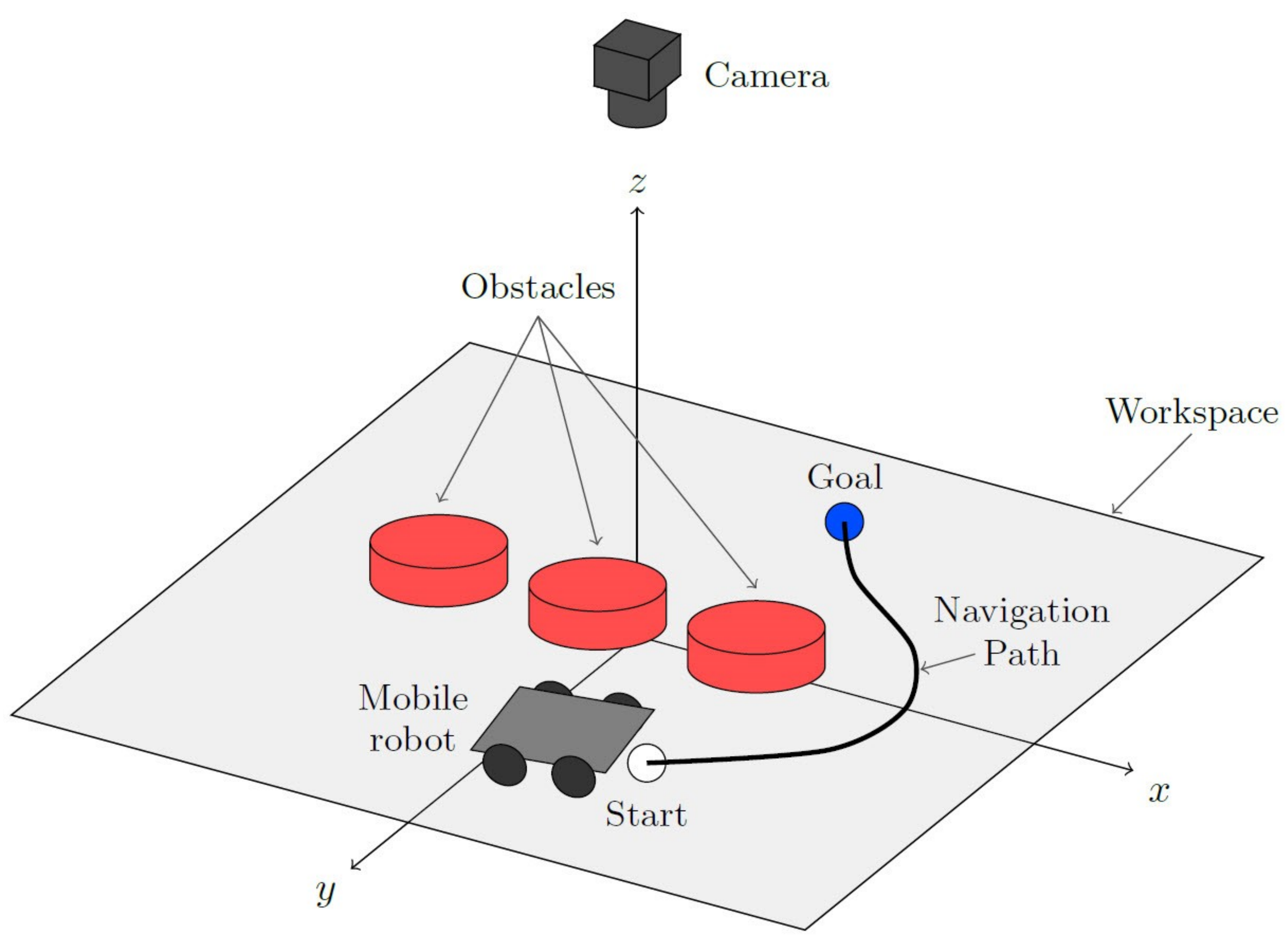
Parallel Path Evaluation for Mobile Robot Navigation

