ABSTRACT

This project has been realized as test for our practical comprehension of the subject of “Image Processing and Computer Vision”. We managed to find an argument that could be also useful for the society where we live. For this reasons we chose to realize a system able to detect the number of bicycles passed in a particular street or bike trail.

This system, opportunely tuned, could be a powerful tool in municipalities’ hand interested in sustainable motion, to detect which streets of the city are strongly used by bicycles, and so maybe to find the correct places where build a new bike lane.

In fact in our cities, it’s easy to find video detection systems dedicated to motorized vehicles with the purpose of access regulation or traffic control.

Generally the cameras are combined with underground coils to signal the presence of a new car in the street. More difficult is to find such systems dedicated to bicycles or pedestrian, because of the absence of pollution they produce, their smaller volume, and the impossibility to gain money for them with tickets since they don’t have plates.

This project aims to provide this functionality in a way that avoids the needs of invasive installation of coils in the streets or zones where we want to control the flux.

INTRODUCTION

The practical functionality of this system is to detect the number of bicycles that overcome a virtual line positioned on the street (image).   
Many algorithm provide this capacity: some of them are highly accurate but also complex, and they need a training phase and additional processing when the place changes.  
Others, because of the higher simplicity, are more flexible in spite of a minor number of information extracted from the video.

Initially we have implemented the simplest change detection algorithm: the two-frame difference. We have chosen this instead of the three-frame one because the big limitation of this system is due to the foreground aperture, which is more present in the three-frame difference algorithm.

Moreover, the major presence of ghosting in the two-frame algorithm don’t add too many troubles, rather it has been used to discriminate better motor bikes, that generally are faster, from bikes.

LOCATION

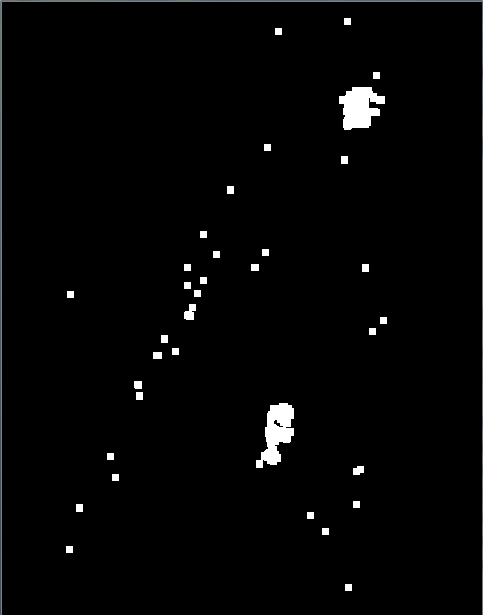
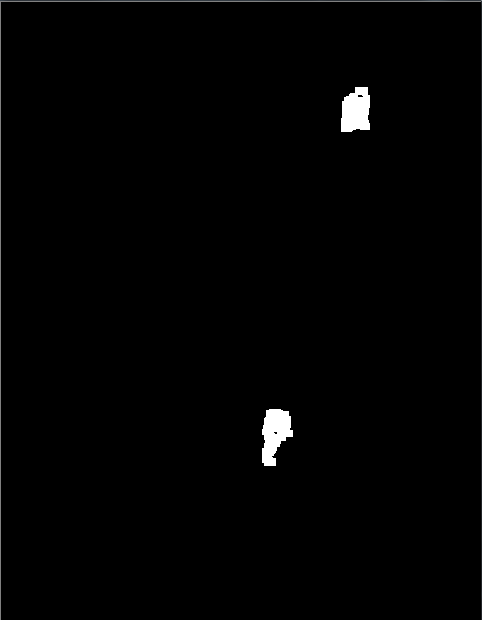


The choice of the location where place the cameras is important for the correct functionality of the system. A too high position reduce the size of the object while a too low position will cause a restrict area of recording. At the same time a lateral position cause a problem when two vehicles are overlapped.

Another problem could be caused by the shadow of the objects, if the street is directly hit by the sun. In this situation could be enough choosing the better angle of the direction of the camera.  
The place used for our test video is on the top of a medieval tower in Via San Vitale at Bologna, but also an “easier to find” one could be possible in general, as on the top of a lamp post, a semaphore, etc..

