FPGA Accelerated Abandoned Object Detection

Rajesh Rohilla, Aman Raj, Saransh Kejriwal, Dr. Rajiv Kapoor

DELHI TECHNOLOGICAL UNIVERSITY

Problem Statement

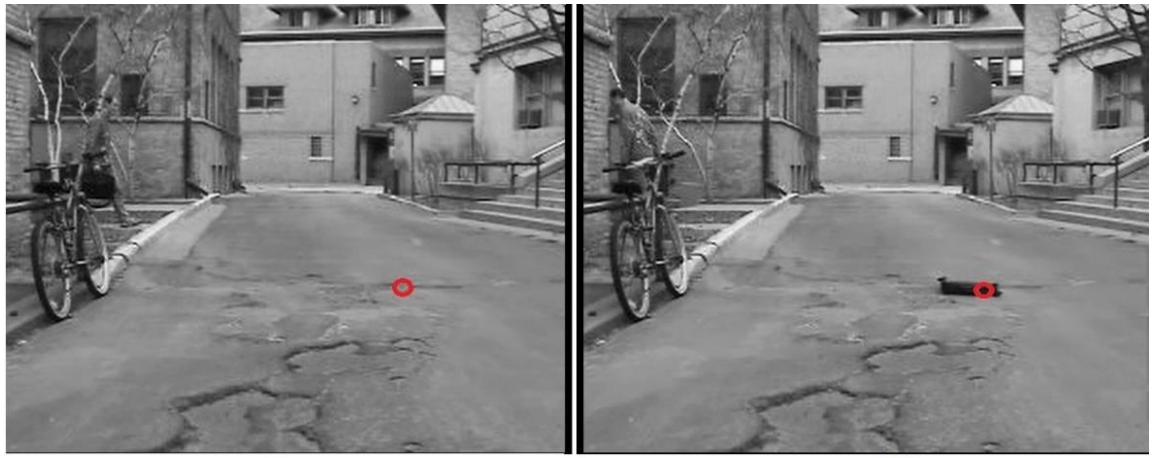
- Abandoned objects - a common sight at public places like railway station, public transport, marketplace etc.
- Can be dangerous for people if they contain explosive material planted by terrorists.
- CCTV monitoring for such objects needs manpower which can be difficult if area to be monitored is large.
- An automatic system is needed in place to detect such abandoned objects.

Snapshot of a typical overhead surveillance footage

What if the camera itself could locate unattended objects, and display their highlighted images to the security personnel ?

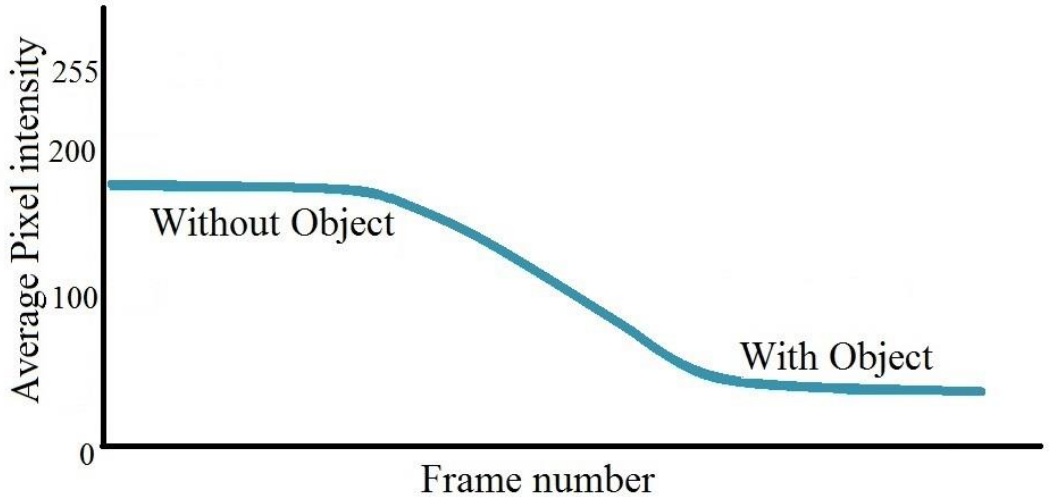
How do we do it?

