

# Spiking Neural Networks for Real-Time Infrared Images Processing in Thermo Vision Systems

Snejana Pleshkova  
Department of Telecommunications  
Technical University  
Kliment Ohridski, 8 Sofia  
aabbv@tu-sofia.bg

**Abstract:** - Thermo vision are used in military, police custom traffic control, industrial and other specific applications for collecting and processing thermo visual information from infrared images. There is a problem in the steps of implementation of the developed methods and algorithms for infrared image processing in real time practical applications of thermo vision systems. Here is proposed to exploit the advances in powerful parallel computer graphics and image processing for computer vision and computer games applications, where are developed graphical processing unit (GPU) and Compute Unified Device Architecture (CUDA) with the ability of parallel processing and the high-speed memory access of graphical processing units (GPU), which is essential in the real time applications with neural networks in most of the infrared image processing applications.

**Key-Words:** - *spiking neural networks; real time infrared image processing; thermo vision systems*

## 1 Introduction

Thermo vision are used in military, police custom traffic control, industrial and other specific applications for collecting and processing thermo visual information from infrared images [1, 9]. There are many hardware or software development tools for testing the methods and application algorithms for infrared captured image processing in thermo vision systems [2, 3, 10]. The problems arise in the steps of implementation of the developed methods and algorithms in real time practical applications of thermo vision systems. In surveillance and security thermo visual systems one of the most practical goal is the moving objects detection and tracking in infrared images captured from a thermo vision camera. The input infrared images are usually separated and processed in small blocks with an appropriate and chosen shape (for example rectangular) and size (for example 8x8). In conventional hardware or software implementation of infrared image processing algorithms the blocks are processed consecutively or in series and the achieving the real time processing is not always possible.

The advances in powerful parallel computer graphics and image processing for computer vision and computer games applications with the developed graphical processing unit (GPU) and Compute Unified Device Architecture (CUDA) [4] offers for GPU-based computing a powerful development framework integrated with high level parallel programming languages like C or C++ languages. Graphical

processing units (GPU) are devices designed to exploit parallel shared memory-based floating-point computation. They provide memory access speeds superior to those of commodity CPU-based systems. These features to update in parallel the model variables every iteration compared to other solutions like programmable logic, integrated circuits, custom shared memory solutions, and cluster message passing computing systems make GPUs attractive in real time image processing and especially in this article for infrared image processing applications.

Here is proposed to exploit the ability of parallel processing and the high-speed memory access of graphical processing units (GPU), which is essential in the real time applications with neural networks in most of the infrared image processing applications.

In most applications of infrared image processing with neural networks the processed algorithms work sequentially by a CPU, which means only one neuron is updated at a given time. As a result the performance degrades quickly with the increase in network size and connectivity. This is especially the case for large connectivity, since sequential processors need to iterate over every connection for each neuron. To speed up the operation, supercomputers or distributed computers are normally used for large-scale neural network simulation. But these solutions incur high cost. Traditional CPU architectures are not designed for parallel processing.

To avoid this problem in real time infrared image processing applications a suitable type of neural network is proposed to use the spiking neural network (SNN)

implemented in graphical processing unit (GPU) and Compute Unified Device Architecture (CUDA).

The example is presented for real time infrared image processing applications like moving objects detection and tracking in infrared images in surveillance and security thermo visual systems.

## 2 Spiking Neural Networks performance useful for infrared image processing

A spiking neural network (SNN) is a model of a biological neural network with a simplified process of synaptic transmission and neurons communication with each other by spikes, modeled as time-stamped potential pulses.

The accuracy of a spike time depends on the choice of numerical integration systems, which can be classified into the following categories:

- clock-driven (synchronous) systems evaluate model variables only at fixed points in time in which the resolution of the time grid, defined by the magnitude of a time step, determines the simulation accuracy and affects the execution time;
- event-driven (asynchronous) systems update variables only at the exact time of a spike event exact time, in which the accuracy of the event time in these systems is not tied to a precision of any time grid, but depends on floating-point format chosen (double or single precision);
- hybrid systems combine advantages of event-driven and clock-driven systems, in which the refresh of the model variables is at fixed points in time, but yet they process events at the exact time.

Two identical spiking neural network (SNN) excited with identical stimuli, but implemented as a clock- and event- driven systems do not produce the same spiking pattern unless a time step in the clock-driven implementation is small enough to achieve the designed accuracy.