Ulises Orozco-Rosas[1,2], Kenia Picos[1,2], Antonio S. Montemayor[2], Juan J. Pantrigo[3], and Alfredo Cuesta-Infante[3]
{ulises.orozco, kenia.picos}@cetys.mx, {antonio.sanz, juanjose.pantrigo, alfredo.cuesta}@urjc.es
[1]**CETYS Universidad,** *Centro de Excelencia en Innovación y Diseño*, [2]**Universidad Rey Juan Carlos,** *NVIDIA GPU Education Center*

Introduction

This work presents an approach for parallel path evaluation for autonomous mobile robot navigation. To achieve the autonomous mobile robot navigation the system employs the integration of template-matching filters [1] for obstacle detection and evolutionary artificial potential field method [2] for path planning.

The recognition system employs a digital camera to sense the environment of the mobile robot, as it can be seen in the illustration below. The captured scene is processed by a bank of space variant filters in order to find the obstacles in the workspace. When the position and size of the obstacles are already known, the path planning system employs evolutionary artificial potential field (EAPF) to generate a feasible, smooth, and safe path for mobile robot navigation. The path generation and evaluation is performed employing parallel computing on GPU to reduce the amount of computation time.



Methodology

In order to achieve the autonomous mobile robot navigation, the system is composed by three basic subsystems: obstacle detection, path planning, and path tracking. Where the path generation and evaluation is the main objective in this work.

1. **Obstacle detection.** An input scene is given to the system, we assume that the scene contains a feasible workspace with obstacles. A filter is build and trained with a template reference. The cross correlation is performed between the input scene and the filter to obtain a correlation plane. The output correlation plane yields quantity levels of the best match. In this case, several peaks are produced in where the area of each obstacle coincides.
2. **Path planning.** The system employs EAPF to perform the path planning. The EAPF uses the start, goal, and obstacles positions as features to obtain a sequence of objective points (path) that the mobile robot must attain.
3. **Path tracking.** The system converts the objective points to rotate and advance motion commands to let the mobile robot moves from the start to the goal.

Parallel Path Evaluation

The flowchart (below) shows the parallel path generation and evaluation using the EAPF. The EAPF uses the start, goal and obstacles positions as an input, and it returns a set of objective points that conforms the path for the mobile robot navigation. The process starts with the creation of a random population *P(t)* of individuals (solutions), each individual is codified with the values of the proportional gains, attraction k_a and repulsion k_r required to generate the path. For the parallel evaluation on GPU, where each path is evaluated. First, the total potential field $U_{total}(q)$ computation is performed,

$$U_{total}(q) = \frac{1}{2}\left[k_a(q-q_f)^2 + k_r\left(\frac{1}{\rho}-\frac{1}{\rho_0}\right)^2\right]$$