- We utilized the following concept:



I II

A video frame- I without object and II with object

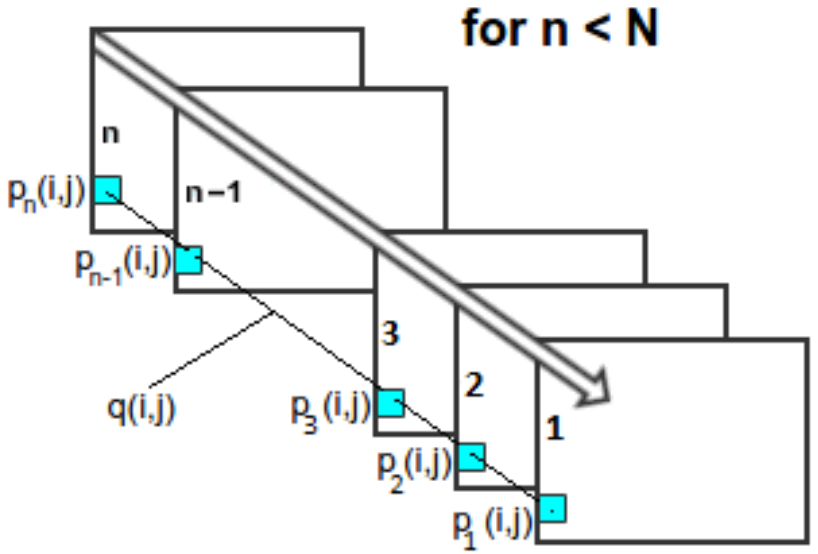


Variation of pixel intensity at encircled position in Fig.(I)-(II) with incoming frames. Averaging is performed over all the coming inputs video frames, highlighting effect of introduction of an object that is black in this case.

Algorithm

Obtaining Reference Static Frame

$I_n(i,j)$ pixel present at i^{th} row and j^{th} column of n^{th} frame



For each such pixel value a queue **Q(i,j)** of size **N**, a sum of pixel values **S(i,j)** and average of pixel values **A(i,j)** is maintained over the incoming frames.

$$\mathrm{Q(i,j)} = I_1(i,j),\ I_2(i,j),\ \ldots\ ,I_{n-1}(i,j),\ I_n(i,j)$$

$$S(i,j) = \sum_{k=1}^{n} I_k(i,j)$$

Algorithm

- When **n = N**, we get the sum **S(i,j)** and the average **A(i,j)** for each corresponding pixel in the frame using it's queue **Q(i,j)**

$$S_N(i,j) = \sum_{k=1}^{N} I_k(i,j)$$

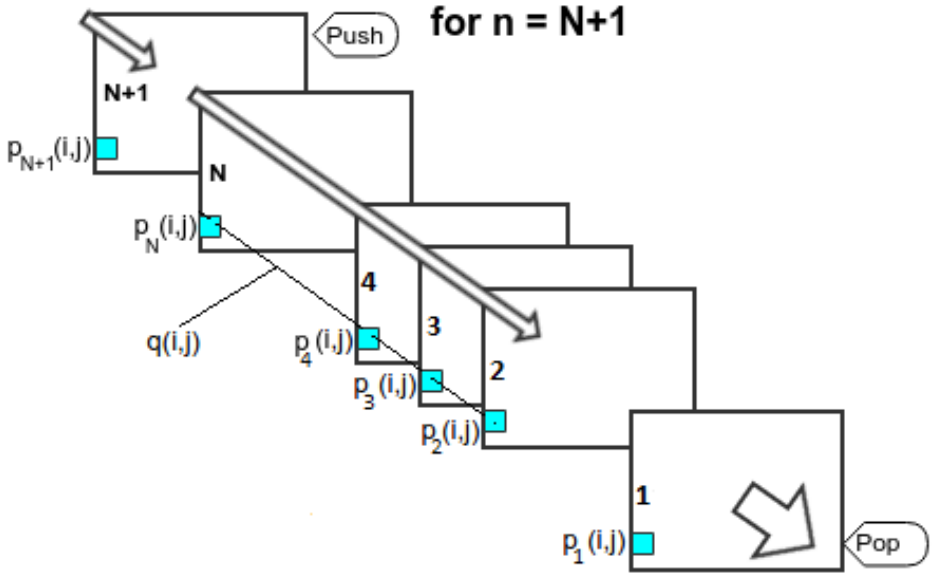
$$A_N(i,j) = S_N(i,j)/N$$

- Computes a reference static background by forming an image using averages of each pixel in first **N** frames. Saved on disk , can be updated after every **X** minutes.
- **N = 100**, is used.