## 3 Implementation of Parcker-Sochcki Integration method in real time infrared image processing

The analysis of the above mentioned choices of numerical integration systems leads to the proposition to use here for infrared image processing the Parker-Sochacki (PS) numerical integration method [5] to the biologically plausible phenomenological neuron model

developed by Izhikevich [6]. This integration method provides accuracy appropriate for simulation of spiking neural network (SNN) with biological mechanisms requiring exact event timing and achieving full double-precision integration accuracy.

The Parker-Sochacki (PS) numerical integration technique is based on application of the Maclaurin series to a solution of differential equations with an initial value problem (IVP),

$$y(t) = \frac{dy}{dt} = f(t, y(t)), y(t_0) = y_0, t \in [t_0 - \alpha, t_0 + \alpha]. \quad (1)$$

The method was developed based on the Picard iteration [12] under the assumption that the solution function is locally Lipschitz continuous in  $y$  and continuous in  $t$  (Picard –Lindelof theorem) [7], and therefore can be described with power series. Consequently, based on the fact that next coefficient in the series can be represented with the derivative of previous coefficient,

$$\sum_{p=0}^{\infty} (p+1) y_{p+1} t^p = \sum_{p=0}^{\infty} y_p' t^p, \quad y_p = \frac{y^{(p)}(0)}{p!}. \quad (2)$$

and after substituting (2) in (1) the IVP (1) can be described in terms of power series:

$$\sum_{p=0}^{\infty} (p+1) y_{p+1} t^p = f\left(t, \sum_{p=0}^{\infty} y_p' t^p\right) \quad (3)$$

Provided that  $f$  is a linear function,  $f(t, y(t)) = ky(t) + b$ , Eq. (3) becomes (constant term is temporary dropped):

$$\sum_{p=0}^{\infty} (p+1) y_{p+1} t^p = k \left( \sum_{p=0}^{\infty} y_p' t^p \right) \quad (4)$$

The equation (4) exhibit loop level parallelism (LLP) and parallel reduction, which can be exploited if all coefficients are pre-calculated.

However, provided that  $f$  is a quadratic function,  $f(t, y(t)) = ay^2(t) + by(t) + c$ , after series multiplication, equation (3) becomes:

$$\sum_{p=0}^{\infty} (p+1) y_{p+1} t^p = a \sum_{p=0}^{\infty} \left( \sum_{p=0}^{\infty} y_i y_{p-i}' \right) t^p + b \sum_{p=0}^{\infty} y_p' t^p \quad (5)$$

Exploiting parallel computation is problematic in this case because of linearly scaled convolution, which introduces loop-carried circular dependence. Partial parallelism still can be exploited in the convolution itself and term  $by_p/(p+1)$ .

#### 4 The spiking neural networks for real time infrared image processing with computer unified device architecture (CUDA)

The equation (4) and (5) shows two important possibilities to use full parallelism in parallel reduction of all pre-calculated coefficients or partial parallelism in convolution, respectively. This assertion is very important in real time application of infrared image processing and is well suited with the advances graphical processing unit (GPU) and Compute Unified Device Architecture (CUDA). Therefore, in this article is presented the structure of a real time infrared image processing with spiking neural network and compute unified device architecture (CUDA), shown in Fig.1.

The Infrared Image Capture in real-time the thermal images to be processing. The type of this infrared sensor is EasIR-9, which is a standard thermo vision camera. The captured infrared images are transformed as Pixel Data to the Spike Convertor. The function of this block is to convert the each value of input Pixel Data of the infrared images to corresponding amplitudes, and time spacing of the pulse sequence (Spikes), representing the inputs of the used spike neural network (SNN) for infrared image processing. The Spikes are inputs of the used necessary computer Unified devices architecture interface (CUDA Interface). This interface distributes the Spikes to the blocks SP, which in CUDA architecture are named as Scalar Processor (SP). The block SP in CUDA architecture are arranged as Grid of Blocks named Streaming Microprocessors (SM) with the corresponding Shared Memory and Local Memory. All of the existing in a CUDA architecture Grids of Blocks are connected to the Global Memory.

The control of the infrared image processing and applying of a chosen algorithm for spike neural network (SNN) is performed from a Digital Signal Processor (DSP) or from Host Computer. Therefore, in Fig.1 is shown a DSP or Host Computer Interface to the CUDA architecture block. Also a commonly used Display Interface connected to LCD Display is shown in Fig.1 for visualization of the input and processed infrared images.