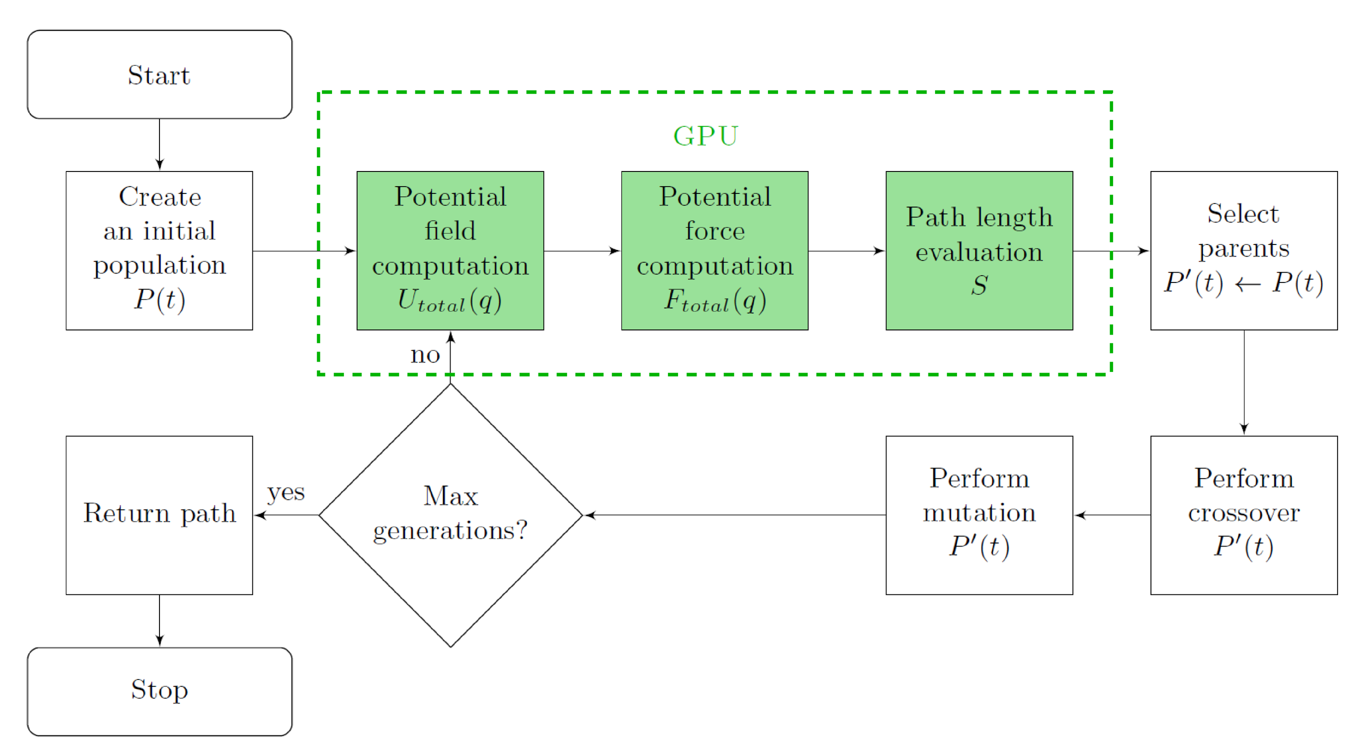
Then, the total potential force $F_{total}(q)$ which is used to drive the mobile robot is computed,

