

MOVING OBJECT DETECTION AND TRACKING USING DYNAMIC BACKGROUND AND FOREGROUND SEPARATION FOR THE PURPOSE OF TRAFFIC ANALYSIS ON MOBILE DEVICES

Anton Prokopov

BSc student of Computer Science Curriculum University of Tartu
Supervised by Assoc. Prof. Gholamreza Anbarjafari and Priit Salumaa

Introduction

This poster presents the traffic analysis system for mobile devices developed as bachelor thesis with support of Mooncascade company. Aim of this system is to detect and track cars only using camera of Android device. Traffic analysis is performed in a way that the system counts moving cars and tells, if the street is busy or not depending on the amount of cars moving within the street over a period of time. The implementation has been done firstly on a computer using C++ OpenCV library followed by translating the code for being executable on Android device.

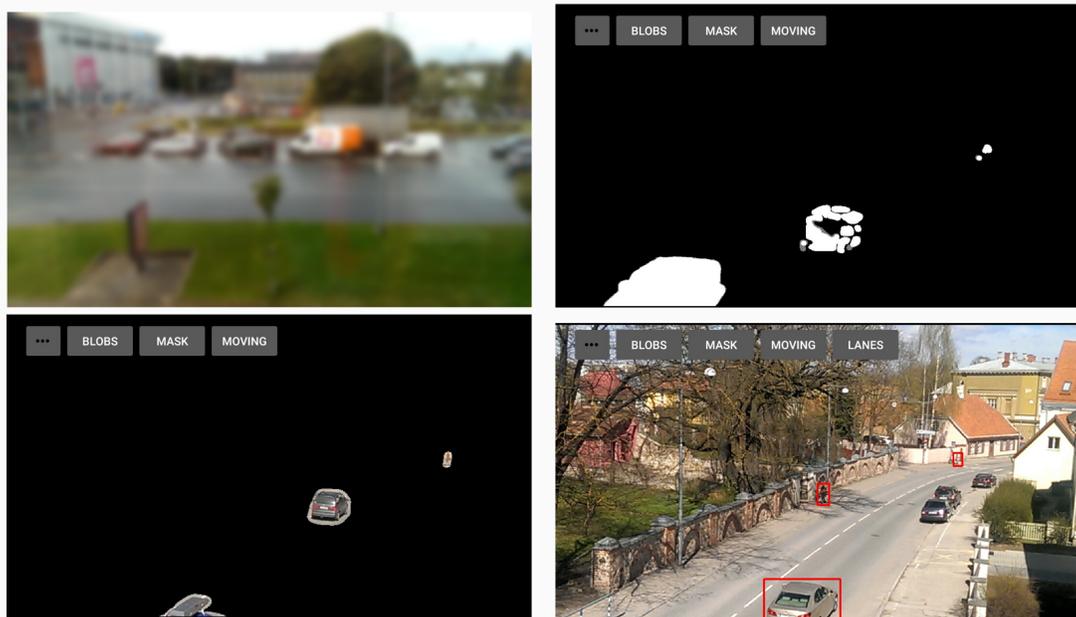
Challenges

In order to accomplish our aims the following tasks need to be conducted properly:

