National Research University Higher School of Economics
Faculty of Business Informatics
School of Software Engineering
Software Management Department

AN APPLICATION FOR DYNAMIC OBJECT IDENTIFICATION BASED ON LUCAS-KANADE ALGORITHM

Student:	Kostenko Dmitry
Group:	472SE
Argument Consultant:	Prof. Ivan. M. Gostev, PhD
Style and Language Consultant:	Tatiana A. Stepantsova

Moscow
2013

Abstract

В данной статье описывается подход к обнаружению и подсчету транспортных средств на атодорогах. Он основан на дифференциальном методе вычисления оптического потока, предложенном Лукасом и Канаде. Отличие данного метода от других состоит в том, что нет необходимости подготавливать модель фона. Так же описываются достоинства и недостатки данного подхода.

Содержание

1 Introduction **4**

2 Related work **5**

3 Problem statement **6**

4 Algorithm **7**

4.1 Allocation of foreground objects 7

4.2 Selection of connected regions 9

4.3 Vehicle tracking and counting 11

5 Conclusion **12**

6 Bibliography **13**

1 Introduction

Nowadays it is possible to notice high grow of vehicles in all Russian cities. According to DPS data annual growth of cars number is 110 - 120 thousand. As a result the significance of traffic conjunction problem increases. It causes higher fuel usage One of the decisions of the problem is installation of Intelligent Transportation System (ITS). ITS ranges from simple traffic light control systems to systems which register velocity of vehicles flow, control traffic flow and recognition of the violations. Such systems can perform the following tasks:

1. ensuring maximum traffic capacity;
2. reducing road accidents and monitoring human factor;
3. collecting information about traffic jams from the vehicle flow and informing its participants;
4. environment protection as a result of real-time monitoring of road situation and well timed decisions making.

ITS can contain various sensors from heat sensors to super-sound ones. Manual processing of a significant amount of data which is received from the sensors is not applicable in the real-life situations. As a consequence, a vital necessity of automation of the process and decision making, based the information gathered during this process, arises. Automatic vehicle detection in this video monitoring is a complex objective of computer vision. On of the tasks of such a system is to count vehicles on the highway. In its turn, it is divided into subtasks of computer vision, such as foreground retrieving (vehicles) and tracking in the next frames. This paper offers an overview of an approach which is related to automatic tracking of moving vehicles. The only source of data used by this approach is a camera video recording. The first chapter of the present graduation paper offers an overview of existing solution in the field of video monitoring. Problem statement and requirements to the algorithm under development are covered in the second chapter. The third chapter deals with description of methods and algorithms. And finally, the brief exploration of expected results will be introduced.

2 Related work

In this section a short review of existing highway video surveillance solutions is presented. There is only one full ITS architecture, developed by the transportation department of the US. It focuses on creating a unified information environment, that connects automobiles, road equipment, dispatch centres, and data centres around the country. This system has been patented by the US government.

There are several analogues around the world that are based on the US system. In 2012 a Russian ITS was rolled out in Moscow. All of these systems are commercial and there are no ways of analysing there methods. However, there are also several open-source projects of smaller scale. As has been mentioned before, one of the main tasks of such systems is counting vehicles on the highway.

Most of the times the vehicle counting algorithm consists of the following steps:

1. create the background model;
2. subtract this model from the current image and obtain all moving objects of the foreground (vehicles);
3. count the amount of vehicles that has crossed the line of interest;
4. update the model of the background.

Examples of such an approach: ****

Our work uses a different approach. We suggest an alternative method for counting vehicles without the pre-modelled background. It is based on the differences sum of the neighbouring frames of a video sequence and highlighting the connected areas.

