A parallel CPU/GPU Algorithm for 3D Pose Estimation using CUDA and OpenMP

Kenia Picos[1], **Víctor H. Díaz-Ramírez**[1], **Antonio S. Montemayor**[2], **Juan J. Pantrigo**[2]
kpicos@citedi.mx, vhdiaz@ipn.mx, {antonio.sanz, juanjose.pantrigo}@urjc.es
[1]Instituto Politécnico Nacional (CITEDI-IPN), [2]Universidad Rey Juan Carlos *NVIDIA GPU Education Center*

Motivation

Pose recognition is characterized by location and orientation parameters, in which introduces high complexity due the huge number of visualization than a target can present within a scene. Several issues such as noise, incident light sources and geometrical distortions, modify the appearance of a target with respect an observation point, making object recognition a complex task. A design of an effective algorithm is needed in order to analyze the physical phenomena implied in the visualization of a moving 3D object, and to improve the execution performance.

3D Pose Estimation

The proposed system employs an adaptive bank of space variant correlation filters to solve 3D target pose recognition. Each generalized matching filter (Eq. 1) is constructed with a image template, which includes the parametric view of the target in a specified 3D pose.

$$H^*(\boldsymbol{\mu}) = \frac{T(\boldsymbol{\mu})+m_bW(\boldsymbol{\mu})+m_tW_t(\boldsymbol{\mu})}{\frac{1}{2\pi}|W(\boldsymbol{\mu})|^2 * N_b^0(\boldsymbol{\mu})+\frac{1}{2\pi}|W(\boldsymbol{\mu})|^2 * N_t^0(\boldsymbol{\mu})} \quad \text{Eq. (1)} \qquad DC = 1-\frac{|c^b|^2}{|c^t|^2} \quad \text{Eq. (2)}$$

The output correlation function is given by $c_i(\mathbf{x}) = IFT\{F(\boldsymbol{\mu})H_i^*(\boldsymbol{\mu})\}$. The best match is determined by the max value of DC (discrimination capability) obtained from the computed correlation planes (Eq. 2). The selection of the output estimation is given by the highest DC value obtained from the entire filter bank defined by $DC_{best} = \max\{DC_1, DC_2, \ldots, DC_M\}$.

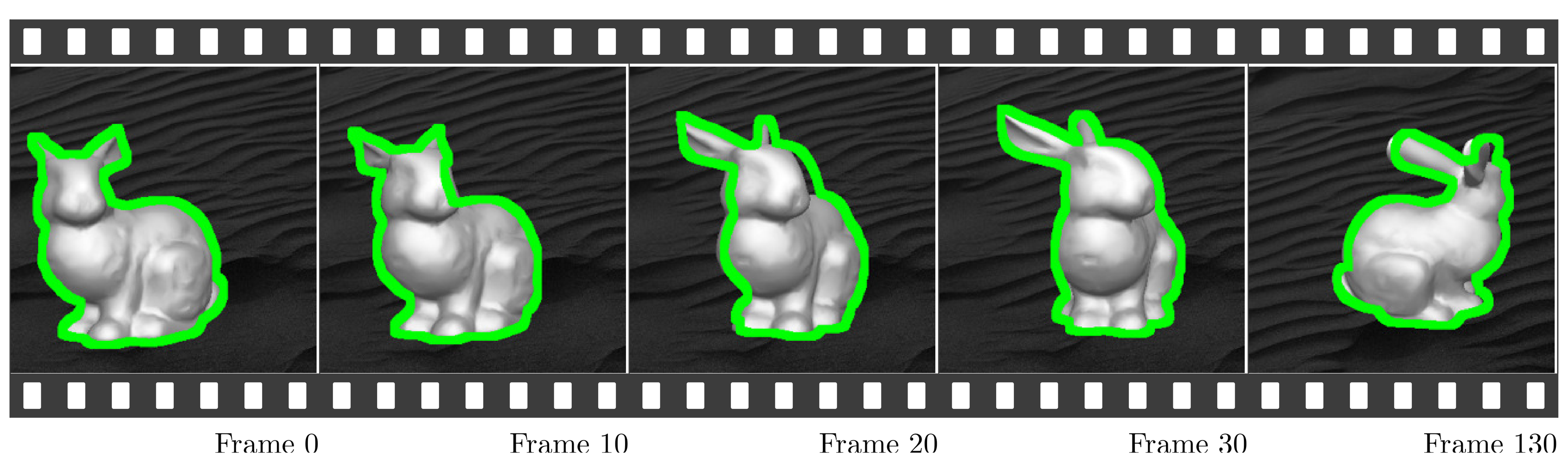

Figure 1: Pose estimation in a synthetic sequence.

