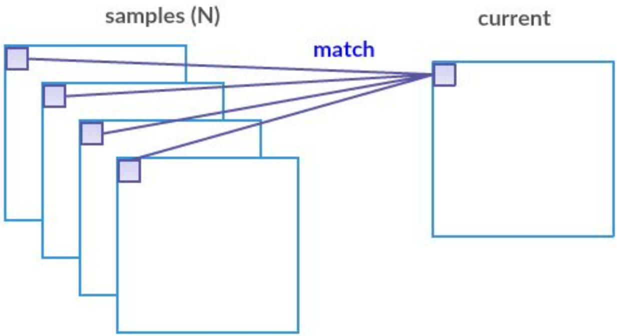
# Background Substration

Diego Javier Quispe
David Choqueluque Roman

# Method: Classification



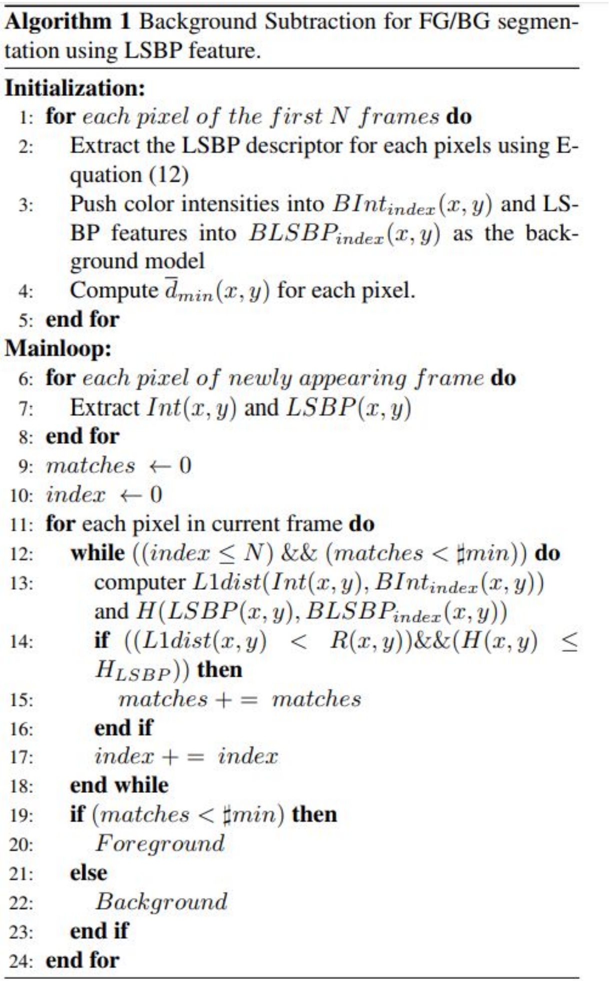
Match:

- Hamming Distance: LSBP
- L1 Distance: Color

Background model

$$B(x,y) = \{B_1(x,y), ..., B_{index}(x,y), ..., B_N(x,y)\}$$

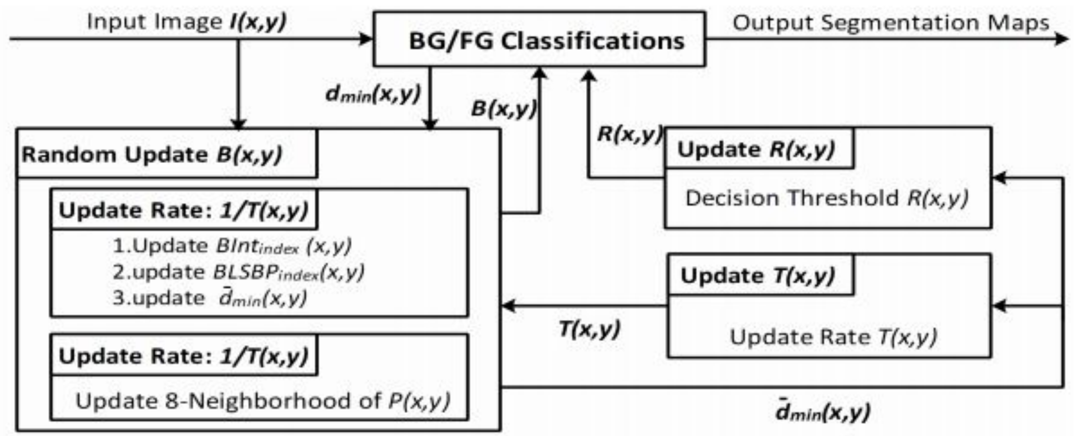
$LSBP(x,y)$ $Int(x,y)$,

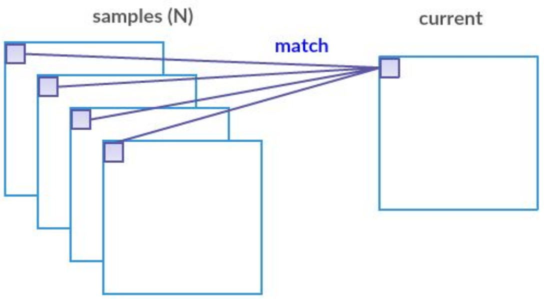


**Algorithm 1** Background Subtraction for FG/BG segmentation using LSBP feature.

```
Initialization:
for $each\ pixel\ of\ the\ first\ N\ frames$ do
    Extract the LSBP descriptor for each pixels using Equation (12)
    Push color intensities into $BInt_{index}(x,y)$ and LSBP features into $BLSBP_{index}(x,y)$ as the background model
    Compute $\overline{d}_{min}(x,y)$ for each pixel.
end for
Mainloop:
for $each\ pixel\ of\ newly\ appearing\ frame$ do
    Extract $Int(x,y)$ and $LSBP(x,y)$
end for
$matches \leftarrow 0$
$index \leftarrow 0$
for each pixel in current frame do
    while $((index \leq N)\ \&\&\ (matches < \sharp min))$ do
        computer $L1dist(Int(x,y), BInt_{index}(x,y))$ and $H(LSBP(x,y), BLSBP_{index}(x,y))$
        if $((L1dist(x,y) < R(x,y))\&\&(H(x,y) \leq H_{LSBP}))$ then
            $matches += matches$
        end if
        $index += index$
    end while
    if $(matches < \sharp min)$ then
        $Foreground$
    else
        $Background$
    end if
end for
```

# Method: Update





Match:

- Hamming Distance: LSBP
- L1 Distance: Color

Background model

$$B(x,y) = \{B_1(x,y), ..., B_{index}(x,y), ..., B_N(x,y)\}$$

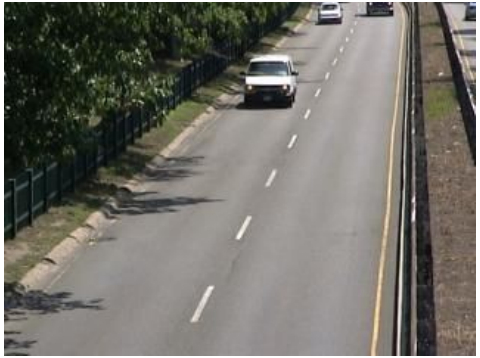
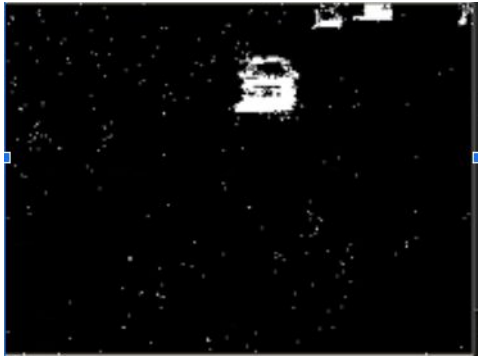
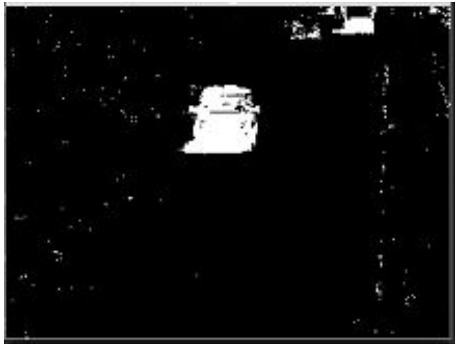
$$LSBP(x,y) \quad Int(x,y),$$