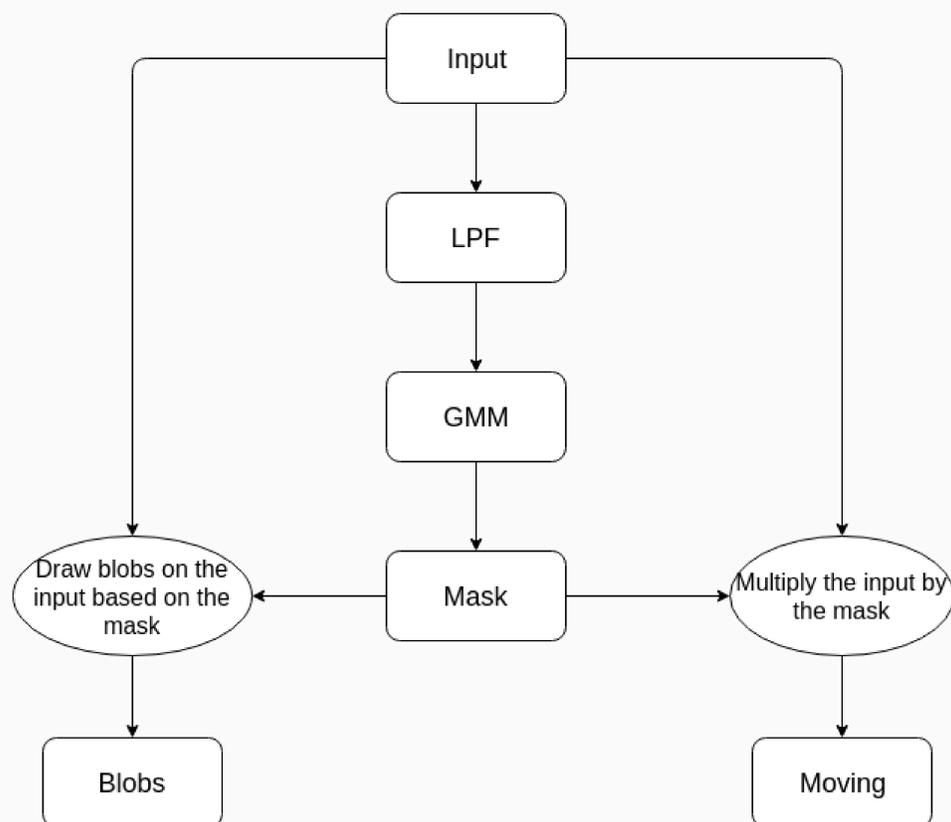
1. Camera that captures the scene is not always static, which means that not only changes caused by moving objects have to be captured, but also changes in the background caused by camera movement have to be discarded. Thus, an algorithm that determines, **what is an actually moving object and what are the changes caused by the camera movement**, is required.
2. Traffic analysis, obviously, can be performed only outside, which means we need to consider **weather changes** as an influencing factor. For example, rain can cause additional noise, wind may perform movement in the objects such as trees and bushes that we would like to recognize as background.
3. **Light conditions** might as well affect object detection mechanism, so it has to be taken into account too.
4. **Reflections and shadows** can cause additional problems to the proposed algorithm due to the fact that they might be falsely considered as moving objects. Improving the detection rate means discarding the changes caused by shadows and reflections.
5. Real-time detection means **high computational abilities**. In our case it means the algorithm needs to be very well optimized as it is supposed to be run on a raspberry pi or an Android device. These devices have their **hardware limitations**, so the algorithm has to correspond to them.

Results

Every step of the algorithm is illustrated in here. **Top left** image is the output of low-pass filter. **Top right** is generated by GMM mask **Bottom left** is a result of subtraction of the mask from the original image - only moving objects are visible. **Bottom right** represents final step, when blobs are drawn around moving objects.



Algorithm



Methodology

When video sequence from the camera is obtained, it is passed to the algorithm frame by frame as an input.

Next step is to apply **low-pass filter (LPF)**. Gaussian low pass-filter is used in case of this algorithm to blur the image and thus reduce small unwanted changes such as movement in the trees and bushes.

Gaussian Mixture Model (GMM) is the next step. It generates a probabilistic model that makes it possible to determine, based on pixel's intensity, which pixels most likely belong to the background. Other pixels then considered to be foreground. Shadows are also detected by GMM and stored in a separate layer, thus easily discarded. **Generated model (mask)** is simply a black-and-white image, where white represents foreground and black - background.

With some additional improvements on the mask, such as applying **erosion followed by dilation** to eliminate unwanted changes, quite accurate foreground is made for further usage.

The final step is to use separated foreground to **find contours** of moving objects and draw rectangles (**blobs**) around them to highlight the area of every moving object. After we have found the moving objects, the last thing to do is to count them.

Conclusion

The recognition has high accuracy in different scenes with different camera angles. Even though the background is complex and might contain movement, the algorithm detects only desired movement, leaving all the trees and bushes as part of the background. Different light conditions do not affect the algorithm's accuracy as well. Real-time recognition is not achieved.

Info