Algorithm

Updating Current Static Frame

- We keep updating our current static background using same computation



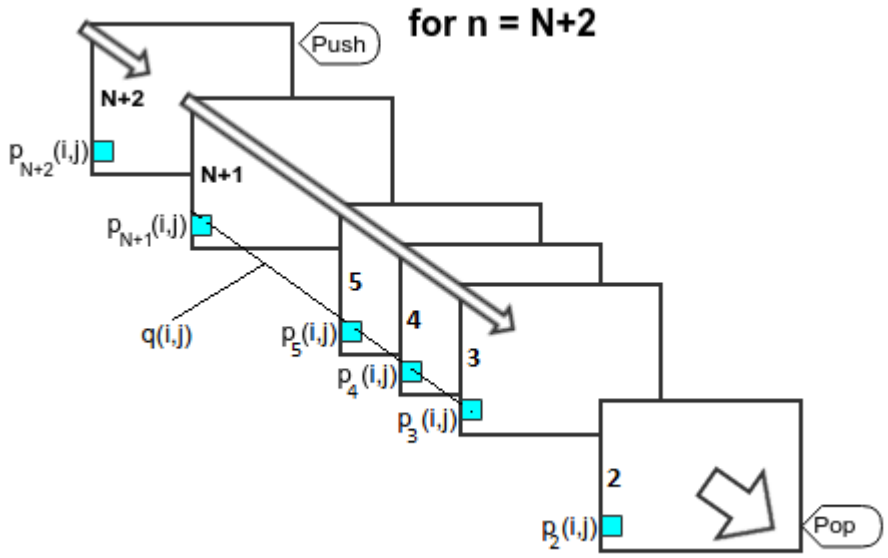
- For **n > N**, queue becomes

$$Q(i,j) = I_{n-N+1}(i,j),\ I_{n-N+2}(i,j),\ \ldots\ ,I_{n-1}(i,j),\ I_n(i,j)$$

Algorithm

$$S_{CN}(i,j) = S_{PN}(i,j) + I_n(i,j) - Q(i,j).front$$

where for the queue $Q(i,j)$, $S_{CN}(i,j)$ is the sum of current N values of pixels in queue, $S_{PN}(i,j)$ is the sum of previous N values of pixels in queue, $I_n(i,j)$ is pixel value in the latest frame that is pushed in queue and $Q(i,j).front$ is the oldest pixel value in the queue that is popped out.



Average of current N elements of the queue $A_{CN}(i,j) = S_{CN}(i,j)/N$

Algorithm

- We model a background using:

$$\left[B_{CN}(i,j)\right]_{a\times b} = \left[A_{CN}(i,j)\right]_{a\times b}$$

where $a \times b$ is the dimension of the frame and $B_{CN}(i,j)$ is the pixel value in the static background frame.

