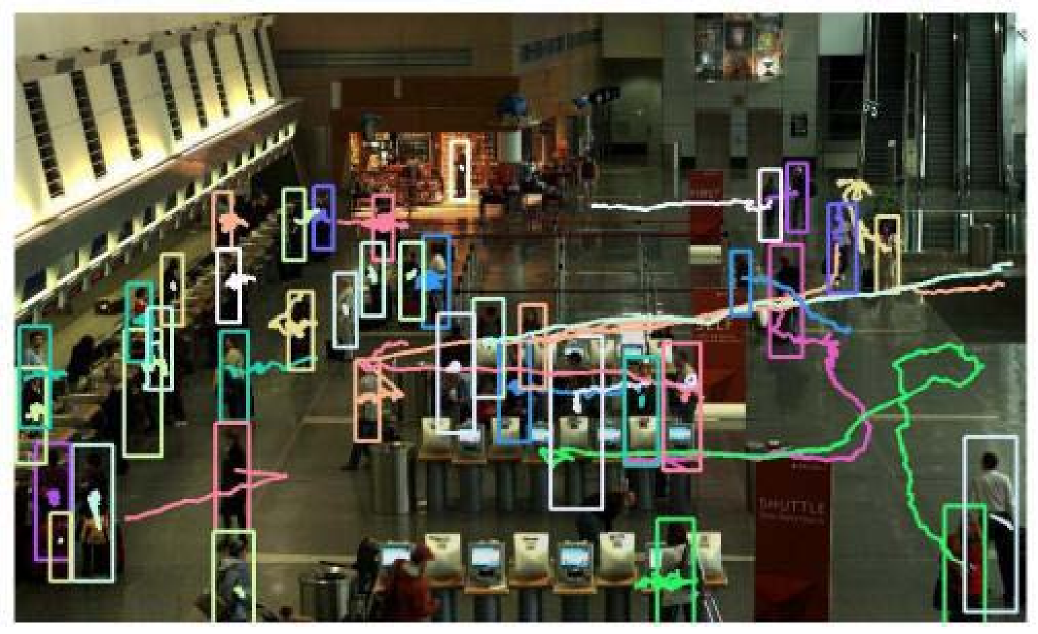
哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

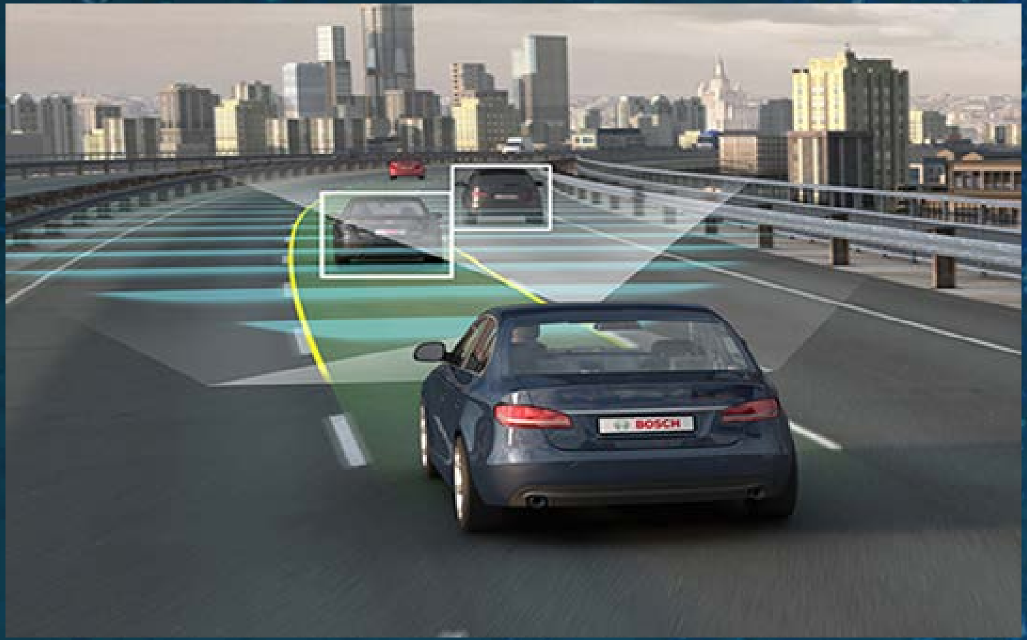
A Novel Low-cost FPGA-based Real-time Object Tracking System

Peng Gao[1], Ruyue Yuan[1], Zhicong Lin[1], Linsheng Zhang[2] and Yan Zhang[1]

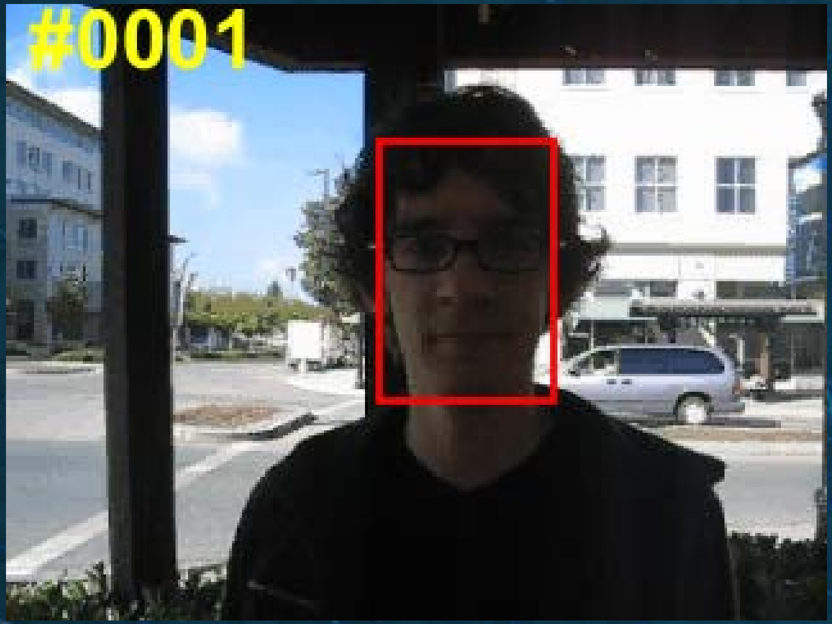
[1] Harbin Institute of Technology [2] Sanechips Technology Co., Ltd

Introduction:





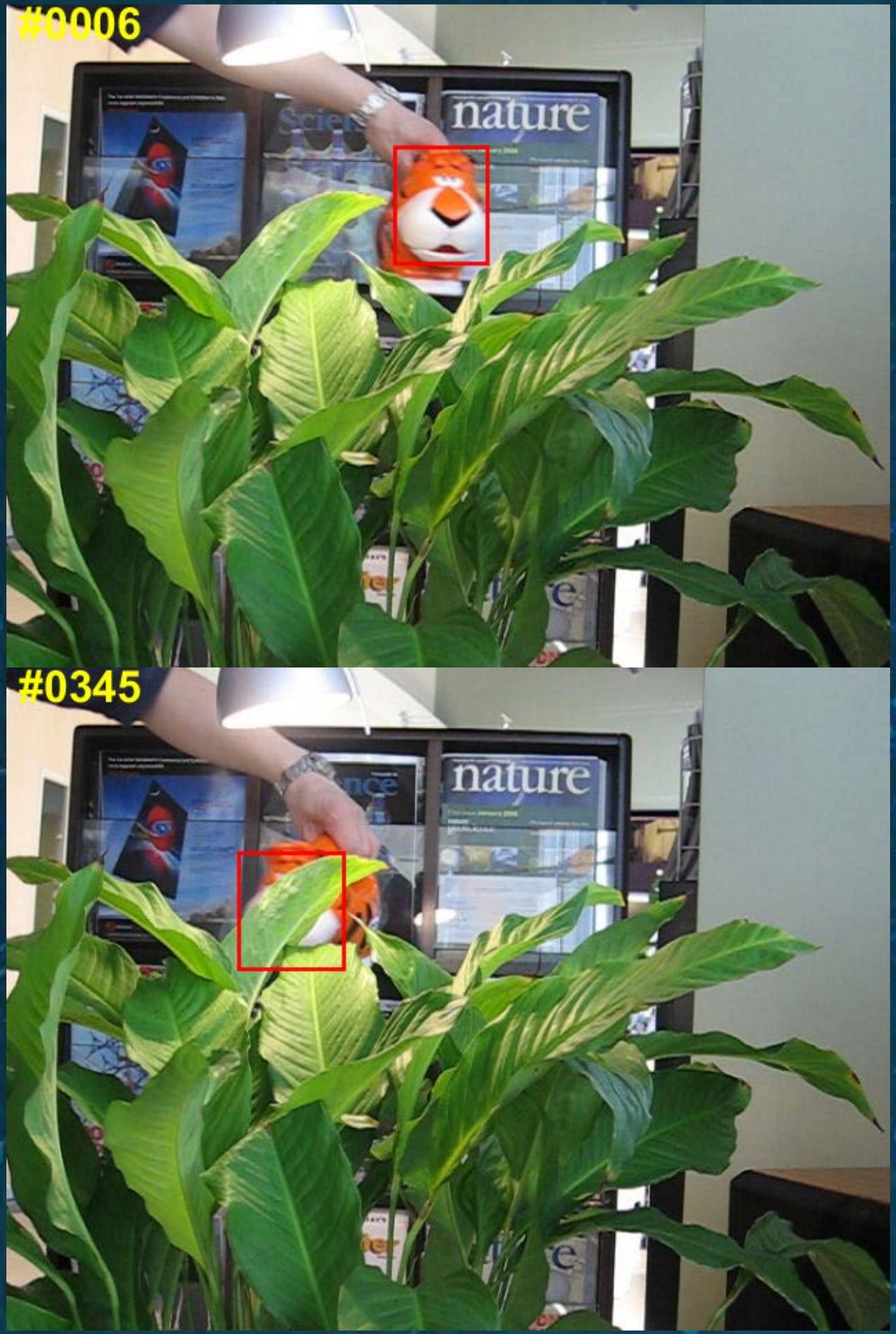
Introduction:



Visual object tracking:

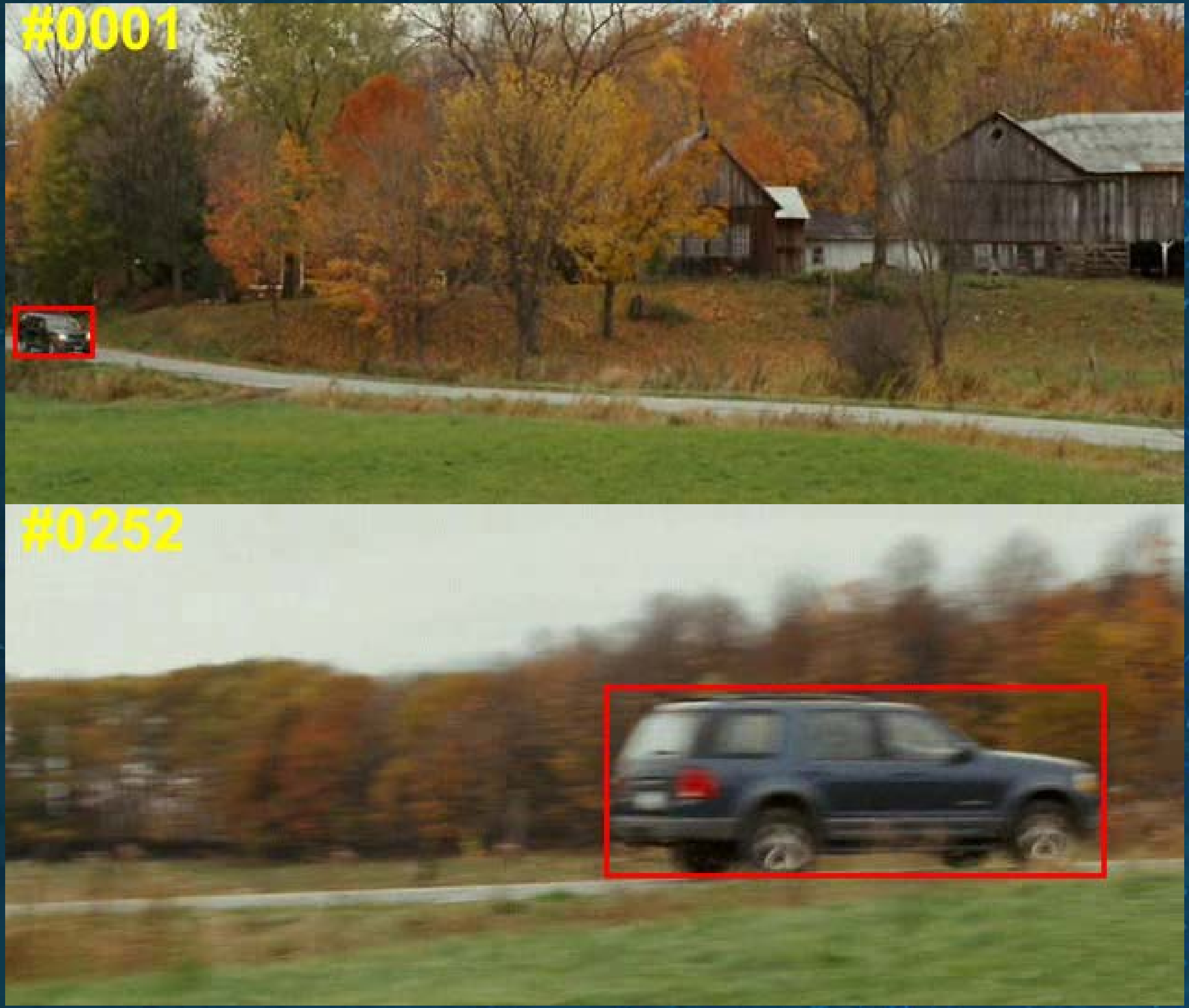
Single Object, Model-Free, Object Given, Real-time

Introduction:



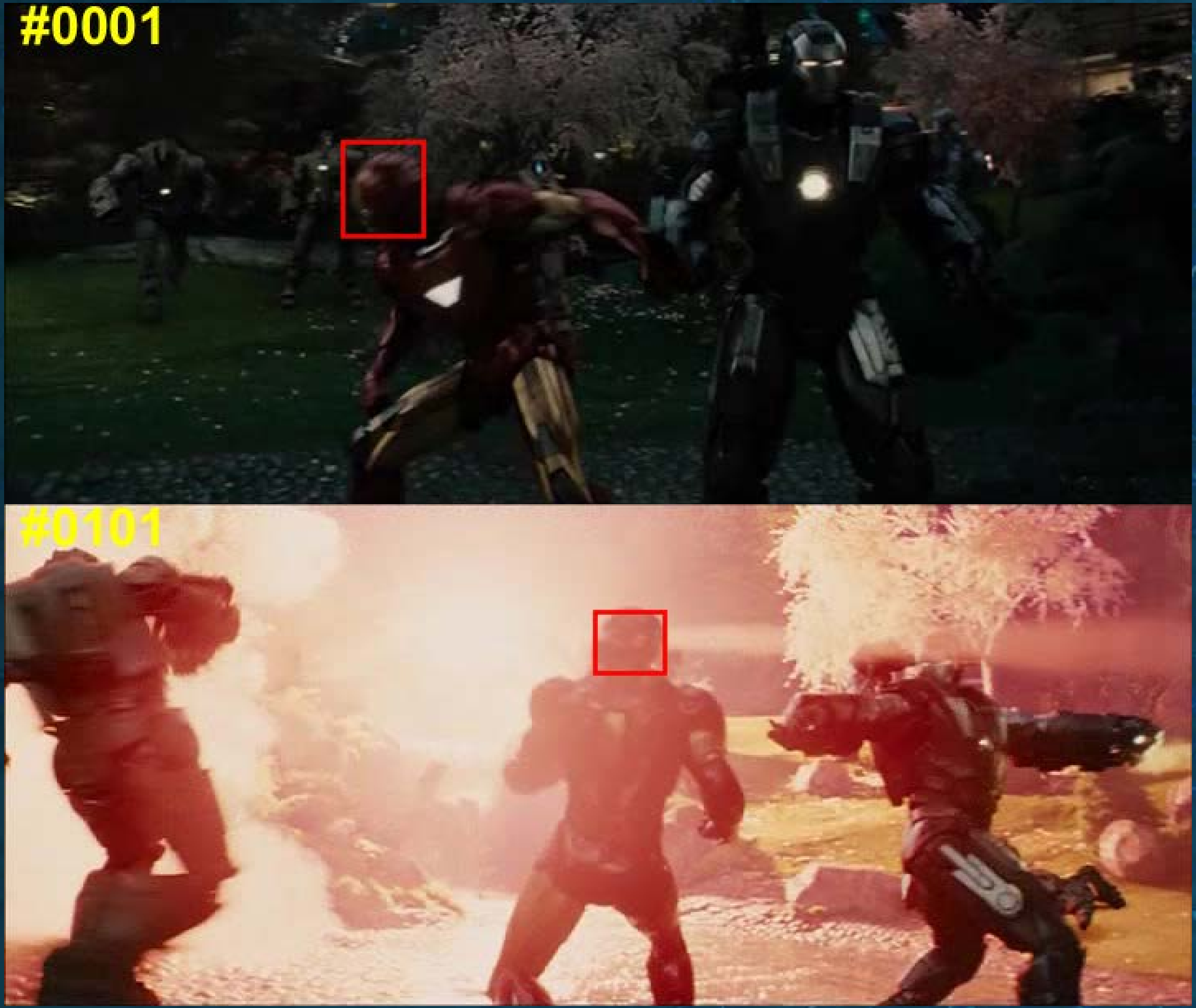
Occlusions

Introduction:



Scale variations

Introduction:



Illumination variations

Introduction:

Visual Object Tracking

- High accuracy and robustness
- Transcendental functions
- High-precision floating-point operations
- …

Introduction:

CPU

GPU

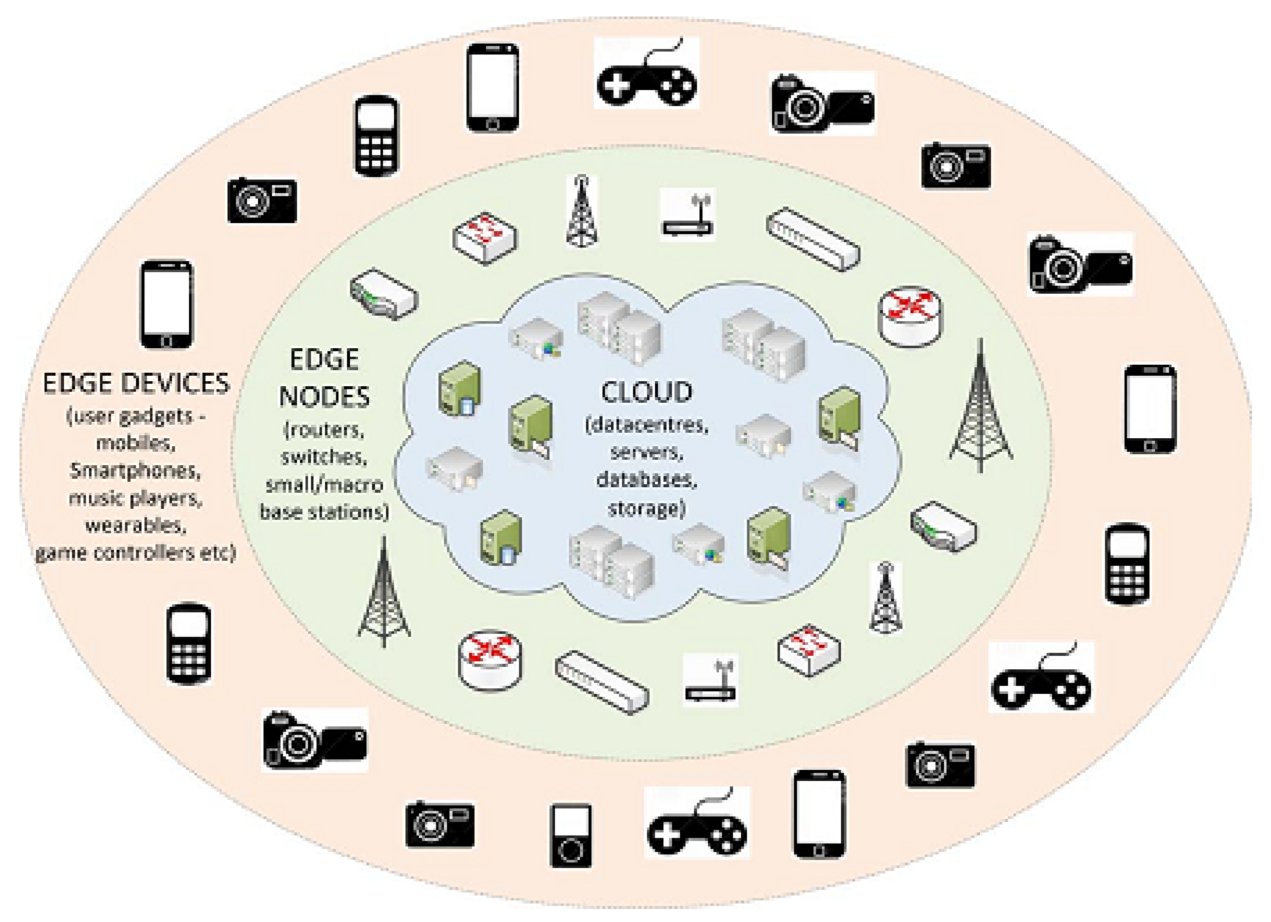
Introduction:

CPU/GPU

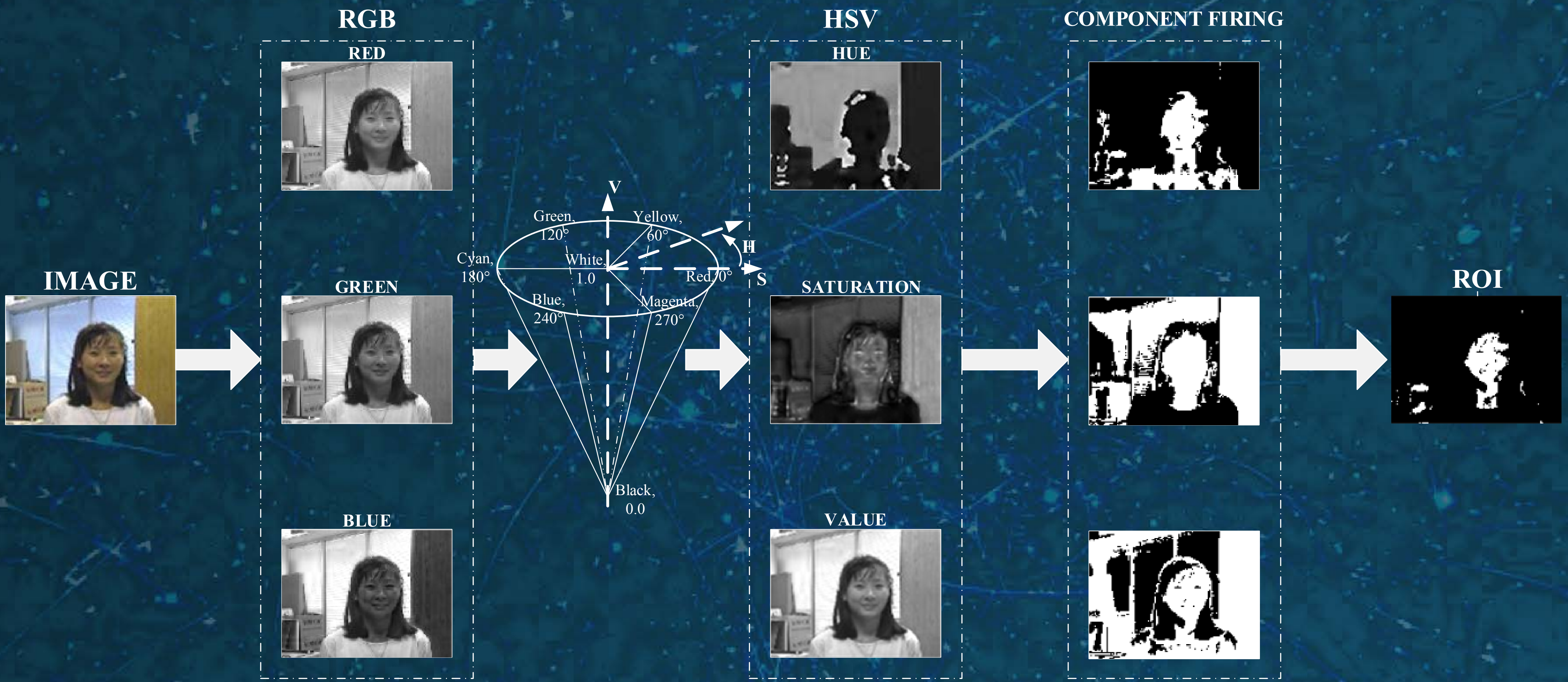
- High computational cost
- Consume a prohibitive amount of power

FPGA

- High energy efficiency
- Reconfigurable Computing



Algorithm and Theory: *Camshift tracker*



Algorithm and Theory: *Camshift tracker*

Frame image format transforming

$$R \leftarrow 1.164 \times (Y - 16) + 1.596 \times (Cr - 128)$$
$$G \leftarrow 1.164 \times (Y - 16) - 0.392 \times (Cb - 128) - 0.813 \times (Cr - 128)$$
$$B \leftarrow 1.164 \times (Y - 16) + 2.017 \times (Cb - 128)$$

Getting ROI Detective Value

$$\mathrm{V} \leftarrow \max(\mathrm{R}, \mathrm{G}, \mathrm{B})$$

$$\mathrm{S} \leftarrow \begin{cases} \dfrac{V - \min(R, G, B)}{V} & if\ V \neq 0 \\ 0 & other\ wise \end{cases}$$

$$\mathrm{H} \leftarrow \begin{cases} \dfrac{60(G - B)}{V - \min(R, G, B)} & if\ V = R \\ \dfrac{120 + 60(B - R)}{V - \min(R, G, B)} & if\ V = G \\ \dfrac{240 + 60(R - G)}{V - \min(R, G, B)} & if\ V = B \end{cases}$$

Algorithm and Theory: *Camshift tracker*

$$\begin{cases} H_{kc} = 1 \quad if\ |H_k - \bar{H}_{k-1}| < H_T \\ S_{kc} = 1 \quad if\ |S_k - \bar{S}_{k-1}| < S_T \\ V_{kc} = 1 \quad if\ |V_k - \bar{V}_{k-1}| < V_T \end{cases}$$

- $\boldsymbol{H_{kc}}$, $\boldsymbol{S_{kc}}$, $\boldsymbol{V_{kc}}$ are the classification value of the current image pixel.
- $\boldsymbol{\bar{H}_{k-1}}$, $\boldsymbol{\bar{S}_{k-1}}$, $\boldsymbol{\bar{V}_{k-1}}$ are the HSV component mean of the pixels in the previous frame bounding box.
- $\boldsymbol{H_T}$, $\boldsymbol{S_T}$, $\boldsymbol{V_T}$ are preset thresholds.

$$A_{kc} = 1 \quad if\ W(HSV) < A_T$$

- $\boldsymbol{W(HSV) = (\alpha|H_k - \bar{H}_{k-1}| + \beta|S_k - \bar{S}_{k-1}| + \gamma|V_k - \bar{V}_{k-1}|)}$ is the weight formula for HSV component.
- $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ are the weight coefficient of each component $(\boldsymbol{\alpha + \beta + \gamma = 1})$.

Algorithm and Theory: *Camshift tracker*

$$
\begin{cases}
M_{10} = \sum_{x} W(I(x,y)) \\
M_{01} = \sum_{y} W(I(x,y)) \\
M_{00} = \sum_{x}\sum_{y} W(I(x,y))
\end{cases}
$$

- $\boldsymbol{M_{10}}$ and $\boldsymbol{M_{01}}$ are the one-moments.
- $\boldsymbol{M_{00}}$ is the zero-moment.
- $\boldsymbol{W}(\cdot)$ represents a weighted function.
- $\boldsymbol{I}(\cdot)$ denotes the classification value of the pixel.
- The closer the current pixel is to the center of the previous frame, the larger the weight.

