# Real-Time High Density People Counter using Morphological Tools

Antonio Albiol, Valery Naranjo
Departamento de Comunicaciones, Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia, SPAIN
aalbiol,vnaranjo@dcom.upv.es

Inmaculada Mora
Departamento de Tecnologías de las Comunicaciones
EPS-Universidad Carlos III de Madrid
Avda. Universidad 30, 28911 Leganés (Madrid), SPAIN
inmoji@tsc.uc3m.es

## Abstract

*This paper deals with an application of image sequence analysis. In particular, it addresses the problem of determining the number of people who get into and out of a train carriage when it's crowded and background and/or illumination might change. The proposed system analyses image sequences and processes them using an algorithm based on the use of several morphological tools and optical flow motion estimation.*

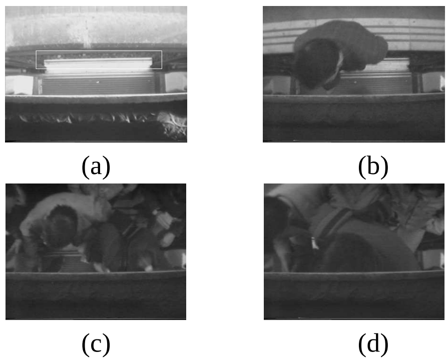


**Figure 1. Example of full camera views. a): With nobody b) Isolated person. c) and d): Crowded situations.**

## 1 Introduction

The purpose of the work presented here was to provide a tool to the Spanish railway company capable of determining the number of people getting into and out of a train. Our system analyses images of a door, doing the surveillance from a zenital position above the train door. The camera is placed in the door mechanism box, and the acquired images are monochrome. The system described below requires neither special illumination nor markings to be deployed. An example of the camera view is depicted in figure 1. The placement of the camera above the door has the significant advantage that no occlusion occurs.

Varying conditions of illumination and background can be found at different stations, time of the day and weather conditions (there are underground and surface stations). This fact denies the use of background substraction techniques. Figure 1 shows sample frames of people passing with different densities.

Since a person usually takes more than one frame to cross the door, some form of time memory must be included in the algorithm. In order to reduce the memory storage requirements and to allow non-causal processing, the only thing that we perform each time we acquire a new frame is to store certain lines of the frame onto what we have called *stacks of lines*, which will be described with more detail in the next section. These stacks of lines contain all the required information to count the people and determine the direction of passing. The counting process actually starts after the doors are closed.

Then, our algorithm can be at different states:

1. When doors are closed the camera lens is covered and the acquired image is completely black. We will assume that this is the initial state, that we have called *Closed-Doors State*. This state is abandoned when doors begin to open.

2. When the doors start to open, the storage of certain

lines of each frame starts. We call this, *Acquisition State*. We move to the next state when the doors end to close.

3. Right after doors have closed, we enter the *Counting State* and begin the actual counting. To do so, the stacks of lines are analysed in order to determine the number of people between at the last station. After finishing the counting we return to the first state.

The time required to process the stacks is considerably shorter than the time needed to acquire them, and much shorter too than the time needed for the train to reach the next station. In this sense we claim that our process is real-time. Strictly speaking we could say that it is a *delayed-real-time* algorithm.

The rest of the paper is organized as follows. In section 2 we describe how to build the stacks of lines. Section 3 gives an overview of the whole processing and is divided into three subsections corresponding to the main steps of the counting algorithm: people-background distinction, segmentation of individual persons and determination of direction. To conclude the paper we will give some results and conclusions.

## 2 Image Acquisition

During the acquisition state, three (horizontal) lines of every frame are stored image onto a separate stack each. These lines are shown in figure 2-b. We will call the first line (the topmost one in the figure) the *black line* and it corresponds to the platform-carriage gap[1]. The intermediate line, we will call it the *gradient line* and should coincide with the upper edge of the train step. The bottom line is located on the step and will be termed *white line*, because it is brighter than the *black* one.


**Figure 2. Detail of figure 1-a) to show the location of the black, gradient and white lines used to build the stacks.**

Black and white lines simply contain the gray-level of the original image. For the gradient line, we compute the vertical gradient, and this is the value that we store. Each of these three lines is stacked on separate images where each row corresponds to a frame. At the end we will have three stacks: one for black lines, one for gradient lines and one for white lines, which we will denote respectively as black, gradient and white stacks. In figure 3 we can see an example of stacks. The vertical axis corresponds to time (increasing downwards) while the horizontal axis is the horizontal dimension of the original images. Since the camera is fixed to the train, the position of the lines in the images does not change from station to station, and is configured (automatically) off-line during the installation.