3 Problem statement

In this chapter we will state the set of requirements for the algorithm. As a formality we will state several definitions first. As any video is a sequence of frames, every moment in time we have one frame called the current frame. If the current frame is defined as i, the previous frame will be defined as (i-1). We will consider all the moving objects such as vehicles, trees, humans as the foreground. Whether the object is considered as a moving one depends on the camera properties and the environment. So we define moving objects as the ones that shift their position in the current frame, relatively to previous frame. When the camera moves, or the object in its focus, it leads to the image changes. The description of the movements which can be seen is called the optical flow. There are several methods for calculating this flow. Lucas and Kanade developed one of them, using differential approach. It will be described later in this paper. Now let us move to the problem statement. One of the most important problems of video surveillance is identifying the object of interest and following its trajectory through the sequence of the following frames. This problem can be partitioned into several sub-problems. In the beginning it is necessary to find the objects of interest, which can also be called the foreground objects. Finding these objects involves separation of the foreground from the background of the image. As a result we get a binary map of the moving objects. After that we need to separate vehicles from other moving objects such as trees or humans. For doing so we limit the area in which we will track movement. A good choice will be a well-observed part of a highway. Then we need to track the changes in objects of interest coordinates in the following frames.

To calculate the optical flow we need to make several assumptions:

1. An image is a continuous function of two variables. This will let us use mathematical analysis, and also carry out mathematical operations with the image;

2. The brightness of an object remains constant in a short period of time. Without this assumption we will not be able to track the object;

3. The location of the object of interest will not change a lot relatively to the previous frame. This is due to the fact that in real life objects do not move with the speed of light.

Сформулируем задачу: посчитать количество автомобилей, пересекающих область интереса.

Теперь мы можем сформулировать требования к алгоритму: 1) Работать без каких-либо предварительных данных об автодороге, 2) Обработка потока данных в реальном времени 3) Не должен требовать высоких вычислительных мощностей. Минимальные требования к оборудованию будут предложены в техническом задании, на стадии написания данной бумаги их невозможно сформулировать.

Перейдем к изложению алгоритма.

4 Algorithm

In the following chapter an approach to the task of vehicles on motor ways detection and counting will be revealed.

As a video is a frame sequence, at every moment only one frame is available. In the beginning, it is necessary to eliminate a noise of each frame. Gauss filter must be applied in order to perform this. The information about object color seems to be not in use in further. Therefore it will be appropriate to transform the image from RGB color space to grayscale. Also, for the sake of increasing the processing performance it is necessary to reduce the size of the image. But it is possible only if the camera allows to do so. For instance, if the camera shoots with a resolution of more than 640 by 480 pixels, every frame of the video stream can be reduced to this resolution to increase the processing performance of every frame. After all of the images pre-processing procedures, the foreground objects are detected.

4.1 Allocation of foreground objects

The primary goal of the method, described in this section, is retrieving the foreground objects location. The traditional algorithms of retrieving the foreground objects location are based on pixel-by-pixel frame difference method. But this method also has several disadvantages. This method's application might lead to the enormous amount of isolated areas (Fig. **??**).

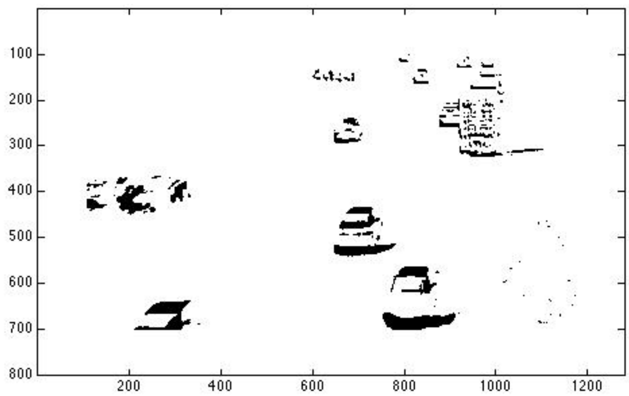


Рис. 1: Pixel-by-pixel frame difference method.

For this reason, this method must be modified to avoid this disadvantage and to make these areas connected with each other. At first, each frame has to be divided into blocks which do not overlap. The optimal size of a block is found empirically and depends on characteristics of the camera and the environment. That is why in the beginning the standard size will be assigned to each block - 3x3 pixels. In Fig. 2.