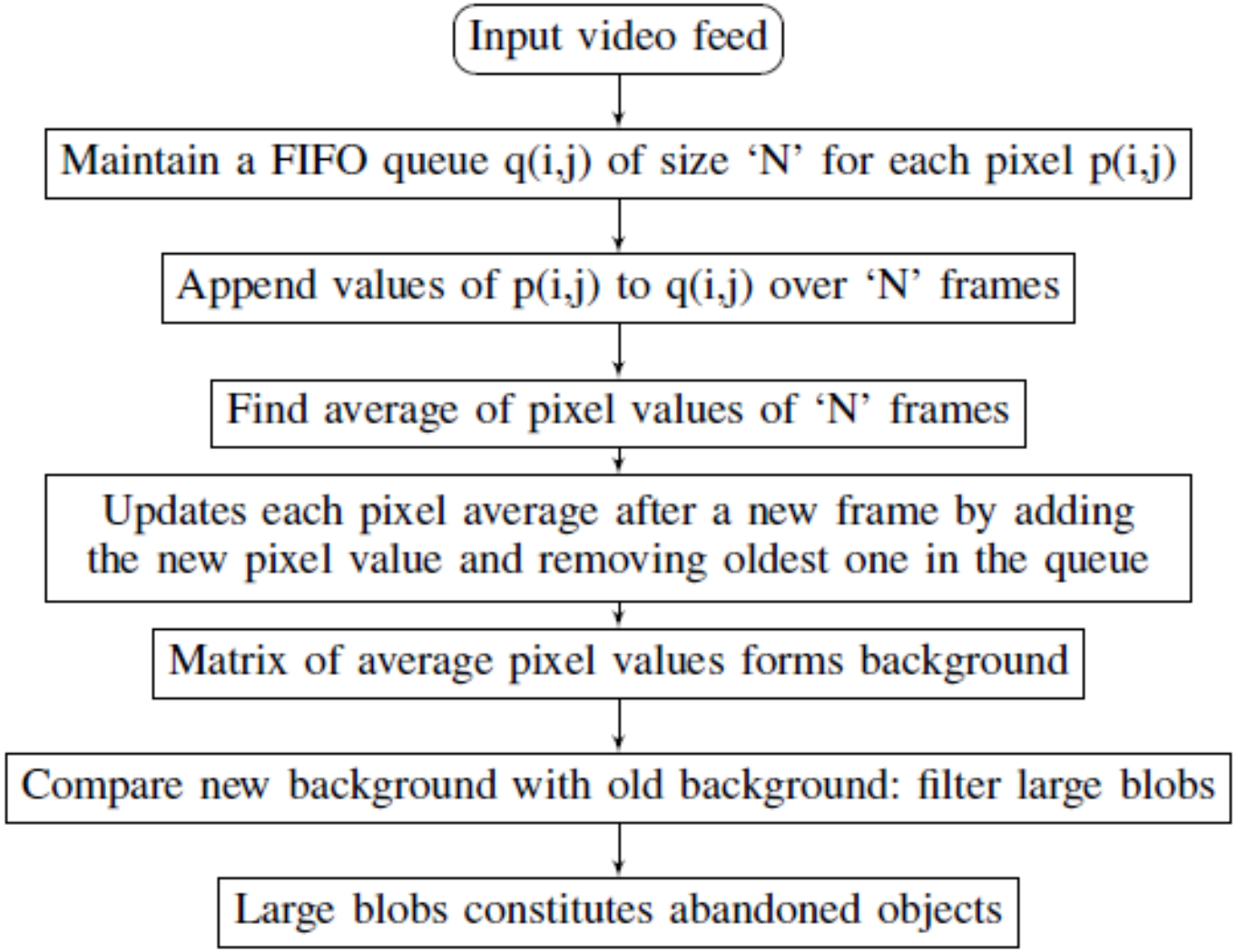
Current Frame Comparison

- Updated current static frame and reference static frame compared by calculating difference to detect abandoned objects.

Blob Detection & Decision Making

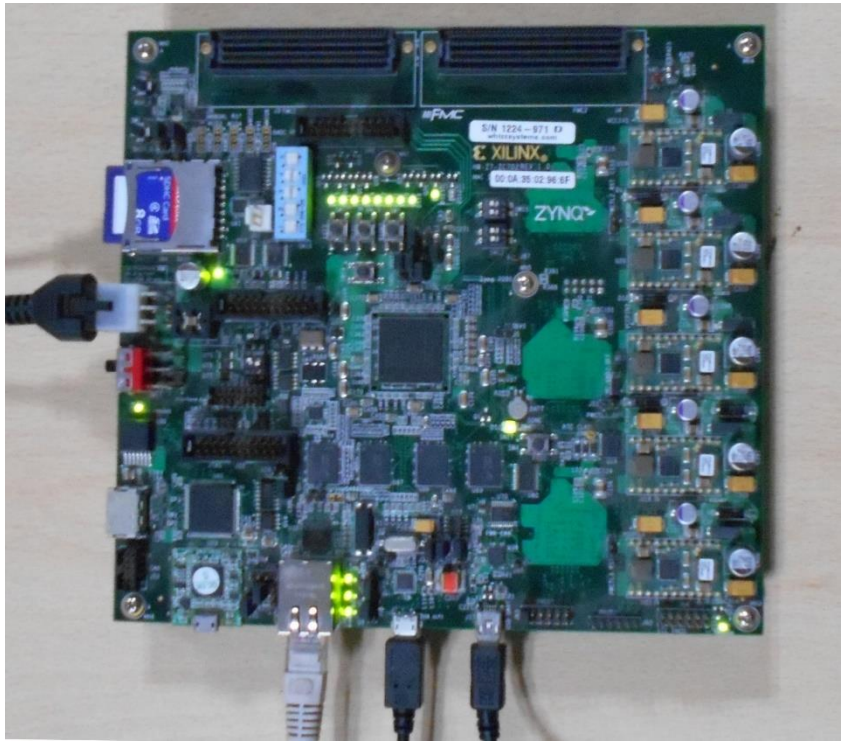
- Attempts to remove the effect of small blobs caused due to intermittent movement in video feed