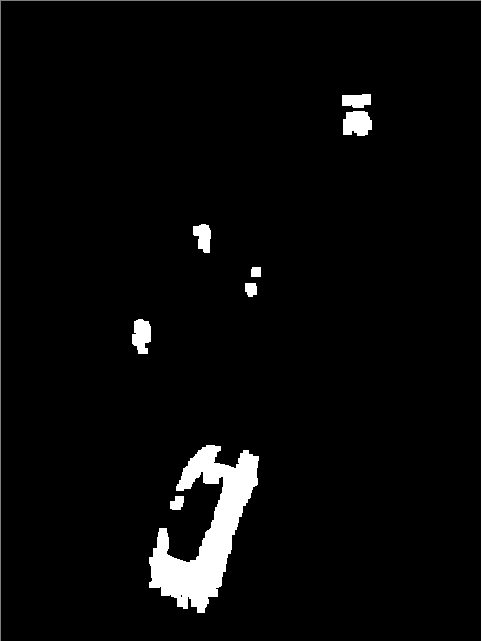
TWO FRAME DIFFERENCE

The two frame difference is a basic change detection algorithm. Its principle is to detect movement of objects by comparing two consecutive frame, and signaling those pixel which values have changed more than the threshold.   
An high value of the threshold will produce no noise but the object will appear fragmented due to less pixel “activated”. On the contrary, a low threshold will cause too much noise and so false “movement” detections on the virtual line.  
We found a good solution using a quite low threshold (a pixel is signaled when the distance from the frame before is more than 3 units) and after that computing the logical operation “AND” between this result frame and the fifths before. This will result in a delay on the detection of a new object in the line, due to the lag of five frame (150ms) before that all the five pixel of the further point of the object in the five frame will become “1”.



A problem of the two frame difference is the ghosting, due to the activation both of the new position of the moving object and the old one. In this contest this is not appeared as a problem; on the contrary this it’s result on an advantage for the discrimination of the motor bikes from the bicycles. This because the ghosting is proportional to the velocity of the object, and generally motor bikes are faster than bikes, and so the area of the detected object will be greater if this is a motor bike.

Another problem of this category of algorithms is the foreground aperture, namely the non activation of pixel in movement caused by the uniformity of the surface of that object. This error appear often on the roof of cars or on the body of a tram. Luckily, this error is not present in bicycles or motor bike, objects at which we are more interested in this project.



OBJECT TRIGGER

A necessity of the software is to include a “system” that signal when a new object appear on the virtual line, and that do it only once. This because a multiple warning for the same object bring about the need of a machine understanding that the object detected is the same to avoid counting it more than once.

This is partially obtained by recording which pixels of the “virtual line” were excited in the previous frame. In particular we have implemented a function, called “AroundExcitation” (mostriamo il codice?) that looks in both the contour of the excited pixel of the line in the actual frame and in the precedent frame, before counting it as excited. This imply that a connected (or almost connected) group of pixel in the line are signaled only once, and that if this group of pixel was already active in the precedent frame no signal occurred.

In practice, we scan the line by doing an AND operation of the result of the two-frame difference algorithm (with the noise removed) and a mask where only the virtual line is present as white.

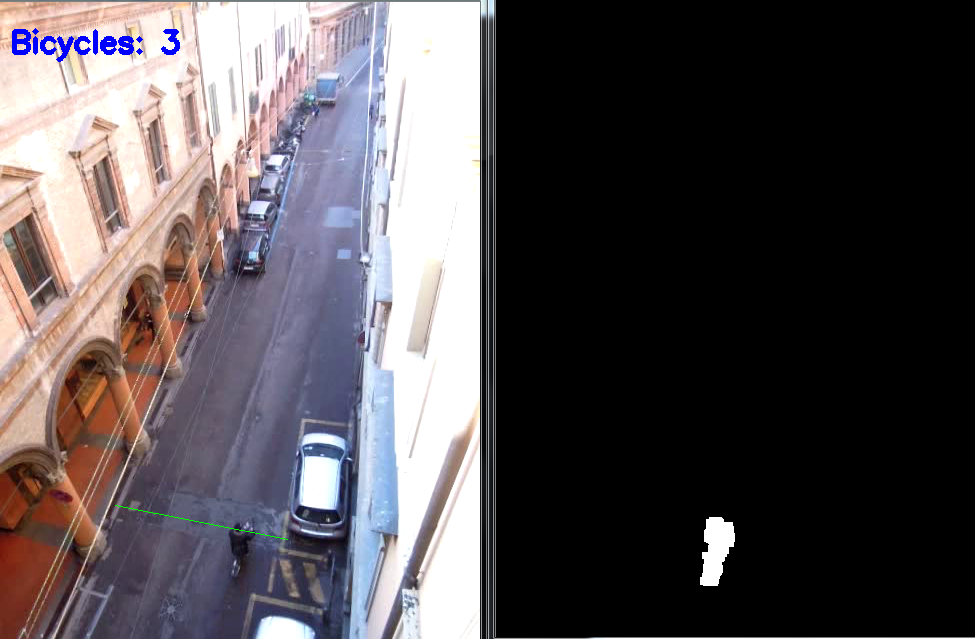
Apart all, the general policy adopted for this trigger is to generate a false signal rather than leave the possibility to lose the transit of an object. It will be then task of the functions that will analyze it to understand if it was an interesting object or a false positive.