After that the previous frame is subtracted from the current frame. Then every pixel block of the result is filtered in the following way: if the amount of foreground objects pixels

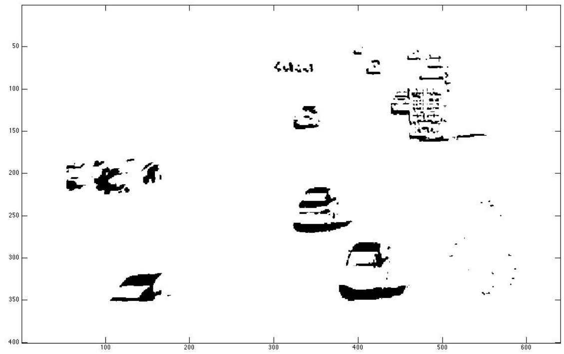


Рис. 2: *The result.*

in every block is more than some pre-defined threshold, this block is considered to be the foreground otherwise, it is considered to be the background

As a result we get the binary image of the foreground objects. But this image is not going to be perfect, it will definitely have noise. For that reason the image must be processed using morphological operations. These operations are the following: erosion and dilation.

It is necessary to consider that the method of subtracting the previous frame from the current frame gives good response near the moving object border. However, inside the objects borders the response is NOT SO GOOD. If the vehicle is relatively large, the probability of its recognition as an object of the foreground is LITTLE.

To get rid of this effect let us consider two definitions: sort-term model of the foreground and long-term model of the foreground.

The short-term model is the model which we get, after applying the steps stated above. The long-term model is the pixel-by-pixel sum of N previous short-term foreground models, where N is a positive integer and more than 1. N is found empirically and depends on characteristics of the camera and the environment.

a) b) c) d) e)

Рис. 3: Зависимость сигнала от шума для данных.

As the objects that we want to track are moving, calculating the long-term model of the foreground we get the object regions, which is more connected. As can be seen from Fig.4 the object region can be separated into several regions. To reduce the amount of these regions we use the dilation operation with pre-defined window.

Рис. 4: *Object, separated into several regions.*

4.2 Selection of connected regions

After creating the binary map of the foreground objects, it is necessary to give every object a certain number. A straightforward segmentation algorithm is used. Suppose, we obtained the binary image. In our case white pixels stand for the background, and black pixels stand for the potential vehicles. The mapping from the original image to the segmented image is mono-semantic. We will call an array of pixels connected if and only if each pixel has at least one neighbouring pixel belonging to this array. There are two ways for pixels to be connected:

1. 4 connected (Fig. 5 (a)) - the neighbours of the pixel are only the following pixels: the one above, the one below, the one to the right and the one to the left of it;

2. 8 connected (Fig. 5 (b)) - the neighbours of the pixel are the pixels above, below, to the right, to the left, and the diagonal pixels.

	1	
2	*	3
	4	

a)

1	2	3
4	*	5
6	7	8

b)

Рис. 5: (a) 4-connected, (b) 8-connected

In our approach the 8 connected way is used, because it gives a significant raise in precision. There are generally known two methods of image marking: recursive and iterative. The shortcomings of the recursive approach are the following:

1. a great amount of memory is needed for it;

2. it is rather slow, compared to the iterative method.

The iterative method is an alternative to the recursive approach. It can be often met in literature by the name of: "Sequential scanning algorithm". Let us present it in detailed steps:

1. To be specific, we start processing the pixels of the image from the left-top corner, going from the top to the bottom and from the left to the right.

2. While cycling through the pixels of the image we ignore the background pixels, and mark the current pixel with the color of the top-left neighbouring pixel.

Идея данного алгоритма основана на использовании уголка — ABC-маски, которая показана on Fig. 6. Проход по изображению данной маской осуществляется слева-направо и сверху-вниз. Считается, что за границей изображения объектов нет, поэтому, если туда попадают B или C — это требует дополнительной проверки при сканировании. На рисунке 4 изображены 5 возможных позиций маски на изображении. Рассмотрим их.

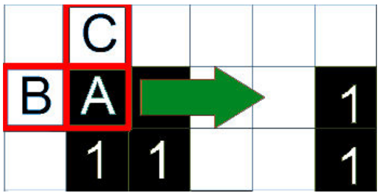


Рис. 6: ABS-mask.

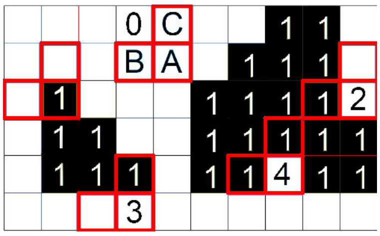


Рис. 7: 5 cases of ABS-mask.

