Multiresolution and Local Search Methods for Optimizing Visual Tracking Processes on GPU

Raúl Cabido,* Antonio S. Montemayor, Juan José Pantrigo
Universidad Rey Juan Carlos (Madrid, SPAIN)

Bryson R. Payne
North Georgia College & State University (USA)

1 Introduction

Some visual applications, such as *virtual blackboards* or gesture recognition for HCI applications, rely on precise visual tracking algorithms. In order to provide this accuracy, some refinement methods are needed as a complement to a coarser approach. In this case, Local Search (LS) methods are typically used, as their purpose is the exploration of a search space to improve a previous estimation. In this category fall many search algorithms such as Tabu Search, Simulated Annealing, Scatter Search, etc.

We propose an extension to prior experiments with Particle Filters (PF) in visual tracking [Montemayor et al. 2006] to obtain more accurate tracking by including Multiresolution and Local Search stages. In particular, PF enables the modeling of a stochastic process with an arbitrary probability density function by approximating it numerically with a weighted set of samples called particles. Each particle, p_i, is composed of two variables, an estimated state of interest (position, size,...) and its weight, p_i=($\mathbf{x}_i$,π_i). At each time step, all the particles are evaluated and properly weighted. Then, the best one of the set is considered to be the estimation of the state, or tracking location. However, as a probabilistic framework, the process lacks a high degree of pixel accuracy. If we perform a local search over the best candidate, we can approximate more accurately a true state.

Particle filter techniques are ideal candidates for execution on the GPU, as their main algorithm is based on multiple independent plausible interpretations that are propagated over time. Their independency leads to a very efficient mapping to modern GPU architecture. The choice to perform the refinement steps on the GPU was more challenging, as there are many methods, and most of them are based on non-deterministic and non-independent building blocks.

2 Method Overview

First, PF returns the position (x, y) of a square Region of Interest (ROI) where the target seems to be located (see Fig. 1.a, 1.b, 1.c, 1.d, and please refer to [Montemayor et al. 2006] for deeper explanations). Then, given the size of that ROI, we apply a multiresolution approach and a Local Search procedure on each Level Of Detail (LOD) (Fig. 1.e, 1.f). The multiresolution approach is performed in order to return an appropriate LOD (larger or smaller) if the trackable object does not fit in the first approximation. Then, LS adjusts a rectangular window inside the best LOD returned by the multiresolution procedure. These rectangular windows can be created sequentially or in a random fashion.

We have tested our two versions of the optimized PF against the straightforward one. Visual tests indicate that the optimized versions produce qualitatively better results, as they avoid some inherent artifacts of a probabilistic estimation that can result in shaky tracking. To this end, we have tested over a static target where the optimized PF was more stable (up to 70% lower standard deviation for some configurations). Also the trackable region is adapted at every time step to the detected object, improving the overall quality of the tracking. The optimized and the naive implementations can run in real time for many proposed configurations. Our experiments, tested on an Nvidia GeForce 7800GTX GPU, exhibit frame rates up to 40 fps (320x240) or 33 fps (640x480), in the optimized mode (45% performance penalty against the straightforward one), for ROIs of 16x16 pixels in RGBA textures containing up to 16384 particles, where we use 3 LODs for the multiresolution stage and the Random Local Search algorithm for the refinement.

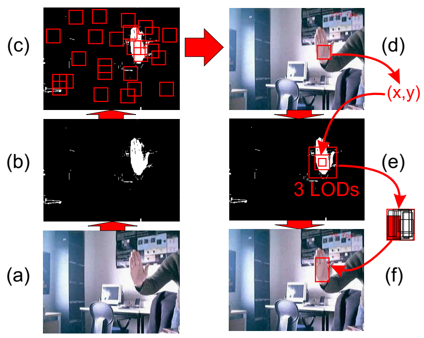

Figure 1: *Optimized Particle Filter scheme.*

3 Conclusion

In this work, we have shown a multiresolution and Local Search procedure extension applied to a previous particle filter approach to increase its 2D visual tracking performance. For every computational part of the process, we are using the Graphics Processing Unit (GPU) as the main processor due to its considerably more efficient handling of pixel-to-pixel comparisons. In addition to allowing us to perform more accurate tracking by adjusting the level of detail, the GPU's speed advantage makes it possible to adapt rectangular windows to the shape of the tracked object, yielding much greater flexibility than the square windows used in coarser particle filter approaches. Results show that we are able to maintain real time processing on quite generous video resolutions, and with optimizations it is possible to extend the level of detail even further.

References

MONTEMAYOR, A. S., PANTRIGO, J. J., CABIDO, R., PAYNE, B. R., CABIDO, Á. S., AND FERNÁNDEZ, F. 2006. Improving GPU particle filter with shader model 3.0 for visual tracking. In *Proc. of ACM SIGGRAPH 2006-Research Posters.*

*e-mail: raul.cabido@urjc.es

[0] This research has been supported by CICYT TIN2005-08943-C02-02