- R(x,y): per-pixel color intensity
- d^min(x,y): average dmin
- dmin: min color distance(L1)(matching)

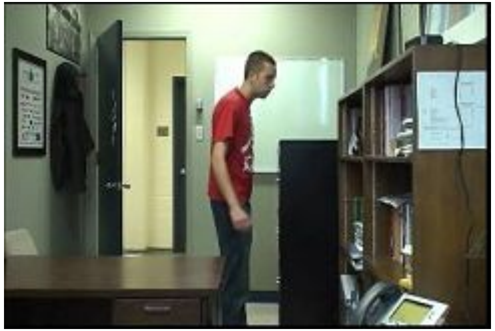
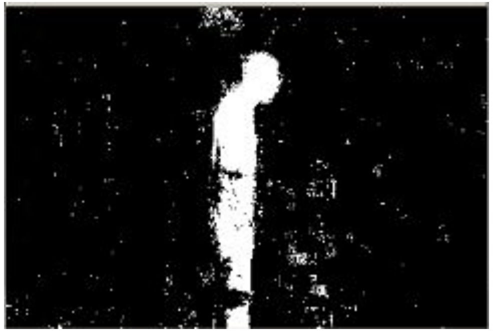
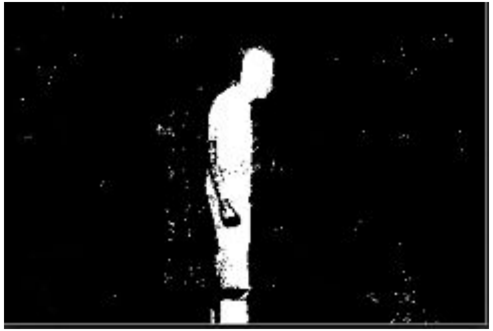
# Processing time per frame

| Stage | Frame | Python | C++ with threads | C++ with cuda |
|---|---|---|---|---|
| highway | 240x320 | 5.508s | 0.864s | 0.484s |
| Office | 240x360 | 5.858s | 0.980s | 0.563s |
| peopleInShade | 244x380 | 6.337s | 1.096s | 0.587s |
| streetLight | 240x320 | 5.131s | 0.874s | 0.493s |
| Own video | 270x480 | - | 1.652 | 0.836 |

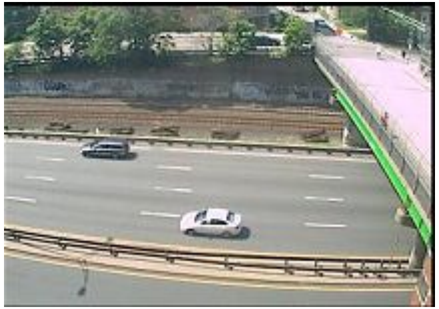
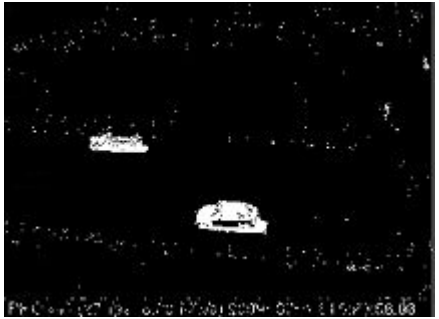
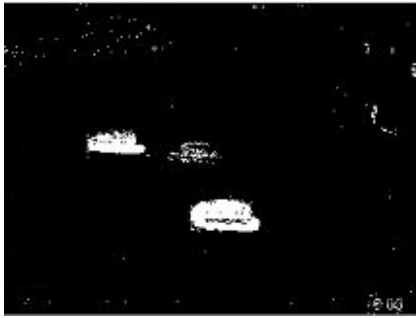
# CDNET2012 - Highway(84 - 108)