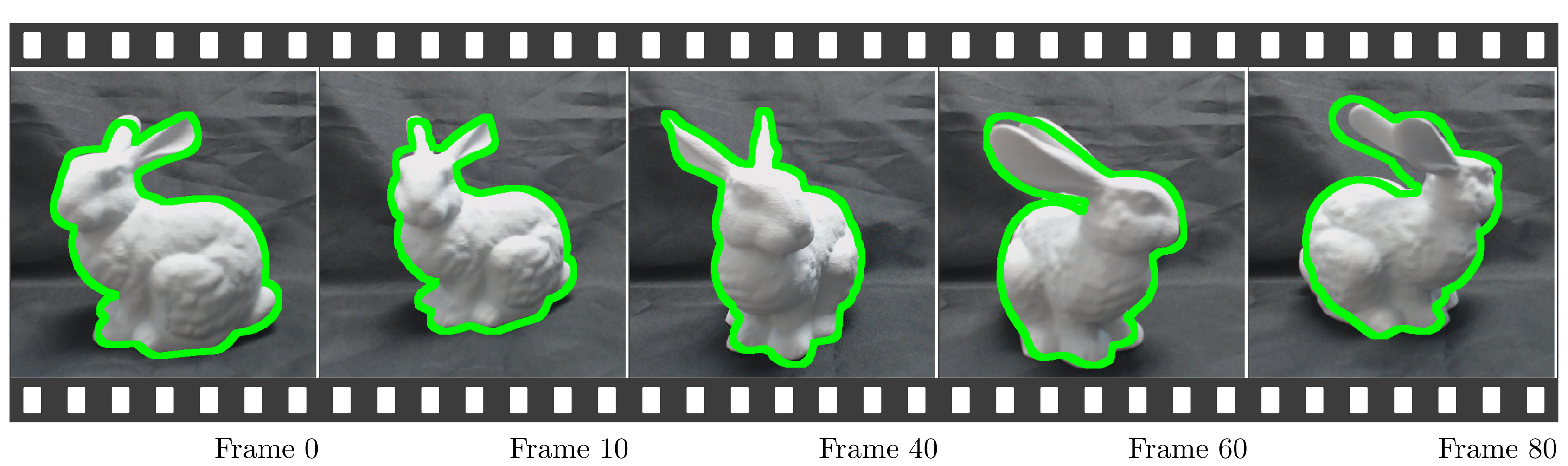

Figure 2: Pose estimation in a real video sequence using a 3D printed model.

The input frame is characterized by a target in an unknown pose, defined by location, orientation and scaling parameters of the target, and embedded into a disjoint background. The appearance of the target depends on pose and lighting properties of the surface material. For this, the image templates was rendered with the Phong lighting model including small highlight reflections using OpenGL. Figs. 1 and 2 show the pose estimation performance in synthetic and real input video, respectively, both with 300 frames of 512×512 pixels.

Results

The current proposal focuses on improving the execution of a 3D object recognition algorithm by using a combination of data and task parallelism, in order to take full advantage of a CPU/GPU architecture. The implementation has been developed in a CPU/GPU architecture running on a Linux OS with a multi-core host processor and NVIDIA graphics processor. The program was developed with CUDA CUFFT with OpenGL interoperability for 3D graphic model generation and display, and OpenMP for CPU thread parallelization.