Algorithm and Theory: *Camshift tracker*

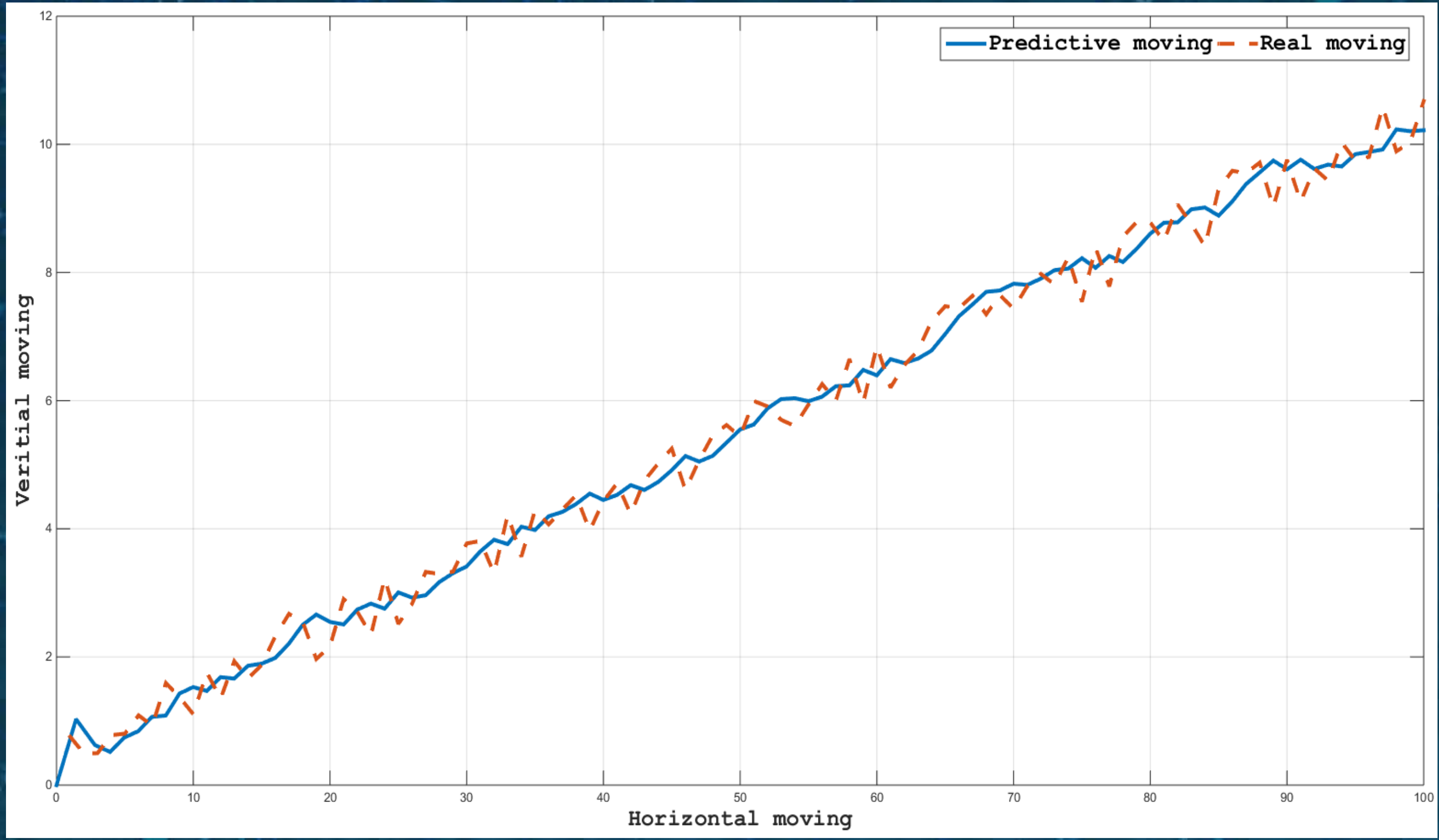
Centroid of bounding box : $\boldsymbol{x_c = \frac{M_{10}}{M_{00}},\ y_c = \frac{M_{01}}{M_{00}}}$

Size of bounding box : $\boldsymbol{w = 2 \times \sqrt{\frac{M_{00}}{256}}, h = 1.2w}$

Algorithm and Theory: *Kalman predictor*

Camshift algorithm has an extremely compelling effect when tracking object which moves in small range. But it becomes less effective when object moving in large range, as opposed to Kalman predictor.

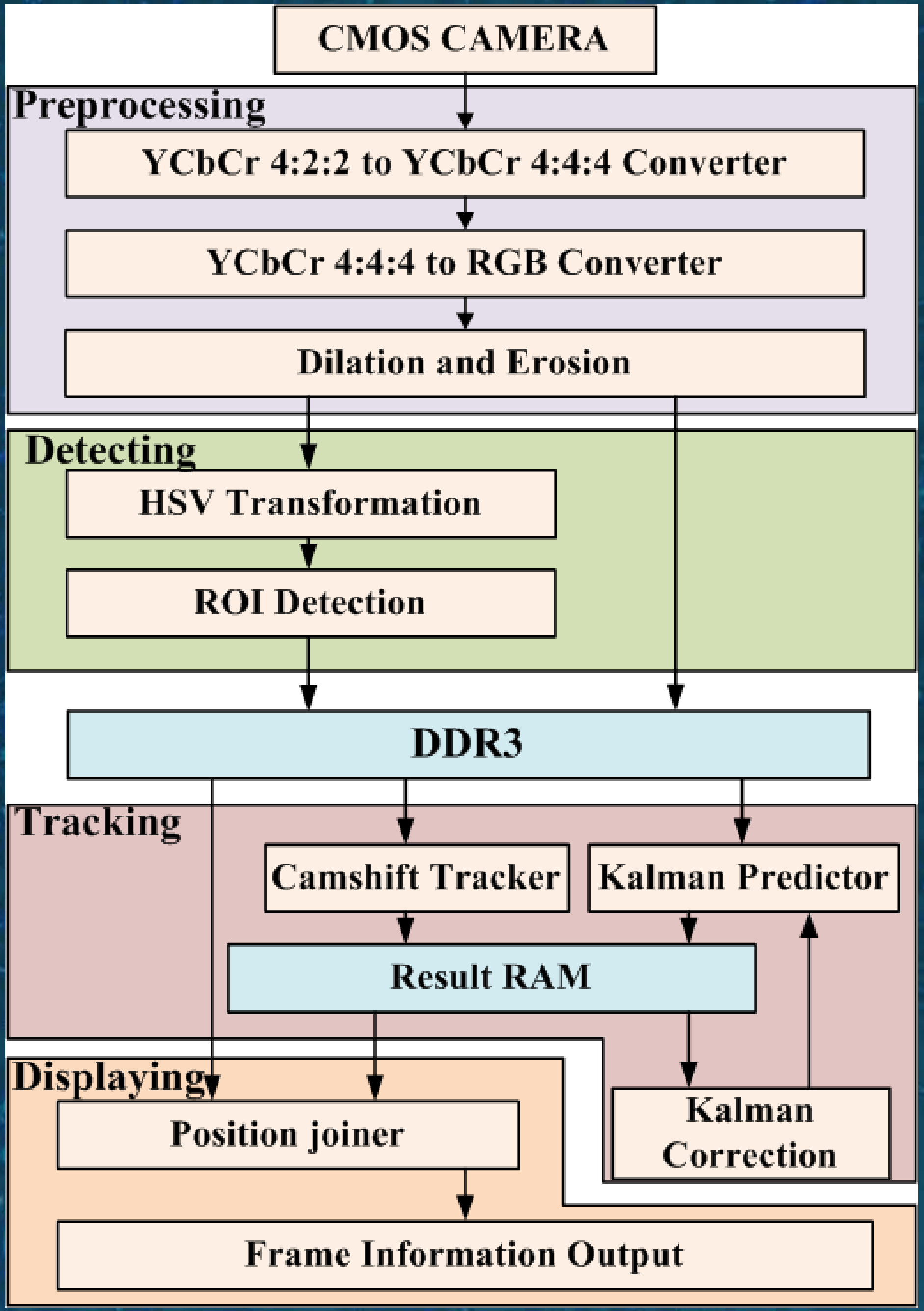
Therefore, based on the characteristics of the two algorithms, we optimized the classical Camshift algorithm by combined binary classifier with Kalman predictor.