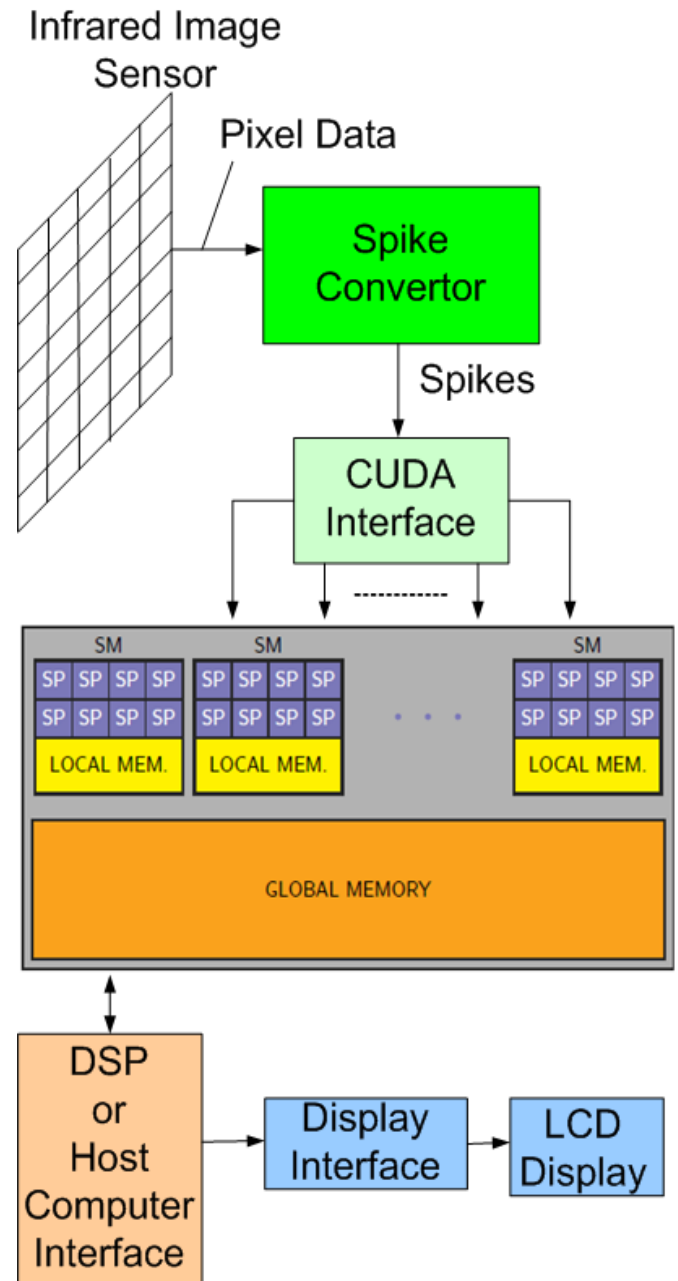


Figure1. Structure of a real time infrared image processing with spiking neural network and compute unified device architecture (CUDA)

A more detailed representation of the Grids of Blocks in CUDA architecture, which execute the spike neural network (SNN) algorithm for infrared image processing, is shown in Fig.2. It is seen that each part of the Grid Block can be regarded as an  $n \times n$  array of sub blocks, named as Thread (1,1) ... Thread (n,n). The names Thread ( , ) are chosen from the terminology of CUDA Programming Model using Open CL programming language [8].

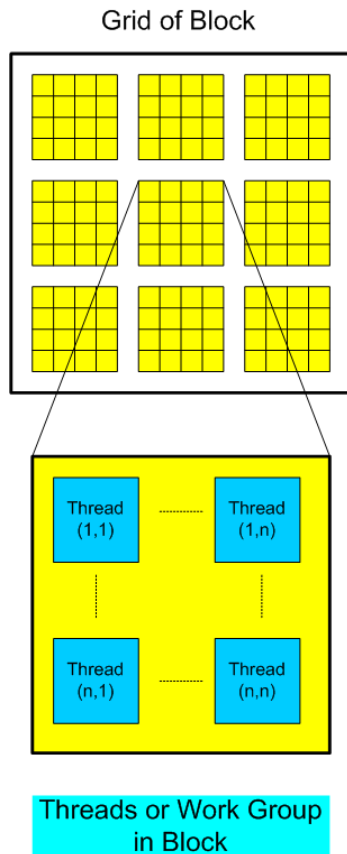


Figure.2. Detailed representation of the Grids of Blocks in CUDA architecture, which execute the spike neural network (SNN) algorithm for infrared image processing

The detailed structure of the Threads ( , ) is shown in Fig.3. each Thread is connected to Shared/Local Memory an indirect to the Private Memory. These types of memories are for storing and updating the local spike signal, coefficients and local executed infrared image processing operations corresponding to the spike neural network (SNN) algorithm for real time infrared image processing in CUDA architecture.