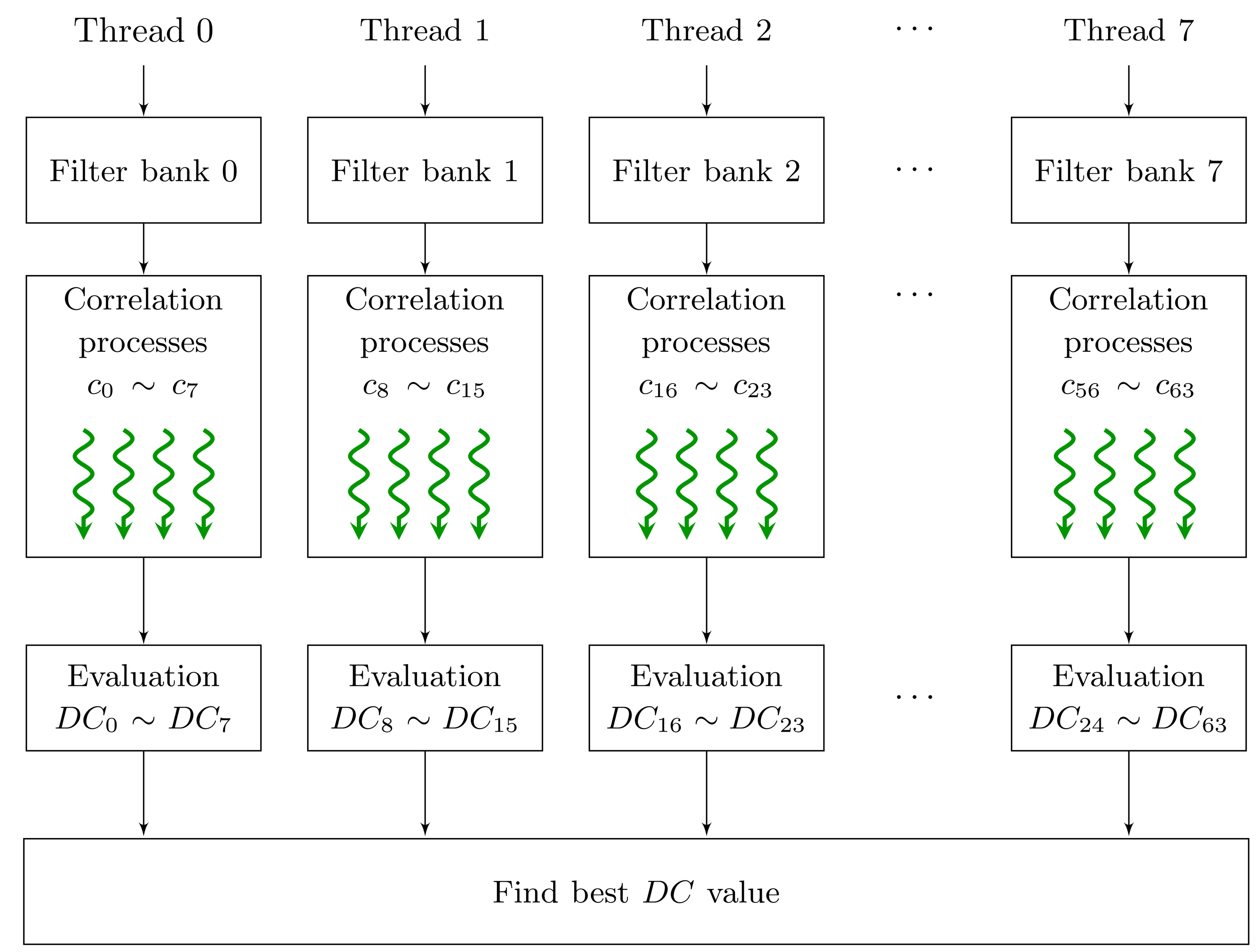

Figure 3: Parallel implementation using OpenMP and CUDA CUFFT.

The proposed algorithm computes a bank of M filters with N CPU threads. As we can see in Fig. 3, each thread executes a defined portion (M/N) of the bank. Then, several correlation processes are computed in a GPU kernel using CUFFT. Finally, the best match is found given the max DC value of the entire bank, given the best correspondence of location and orientation parameters of the target. The proposed parallel approach yields a time execution improvement using a filter bank up to 1000 filters processed in frequency domain.

Location error	Orientation error	Sequential evaluation	OpenMP+CUDA parallel evaluation	Speedup with 1000 filters
8.9±0.6 pixels	**5.6±0.4 degrees**	**200 s**	**68 s**	**3×**

Conclusions

This work presents a proposal for pose recognition with adaptive correlation filters to accurately estimate the location and orientation of a target within a 3D space. A parallel algorithm was designed in order to execute massive concurrent correlation processes in frequency domain. A CPU/GPU implementation using OpenMP and CUDA achieves an improvement of almost 3 times the sequential execution. Our future work will focus to optimize the performance of multi-pose object recognition for real-time applications.