Flowchart


Input video feed
Maintain a FIFO queue q(i,j) of size 'N' for each pixel p(i,j)
Append values of p(i,j) to q(i,j) over 'N' frames
Find average of pixel values of 'N' frames
Updates each pixel average after a new frame by adding the new pixel value and removing oldest one in the queue
Matrix of average pixel values forms background
Compare new background with old background: filter large blobs
Large blobs constitutes abandoned objects

FPGA Implementation

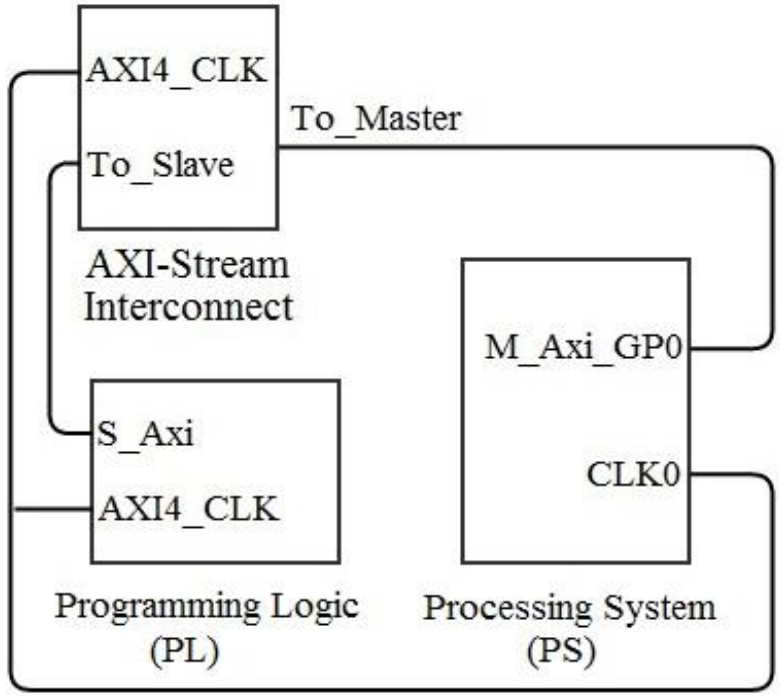
- Serial processing of such pixel queues on a conventional computing platform is a relatively slow process, so algorithm synthesized on FPGA.
- Hardware implementation speeds up algorithm execution by exploiting it parallel nature.
- Xilinx Zynq-7020 all programmable system on chip (SoC) FPGA board used.



Xilinx FPGA Board

FPGA Implementation

- Processing System (PS) that contains Dual ARM Cortex-A9 Processor
- Programmable Logic (PL) that contains Artix-7 FPGA
- Our logic is programmed on PL part using Vivado High-Level Synthesis (HLS) library provided by Xilinx.
- Data transfer using the AXI-Stream bus which is highly efficient and fast for real-time high-bandwidth data transfer



Block diagram of our FPGA system

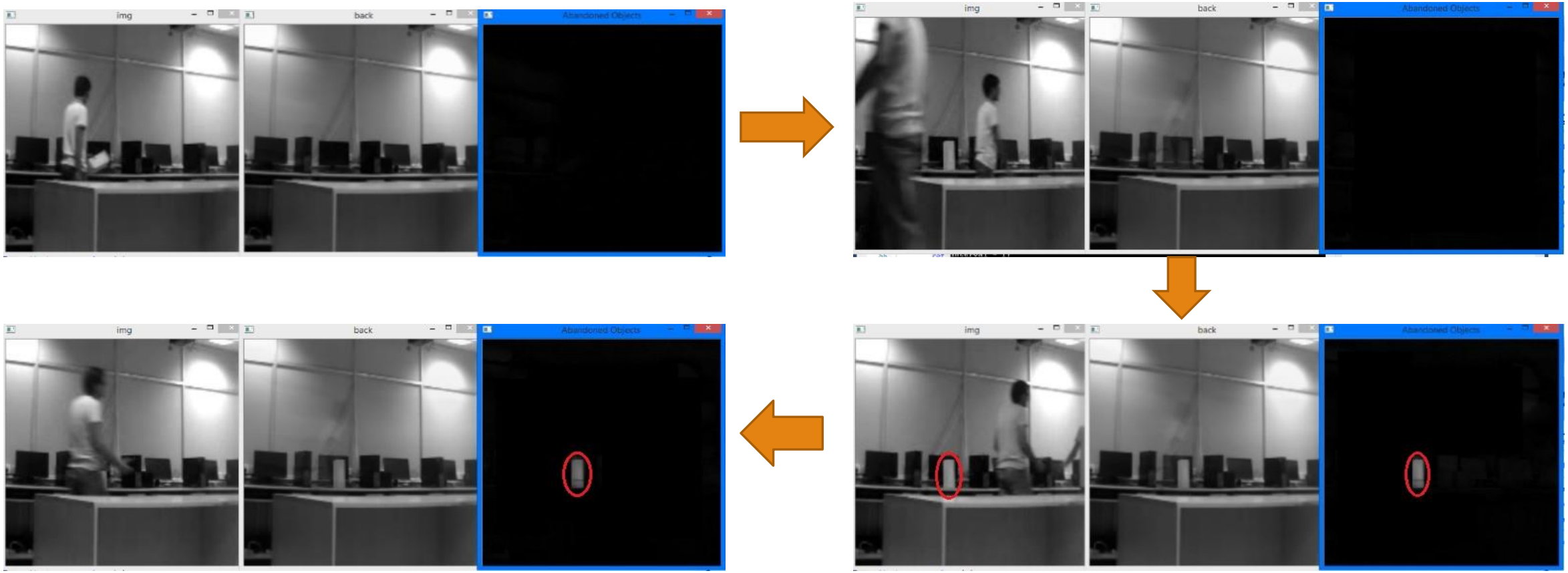
Results (AVSS2007)