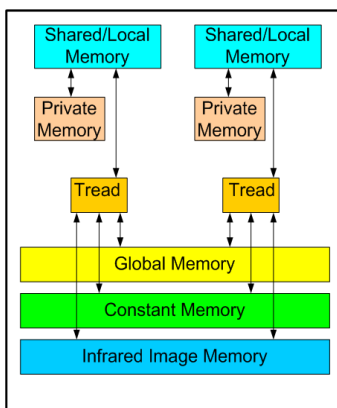


Figure.3. Detailed structure of the Threads

There are shown also in Fig.3 the necessary block Global Memory, Constant Memory and Infrared Image Memory, which are globally connected to all Thread blocks, transferring and distributing to these Thread blocks the global data, constant values and infrared image information as Spikes values.

## 5 Results and Conclusion

The experiments for real time infrared image processing and spike neural network (SNN) with CUDA architecture are carried out with NVIDIA GTX280 GPU card that consists of 240 scalar processors grouped into 30 Streaming Multiprocessors (SM), each operating at 1.2 GHz. The sustained performance of the GTX280 GPU card is approximately 350 GFLOPS. Each Streaming Multiprocessor (SM) has a hardware thread scheduler for spike neurons that selects a group of threads for execution. If any one of the spike neuron threads in the group issues a costly external memory operation, then the spike thread scheduler automatically switches to a new spike thread group.

At any instant of time, the hardware allows a very high number of spike threads, approximately 768 spike threads per Streaming Multiprocessors (SM) in GTX280, to be active simultaneously. By swapping spike thread groups, the spike thread scheduler can effectively hide costly memory latency. Each GTX 280 GPU contains a 512-bit DDR3 interface to the graphics display memory with a peak theoretical bandwidth of 143GB/s.

The comparison of the results achieved in the experiments for real time infrared image processing with spike neural network (SNN) and CUDA architecture implemented in NVIDIA GTX280 GPU card are made with the same algorithm for infrared image processing with spike neural network (SNN) using standard Pentium chipset with a 64-bit quad-pumped DDR3 interface. The results from this comparison are presented in Table I.

Table 1

Spike Neural Network (SNN) for Infrared Image Processing	In Programming Language	Speed of Execution Real Time ability
With CUDA Architecture and NVIDIA GTX 280 GPU	Open CL	350 GB/s Yes
With Standard Pentium Chipset and 64-bit quad-pumped DDR3 Interface	Microsoft Visual Studio 2010 and Open CV	28 GB/s No

In conclusion is possible to summarize the effectiveness of using graphical processing unit (GPU) and Compute Unified Device Architecture (CUDA) in spiking neural network for real time infrared images processing: parallelism, high memory access, high speed processing.

## Acknowledgements

This work was supported by National Ministry of Science and Education of Bulgaria under Contract DDVU 02/4-7: “Thermo Vision Methods and Recourses in Information Systems for Customs Control and Combating Terrorism Aimed at Detecting and Tracking Objects and People”.

## References:

- [1] Lebold J. Infrared Thermography and Distribution System Maintenance Electricity Today, Volume 3. 2008, 18-19.
- [2] FLIR Application Book. FLIR Company 2010
- [3] Coon D. D. and Perera A.G. U. Spectral information coding by Infraredphotoreceptors. International Journal of Infrared and Milimeter Waves, Volume7, Number 10 1571-1583.
- [4] NVIDIA CUDA. <http://developer.nvidia.com/>
- [5] G. E. Parker and J. S. Sochacki. Implementing the Picard iteration, Neural, Parallel Sci. Comput., vol. 4, pp. 97-112, 1996
- [6] E. M. Izhikevich and G. M. Edelman. Large-scale model of mammalian thalamocortical systems, Proceedings of the National Academy of Sciences, vol. 105, pp. 3593-3598, 2008.
- [7] E. Picard, Traite D'Analyse. Gauthier-Villars, 1922-1928, vol. 3.
- [8] (2009, Jul.) NVIDIA CUDA C Programming Best Practices Guide. [Accessed online 04/30/2010]. <http://developer.nvidia.com/>
- [9] Andonova A., Thermographic evaluation of electro-mechanical relays'quality in railway automation, International Journal of Electricaland Computer Engineering (IJECE), Feb. 2012, vol.2, No1, 2012, pp.1-6,ISSN:2088-8708
- [10] Andonova A., S. Todorov, Buried Object Detection by Thermography,Annual Journal of Electronics, vol.4, № 1, Sofia, , pp. 133-136,2010,ISSN 1313-1842