On Fig. 7 we can see 5 cases of ABS-mask:

1. Позиция под номером 0, когда не размечены все три компонента маски — в этом случае мы просто пропускаем пиксель.

2. Позиция под номером 1, когда помечен только элемент A — в этом случае мы говорим о создании нового объекта — новый номер.

3. Позиция под номером 2, когда помечен элемент элемент B — в этом случае мы помечаем текущий пиксель A меткой, расположенной в B.

4. Позиция под номером 3, когда помечен элемент элемент C — в этом случае мы помечаем текущий пиксель A меткой, расположенной в C.

5. Позиция под номером 4, тогда мы говорим о том, что метки (номера объектов) B и C связаны — то есть эквивалентны и пиксель A может быть помечен либо как B либо как C. В некоторых реализация составляют граф эквивалентности таких меток, с последующим его разбором, однако на мой взгляд в это нет необходимости. Мы будем делать так — в том случае, если B не равно C то перенумеруем все уже обработанные пиксели помеченные как C в метку B. Но об этом в самом конце.

After processing the image, we need to go through the array of pixels again and do the marking procedure with the consideration of areas equality, As a result, an image is obtained, where each connected area is marked with its own number.

Понятно, что при таком порядке обхода у текущего рассматриваемого пикселя верхний и левый сосед уже должны быть размечены.

Now we cut out the areas which can not be classified as vehicles. To do so, we need to count the amount of pixels in each area. The areas in which the amount of pixels exceeds a certain threshold are eliminated. This threshold is found empirically and will depend on the camera characteristics and the environment conditions. At the time of writing we can not tell exactly what the threshold is going to be. That is why, for now we will just ignore it and consider all of the connected areas to be the vehicles that we are looking for.

4.3 Vehicle tracking and counting

После того, как мы получили связные области (которые являются транспортными средствами) на текущем кадре, нам необходимо понять, какие из этих объектов новые, а какие уже существовали на прошлом кадре.

Будем считать, что положение связной области определяется, как координаты центра связной области. Такой центр вычисляется как среднее значение всех x-координат и y-координат пикселей, принадлежащих области.

Т.е. задача формулируется таким образом: необходимо сопоставить текущие связные области с такими же на предыдущих кадрах или инициализировать новые.

Ранее мы сделали несколько ключевых предположений. Одним из них было продположение о том, что атвомобиль на текущем кадре сдинулся на небольшое расстояние отоносительно себя на предыдущем кадре. Таким образом, храня положения цетров связных областей мы можем найти ближаший центр данной области на текущем кадре к этой же области на предыдущем кадре, используя подход, предложенный Лукасом и Канаде.

Основное идеей данного метода является нахождение ближайшей точки к данной путем навешивания весов на точки. Подробнее алгоритм описан здесь: ==============

Приведем содержание нашего подхода и общую схему алгоритма.

1) Перевести изображение в серое 2) обработать изображение фильтром гауса (против шумов) 3) вычесть из текущего изображения предыдущее 4) наложить друг на друга n предыдущих разностей (для получения транспортного средства) Делать это каждые 5 кадров 5) выделить связные области, которые и будут предположительно транспортными средствами 6) сопоставить текущия связные области с такими же на предыыдущих кадрах методом Лукаса и Канаде или инициализировать новые 7) если такая связна область пересекает линию интереса, то прибавляем 1 к счетчику

