Nowadays, this whole world is undergoing a revolution of artificial intelligence,

which has entered almost every aspect of human industry and daily life.

Computer vision, acting as the eye to human beings, is a crucial source of information acquisition,

thus plays a deciding role in artificial intelligence.

Visual tracking is responsible for estimating the location and status of moving objects,

which makes it a building block of many computer visual applications and of

fundamental theoretical importance as well as irreplaceable application value.

In spite of 30 years’ research efforts, since the requirement for intelligence is continuously increasing,

visual tracking is still under the pressure of many challenging factors.

Firstly, the accuracy, robustness and adaptability of visual tracking algorithms are not developed enough to deal with various baffling factors and sophisticated object motion.

Secondly, with the purpose of higher accuracy,

tracking algorithms are becoming increasingly sophisticated,

which often leads to unacceptable tracking efficiency.

Last but not least, with the availability of heterogeneous compute platform,

the high-performance implementation of tracking algorithms is hard to achieve,

and the implementation also suffers from poor portability in different hardware devices.

To solve the mentioned issues,

this paper focuses on ``key techniques for high-performance visual tracking’’,

and tries to achieve high tracking performance through two research directions:

high-performance algorithm for visual tracking, and high-performance implementation of tracking algorithm.

The primary contributions and innovations of this paper are as follows:

1. A general method for cooperating detection proposal from object detection area with

visual tracking is proposed. In order to improve the scale and aspect ratio adaptability

of visual tracking algorithm, this paper combines a detection proposal generator named

EdgeBoxes, with a correlation filter based tracker.

To precisely discriminate flexible detection proposals, the correlation filter is optimized with

feature integration and robust updating scheme.

To improve tracking efficiency and robustness, proposal rejection and update with damping

are added into tracking process.

The resultant tracker shows superior robustness and adaptability on a large video dataset,

and satisfactory tracking speed is also achieved.

2. The effect of detection proposal in visual tracking is revealed, and a proposal generator is optimized for tracking task.

This paper adapts several representative detection proposal generators to tracking task,

and integrates them properly into a visual tracker.

Moreover, in order to conquer the weakness of EdgeBoxes in tracking,

an optimization step called ``background suppression’’ is added.

Through sufficient experiments, this paper proves that there is a strong positive correlation between

proposal quality and tracking accuracy.

Experiment results also validate the effectiveness of background suppression optimization for

EdgeBoxes.

3. The whole TLD tracking application is implemented on a heterogeneous compute platform

based on OpenCL parallel programming model.

By analyzing and extracting the computation-intensive parts and bottlenecks of TLD algorithm,

this paper parallelizes the follows components:

the feature extraction and classification procedures of Fern random forest,

the NCC computation of Nearest Neighbor classifier, and the overlap ratio calculation and sample

selection procedures of learning module.

Moreover, because the Fern random forest and LK optical flow tracking are two independent components, this paper overlaps their executions on two different compute devices.

By assigning each Kernel program to its suitable device, satisfactory over-all speedup is achieved,

which can adequately satisfy real time tracking demand.

4. A novel code transformation method is proposed to improve the performance portability of

GPU-specific OpenCL Kernels on CPUs.

This method is based on a new kind of array access descriptor from this paper,

and thus can eliminate all the redundant local memory usages along with corresponding

synchronizations in work-item coalescing step.

In the post-optimization step, this method not only extracts parallelism

and locality from the original GPU-specific Kernels,

but also considers the architectural details of the target CPUs,

which leads to further performance gains.

Experiments show that, for GPU-specific Kernels,

the new OpenCL runtime equipping with the method above can outperform the official

Intel OpenCL runtime on CPUs.