In fact, cars or uniform object like trams, will probably activate the trigger more than once, and this introduce also the problem of recognize cars and don’t count them multiple times.

RECONSTRUCTION OF THE OBJECT

Once we receive the call from the software trigger, our task become to reconstruct the object at fault of this call. So, as first step, the software will scan the frame results from the two frame (completely, not only the line) and set “white” each pixel that touches an active pixel. For optimization reason, it will alternate an downward and an upward scan, until no connected pixels are left.

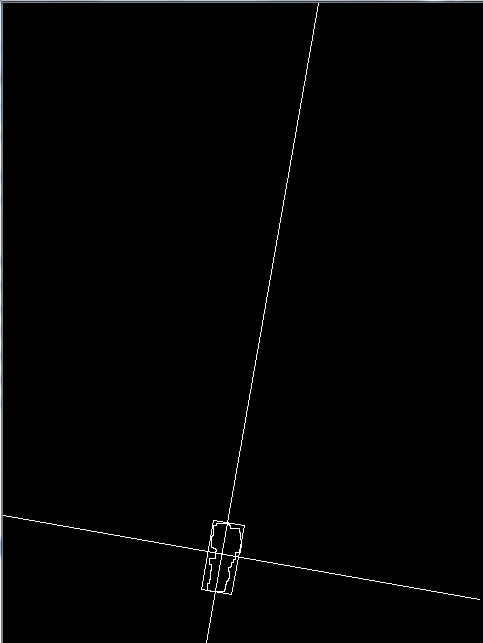
Each time a pixel is set, a counter is incremented: his final result it will be the area.

  
An other operation is computed, that is the calculus of both coordinates of the barycenter, that it will be useful to find the dimensions of the vehicle analyzed.

After that, we scan again the final image, and we set as black all those pixel which are completely surrounded by white pixel, leaving in this way only the perimeter, gaining another interesting parameter, used to find the compactness (Perimeter\*Perimeter/Area).

Finding perimeter gives us an other interesting support, that is avoid to calculate the dimensions, width and length, of the object looking at each pixel of the object, and do that only in those pixel belonged to the boundary.

To find the length we calculate the distance of each pixel from the line parallel to the “virtual line” passing for the barycenter and choosing the bigger (positive) and the smaller (negative); the final value will be the sum of the two absolute value.  
The same needs to be done for the length, but instead of doing with a line parallel to the crossing line, it has be done with a perpendicular line, always passing for the barycenter.  
Now, with some more geometrical calculation, we are able to draw the bounding box of the object related to a direction chosen as the normal direction of the traffic in the street



DISCRIMINATION POLICY

Actually the discrimination policy adopted is “direct” and “in or out”.  
“In or out” it means that we count the object as a bicycle if it has all the controlled parameters in the chosen ranges, which are the area, the compactness and the factor form (length/wide). A possible alternative is to give a weight to each parameter, using it to give a score to the object, and then look if this score is enough to be classified as bicycle.

“Direct” it means that the ranges are chosen for a determined line, and they need to be rewritten if this lane changes.   
An other possibility is to chose ranges as function of the line chosen. But this strategy will not so useful, because if the place changes, this function needs to be completely rewritten, operation more complex and more time consuming than change manually the parameter.

https://www.youtube.com/watch?v=zRRuqOu6cgM