$$F_{total}(q) = -\nabla U_{total}(q)$$

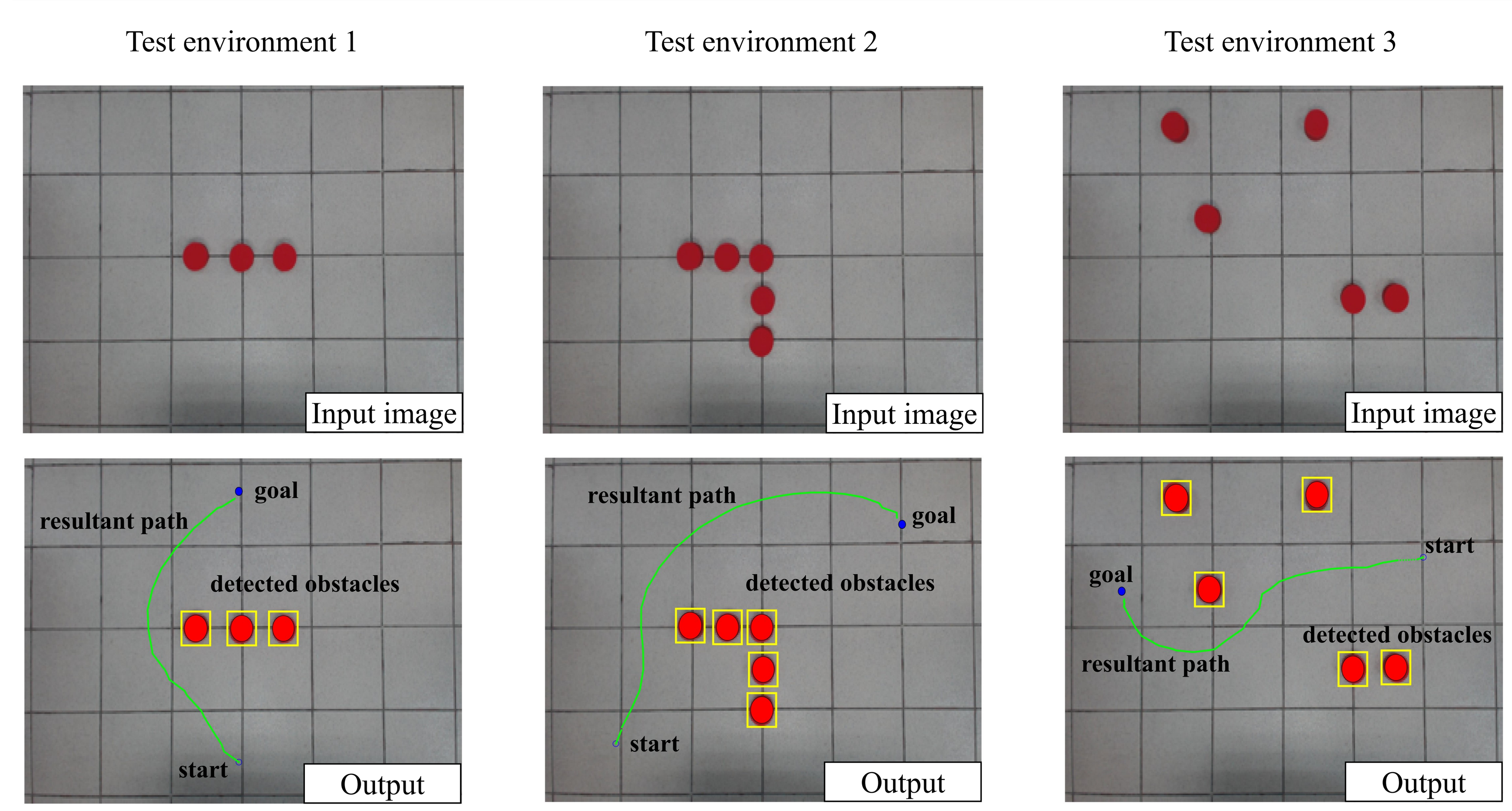
Last, for the parallel path evaluation on GPU, each path is evaluated using

$$S = \sum_{i=0}^{m} \|q_{i+1} - q_i\|$$

After the parallel path evaluation on GPU, the selection, crossover and mutation operators are applied to evolve the individuals in *P(t)*. All the path planning is enclosed in an iterative process until the maximum number of generations has been achieved.