- Algorithm tested on AVSS2007 video dataset, that contains abandoned objects in public places

Detection results of the sequence AB-Easy of AVSS2007

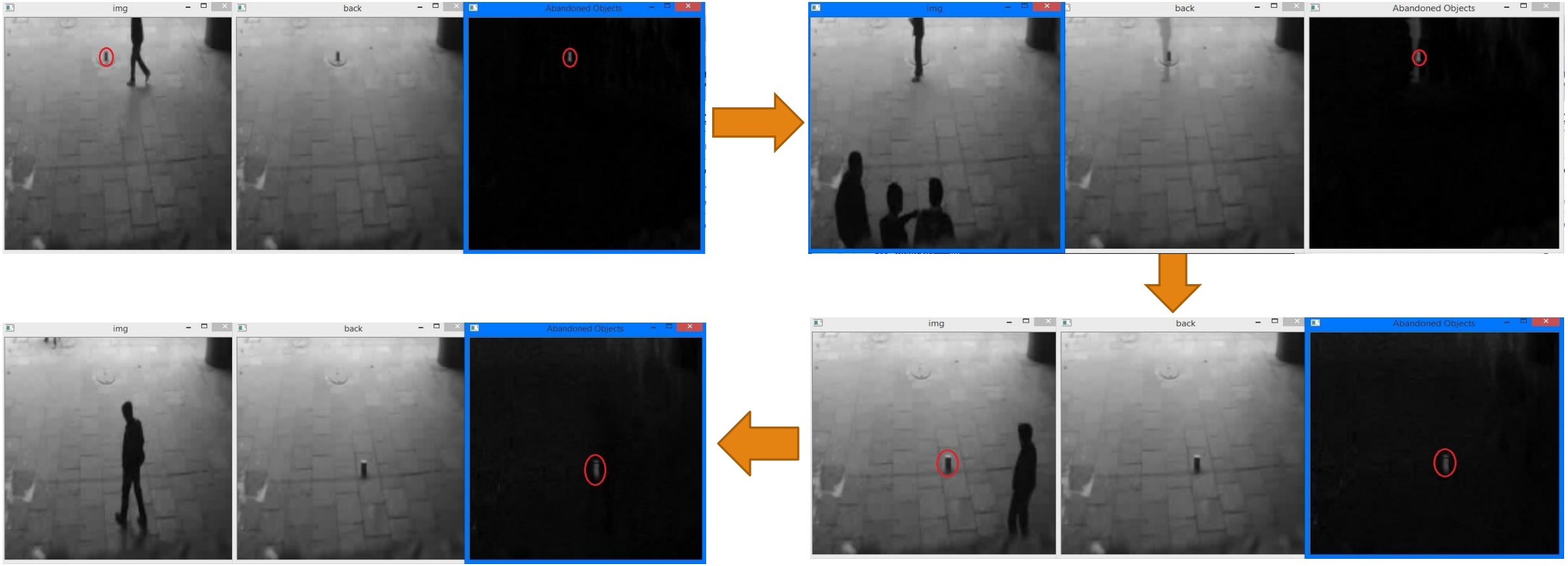
Results (our Dataset)

- Horizontally placed camera on table top in minimally crowded place, our lab:



Results (our Dataset)

- An overhead Surveillance camera in very crowded place.



Thank You