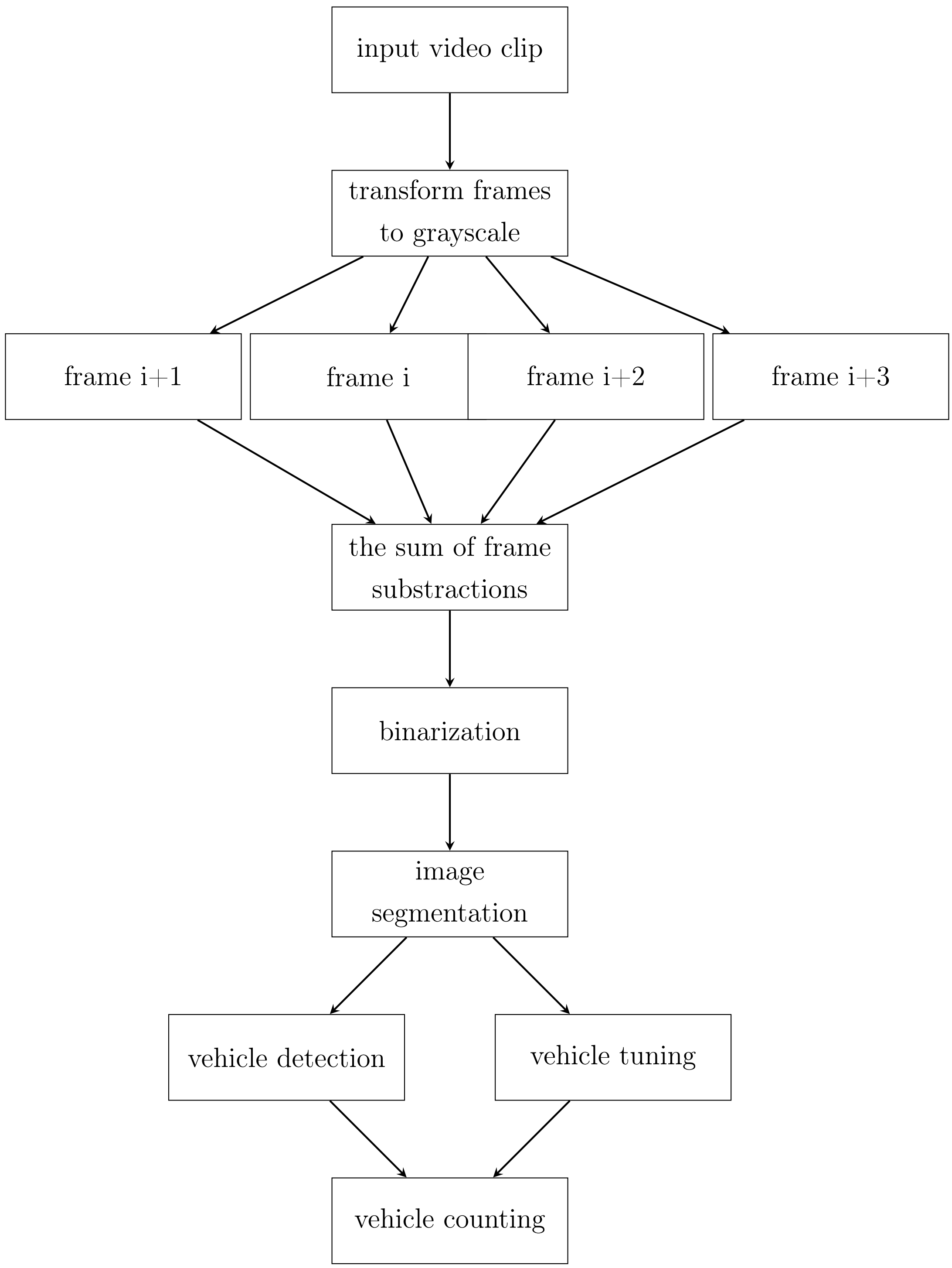
5 Conclusion

В данной работе мы предложили еще один подход к подсчету автомобилей на автодороге. Ключевыми моментами являются: 1) исключение необходимости постоянное хранить и модифицировать модель фона, 2) отслеживание транспортных средств методом Лукаса и Канаде.

Помимо положитлеьных сторон у нашего подхода есть свои существунные недостатки. Например, мы предпологаем, что камера является неподвижной. Хотя в реальных условиях камера может быть подвержена движению, от порывов ветра. Или на объектив камеры могут попадать капельки дождя, которые могут быть зафиксированы, как объекты переднего плана. Скорее всего борьбу с некоторыми из них следует вынести за рамки программы и бороться с ними на аппаратном уровне.

6 Bibliography

1) Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways Mrs. P.M.Daigavane † and Dr. P.R.Bajaj ††, S. D. College of Engineering, M.S., INDIA


input video clip
transform frames to grayscale
frame i+1
frame i
frame i+2
frame i+3
the sum of frame substractions
binarization
image segmentation
vehicle detection
vehicle tuning
vehicle counting