Path Planning Results

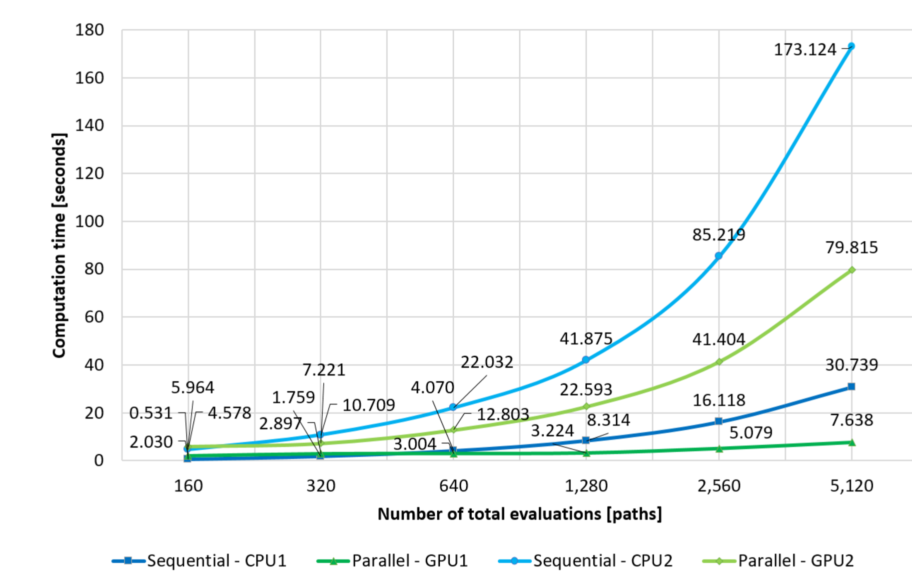


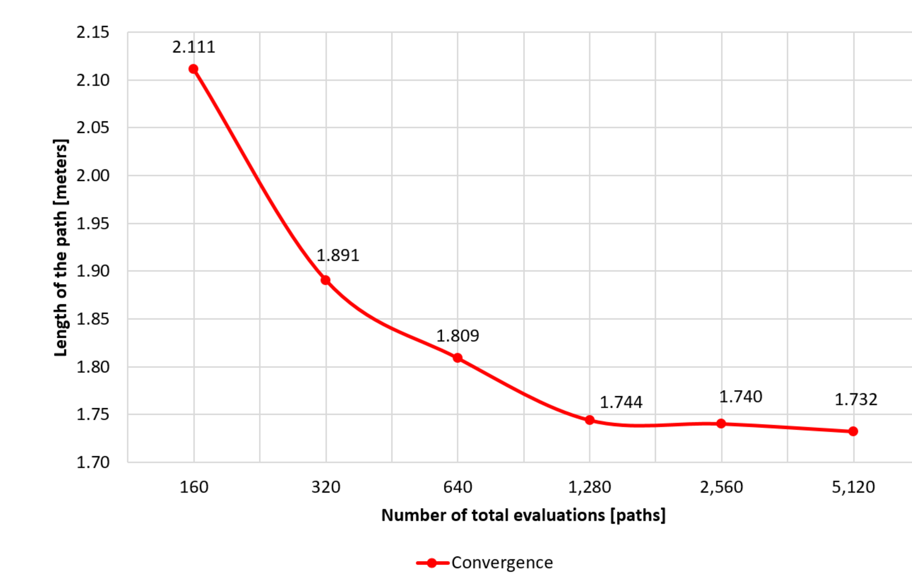
Performance Results

The experiments were achieved using two different computers:

- Computer 1: Ubuntu 16.04, CUDA 8.0, CPU1: Intel quad core i7-4710HQ, GPU1: GeForce GTX 860M with 640 CUDA cores.
- Computer 2: Ubuntu 14.04, CUDA 6.5, CPU2: quad core ARM Cortex-A15, GPU2: NVIDIA Kepler with 192 CUDA cores (Jetson TK1).

To evaluate the performance of the parallel path evaluation on GPU versus the sequential path evaluation on CPU, we carried out independently thirty tests for each number of total evaluations.





Conclusions

- In this work, we have presented the parallel path evaluation on GPU for mobile robot navigation using the EAPF programmed in C++/CUDA.
- The performance results show that the parallel path evaluation on GPU1 accelerates the evaluation process by a factor of 4.0x for the bigger population tested in comparison with sequential path evaluation on CPU1.
- The performance results show that the parallel path evaluation on GPU2 accelerates the evaluation process by a factor of 2.2x for the bigger population tested in comparison with sequential path evaluation on CPU2.
- We can see the advantage of using the parallel path evaluation on GPU, as well as we can see that this advantage applies for onboard computers like the Jetson TK1.

References & Acknowledgments

[1]Diaz-Ramirez, V.H., Picos, K., and Kober, V., "Target tracking in nonuniform illumination conditions using locally adaptive correlation filters," Opt. Commun. 323, 32-43 (2014).
[2]Montiel, O., Sepúlveda, R., and Orozco-Rosas, U., "Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field," Journal of Intelligent & Robotic Systems 79(2), 237-257 (2015).

We thank to the Mexican National Council of Science and Technology (CONACYT) for supporting our research activities.