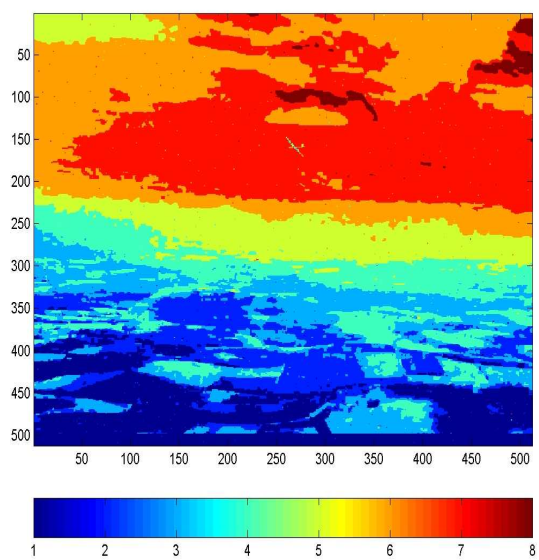
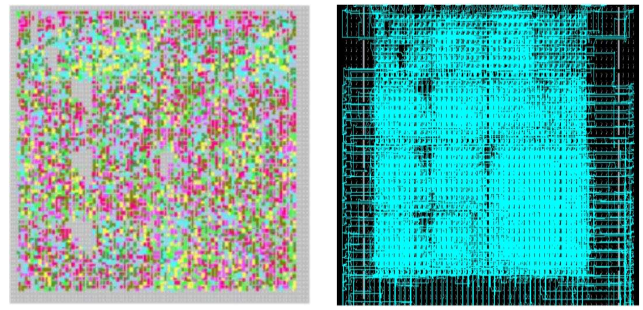


Fig. 6. Segmentation with 8 classes.



(a) Floorplan of MRF (b) Place & Route of MRF

Fig. 7. Markov Random Field realization. Size 64x64 and 1. order neighborhood system. Semiconductor technology FPGA Xilinx Spartan3.

5. REFERENCES

[1] J Besag, On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, Series B, 48 (3):259-302, 1986 (with discussion)

[2] S. Geman and D. Geman, Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6(1):721-741, 1984

[3] J.M. Hammersley and P. Clifford, Markov field on finite graphs and lattices, 1971 (unpublished)

[4] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimisation by simulated annealing. *Science*, 220:671-680, 1983

[5] A. Moini, Vision Chips or seeing silicon. A. Moini, 1998

[6] S. Stilkerich and J.K. Anlauf, High-Level Design Environment for massive parallel VLSI-Implementations of statistical signal- and image processing models. *Proceedings ISCAS 2004, Vancouver, Canada.*

[7] G. Winkler, Image Analysis, Random Fields and Dynamic Monte Carlo Methods. Number 27 in Applications of Mathematics. *Springer*, 1995.