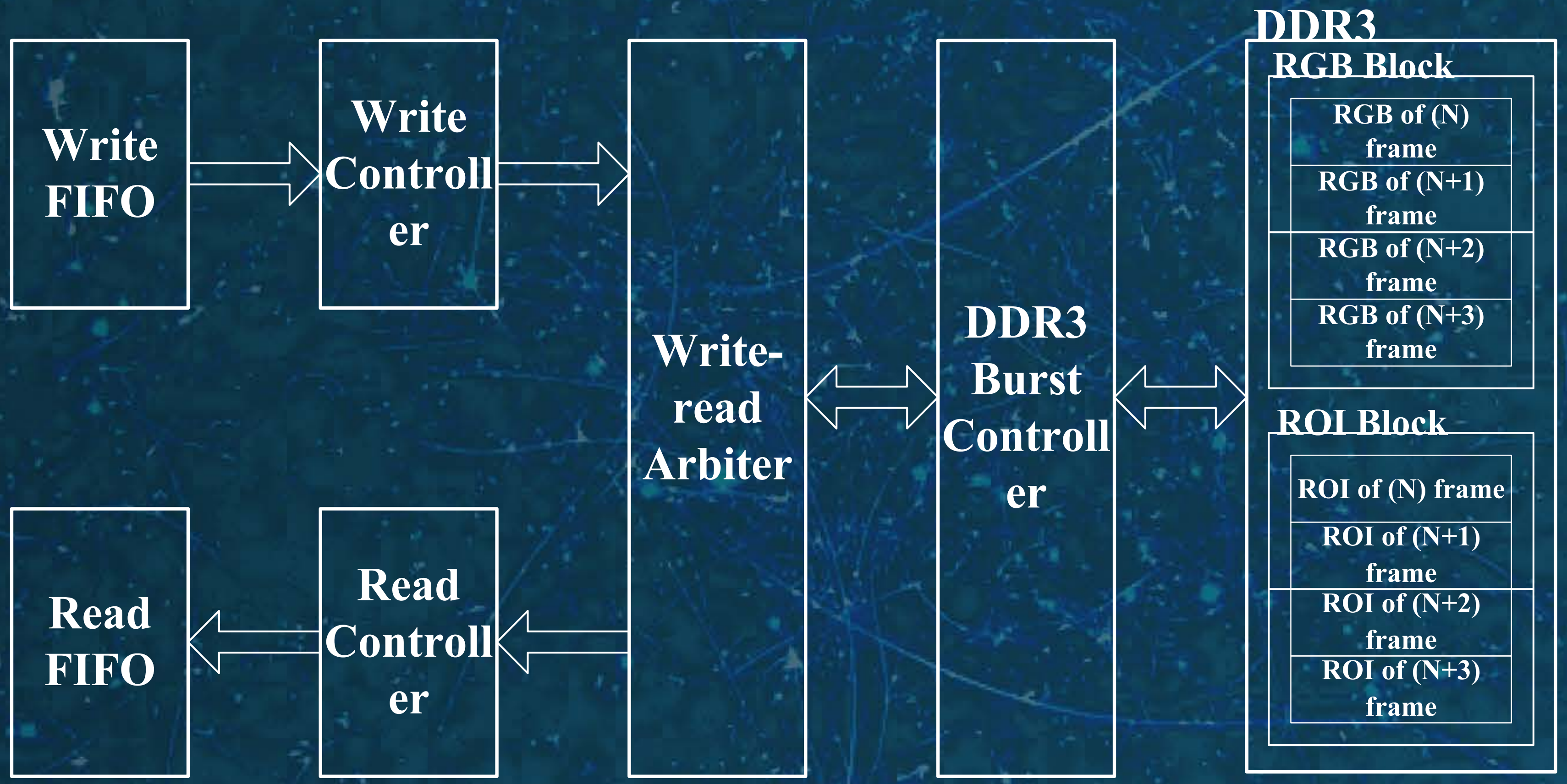
Hardware Implementation: *Overview*

We implement our hardware system based on a Xilinx Spartan-6 platform.

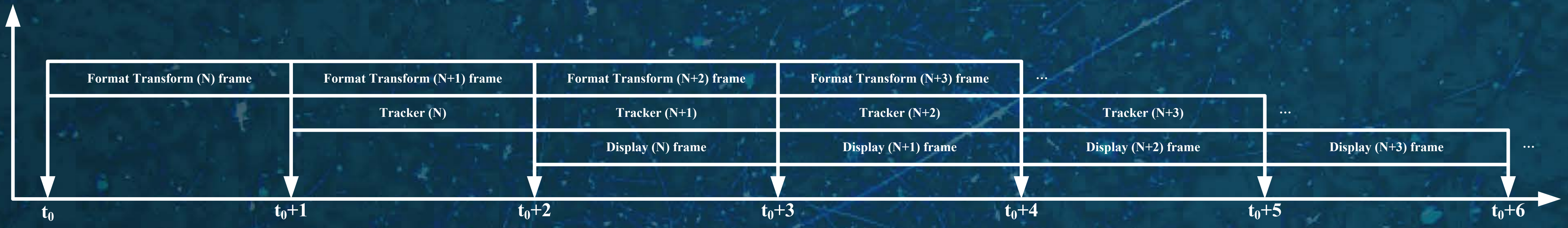
Hardware	Configuration
FPGA	Xilinx Spartan-6
System clock	148.5MHz
DDR3	2Gbit
Maximum bandwidth	10 Gbit/s



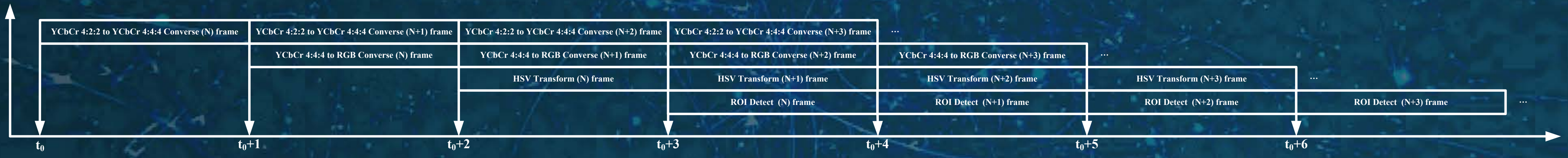
Hardware Implementation: *Optimization of DDR*