| Input | Python | Our Implementation |
|---|---|---|
|  |  |  |

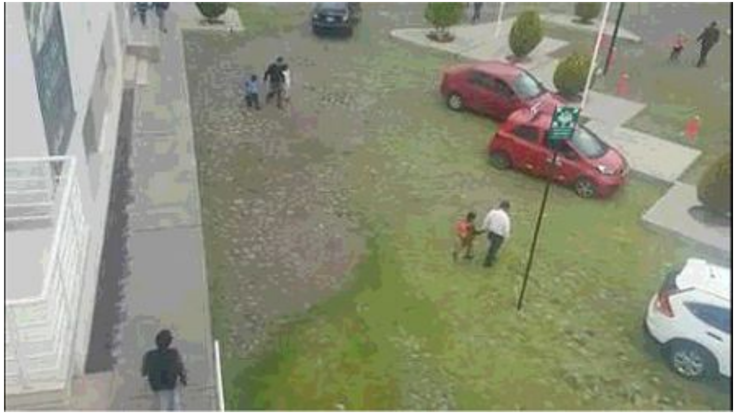
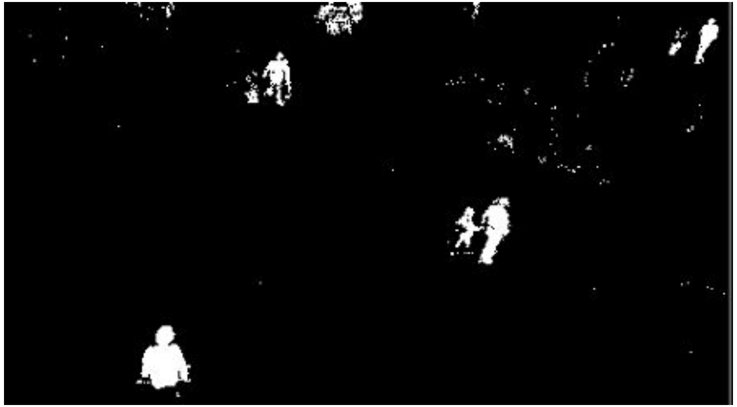
# CDNET2012 - Office(95)

| Input | Python | Our Implementation |
|---|---|---|
|  |  |  |

# CDNET2012 - StreetLight(36)

| Input | Python | Our Implementation |
|---|---|---|
|  |  |  |

# San Pablo video

| Real Image | Out Implementation |
|---|---|
|  |  |

# Problems with the camera movements

It generates noise when the camera is not fixed.

# Observations

- When the camera have little movements this generates noise in the result.
- We obtain better results comparing with the implementation in python but no better than the results of the autor.

# Conclusions

- We implemented an improvement to obtain the LSBP of a frame using threads and cuda.
- Compare with the implementation in python, our implementation in c++ reduce the time per frame in a proportion of 6:1 using threads and 11:1 using cuda.
- We reduce noise by identifying and correcting errors in the python code.

# References

[1] Lili Guo, Dan Xu, Zhenping Qiang, “Background Subtraction using Local SVD Binary Pattern”, 2016.

[2] M. Hofmann, P. Tiefenbacher, and G. Rigoll, “Background segmentation with feedback: The pixel-based adaptive segmenter”.In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pages 38 – 43, Providence, RI, United states, 2012.