Hardware Implementation: *pipeline*

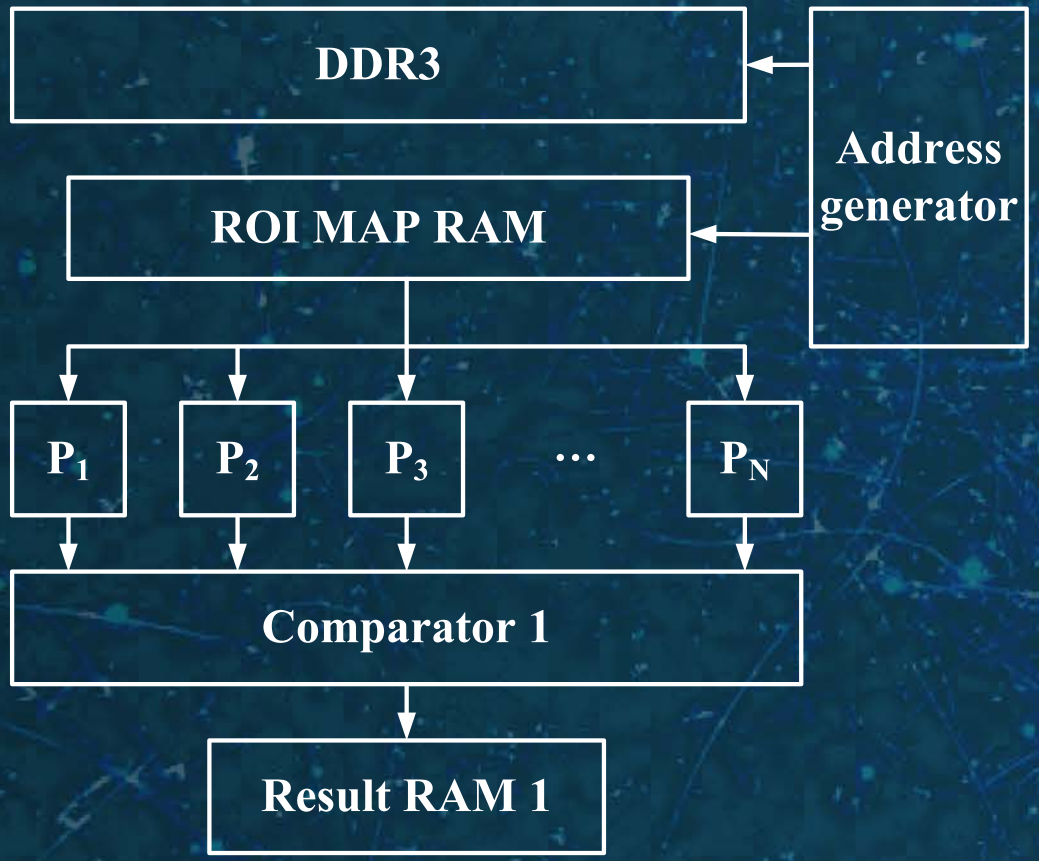


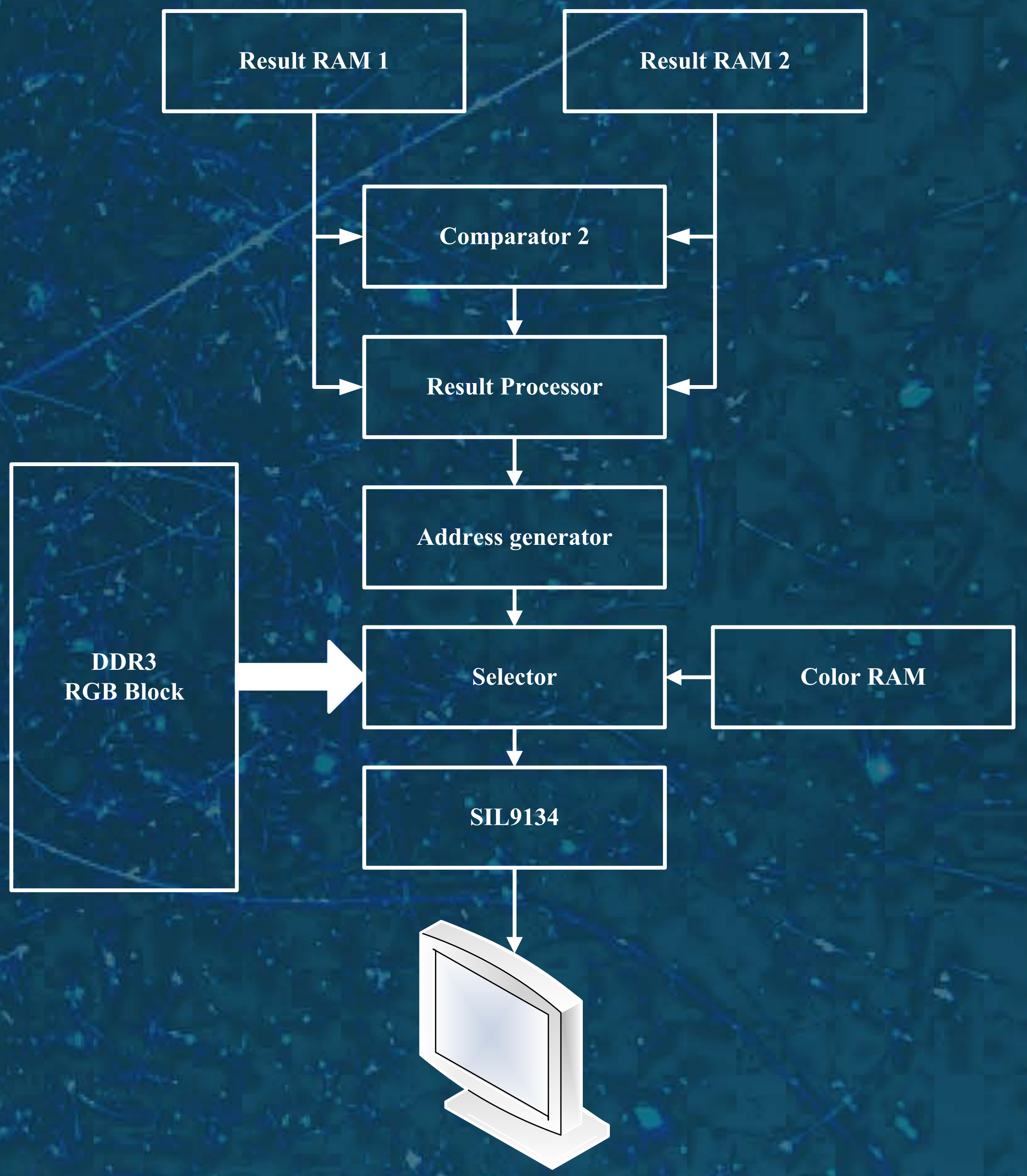
System stage-level pipeline



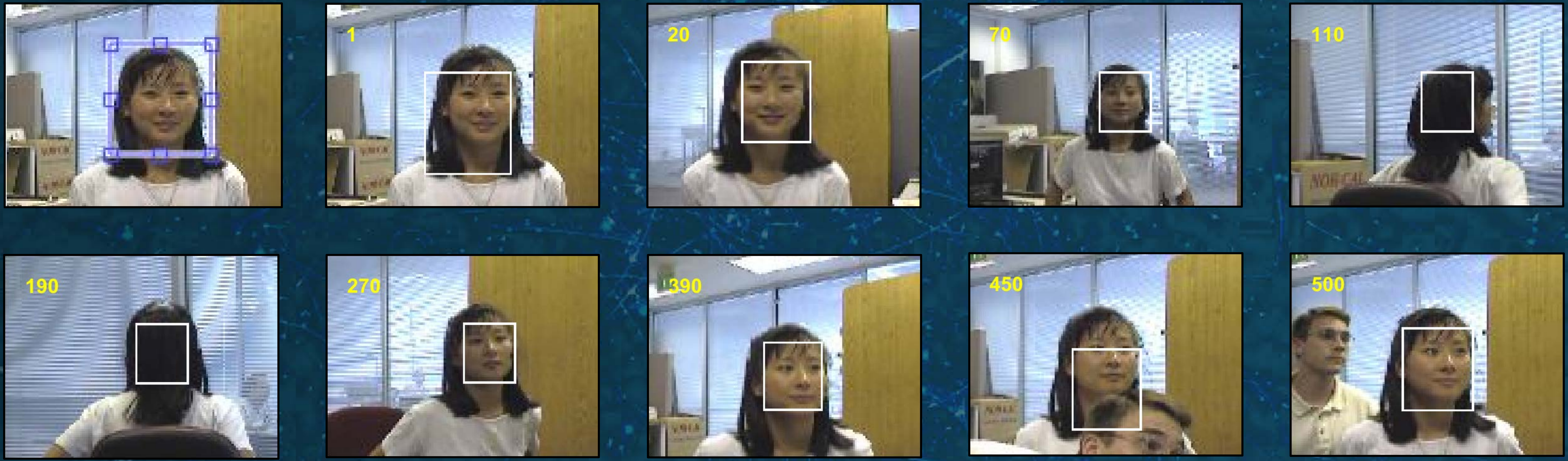
Preprocess stage pipeline

Hardware Implementation: *Parallel operating of algorithm module*





Results:

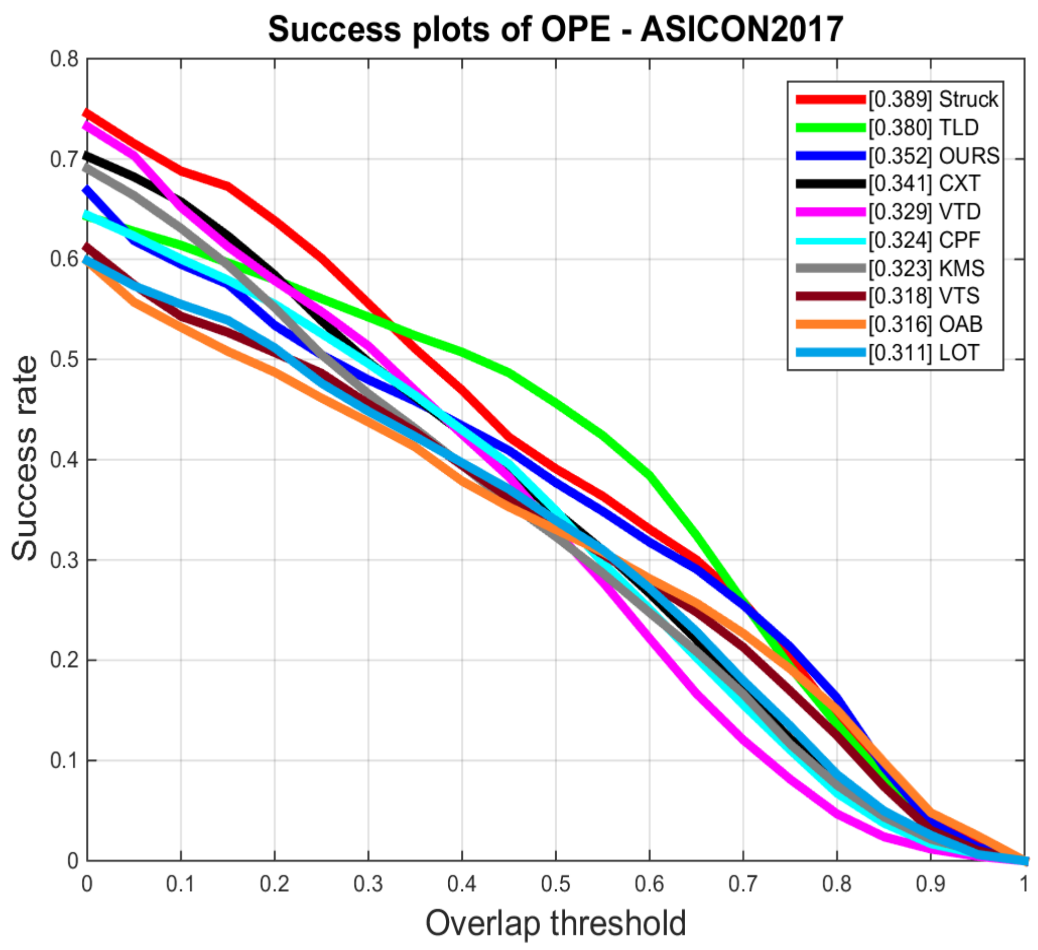


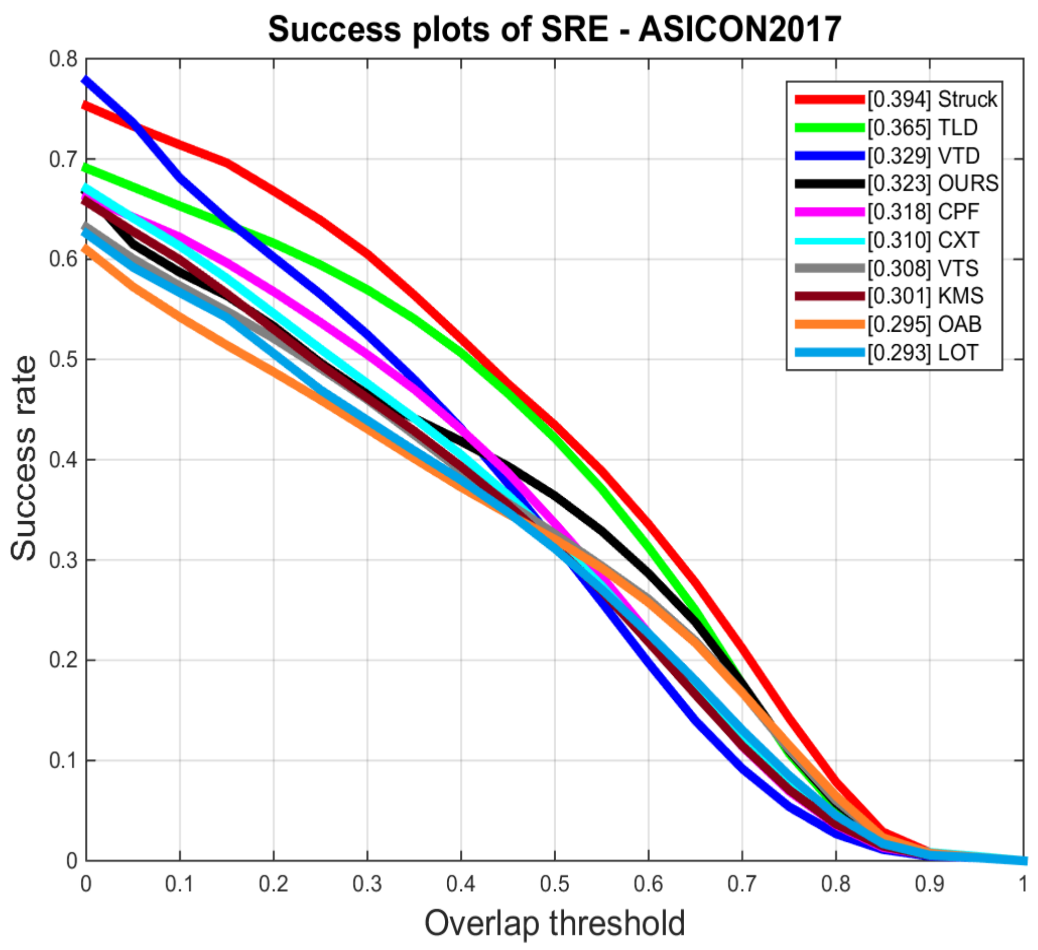
(Girl)

Results:

(DragonBaby)

Results: *The success plots of our tracker and other state-of-the-art trackers*





[1] Yi Wu, Jongwoo Lim and Ming-Hsuan Yang, Online Object Tracking: A Benchmark, 2013 IEEE Conference on Computer Vision and Pattern Recognition, p.2411-2418 (2013).

Results: *Tracking speed of our tracker and other state-of-the-art trackers*

Tracker	Precision	Mean FPS
Struck	53.5%	9.8
TLD	51.9%	24.4
OURS	48.4%	309.91
CXT	45.4%	13.9
VTD	44.4%	-
CPF	43.1%	-
VTS	42.2%	6.3
LOT	40.7%	0.5
OAB	40.5%	5.5
KMS	38.9%	-

FPGA Resource Utilization Summary :

Resource	Used Resources	Utilization
Slice Registers	16019	29%
Slice LUTs	14630	53%
DSP48A1s	42	72%

Performance Comparison :

System	Resolution	Real-time	Camera	Tracking Window
Ours	1920×1080	Yes	Moving	Adaptive
Liu's [2]	80×60	Yes	Fixed	Adaptive
Singh's [3]	-	Yes	Moving	Fixed
Elkhatib's [4]	640×480	Yes	Moving	-

[2] S.Liu, et al. Real-Time Object Tracking System on FPGAs. 2011 Symposium on Application Accelerators in High-Performance Computing, SAAHPC 2011. 1-7.
[3] L.Elkhatib, et al. An Optimal Design of Moving Objects Tracking Algorithm on FPGA. IEEE 4th International Conference on Intelligent and Advanced Systems, ICIAS 2012. 745-749.
[4] S.Singh, et al. FPGA-based Real-time Object Tracker using Modified Particle Filtering and SAD Computation. IEEE 18th International Symposium on VLSI Design and Test, 2014. 1-2.

Hardware system experiment:

哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

Thank you!

This work is supported by the Science and Technology Planning Project of Guangdong Province (Grant No.2016B090918047).