[1]In surface stations it is not necessary black, but for convenience we will always call it this way.

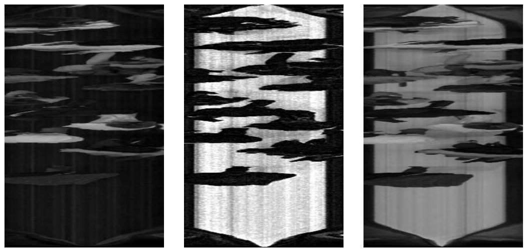


**Figure 3. Example of stacks. From left to right: black, gradient and white stack.**

Stacks like these will be the input to the actual processing algorithm. From the stacks it is quite obvious for a human observer (after a certain training) to see how many people have passed through the door. It suffices to count the number of blobs. At the top and bottom of the stacks a characteristic pattern due to opening and closing of doors can also be seen. In the following section we will explain how to do the counting from the stacks automatically.

## 3 Processing Algorithm

Once we have the stacks of lines, we must analyse them in order to find out how many people they contain. The processing is performed in three different stages:

1. Presence detection: The aim at this point is to create a binary image (with the same size as the stacks) that indicates when and where the *border line* is hidden by someone or something passing. The *border line* is the position of the frame that must be crossed in order to make a count and corresponds to the position of the gradient line.

2. Segmentation: its purpose is to segment the presence image into individual prints each one corresponding to a single person.

3. Direction estimation: After segmentation, we will have an *image* which will be null when nobody is passing and will contain a different label for each portion of the stacks corresponding to a different person. The objective is now to estimate the direction of passing at each of these *labels* of the stacks of lines.

### 3.1 Presence Detection

After examination of the three stacks of lines we decided to use the gradient stack to make the presence detection. This was because the gradient stack, unlike the other two stacks, bright normally means nobody and dark the presence of someone. The other two stacks show the actual gray-level and this can be very different for each person. In other words, we obtain a binary image, with the size same size of the stack, indicating the presence of a person at the border line.

The reason why gradient is low when someone is passing is because the lack of contrasted elements on a person image (see figure 1). However, the position of the gradient line was selected because it contains a high gradient. Nevertheless, a person may contain contrasted elements that may be at the gradient line position at some instant. See for instance the topmost blob of the gradient stack in figure 3. Fortunately when this happens, this situation lasts only for a few frames since people are moving. This property will be exploited to perform a prefiltering before thresholding.

The process for presence detection can be summarized as:

1. Pre-filtering: the purpose is to eliminate transients with too short duration that breaks the person print, given a high gradient value inside it. Morphological filter are very useful at this point because, unlike other filters like linear or median, they operate selectively on bright or dark elements of the image. However, the use of such filters in space-time images must be very careful, in order to obtain a meaningful filtering. In particular, we have used the morphological filter named opening by reconstruction [1]. We could enunciate, its action as 'remove all bright spots that do not have a minimum duration (height on the stacks) at any horizontal position'.

2. Determination of threshold: Due to illumination variations the needed threshold is different for each station. We find out the threshold by finding the rows of the stacks where we are sure that nobody is present.

3. Segmentation People/Background: Thresholding the filtered gradient stack yields a binary image indicating when and where was people passing.

   The result of the presence detection of the stacks in figure 3 is shown in figure 4

### 3.2 People segmentation

In order to count the people, we must segment the presence mask into individual prints due to a single person. The situation would be straightforward if no contact between persons could happen. Since this is not the case we have

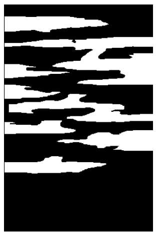


**Figure 4. Presence detection result of stacks in figure 3.**

to design techniques for segmentation of the prints corresponding to each person. For explaining the separation techniques that we have used, we have prepared a synthetic image, shown in figure 5(a), to show how the different separation techniques work. That synthetic image contains the most important kind of problems encountered in our analysis. From top to bottom we can see the following prints:

1. An isolated person, passing quickly. This can be noticed by the short vertical size of the print.

2. Two people passing side by side.

3. A single person passing exactly under the camera. The camera height is small, so wide angle lenses are used. In this case the print can be as wide as that of two people if the person happens to pass exactly under the camera.

4. Two people passing one immediately after the other, leaving no frame (row of the stack) of gap between them.

5. One person. Print with branches. This is a very common situation. Legs, arms, bags, etc. appear normally as narrow branches.

6. One person passing slowly. Irregular print. One person passing slowly takes many frames to completely cross the door. This causes a large vertical size of the print. The shape of real prints has irregularities as those shown.

The separation process is based on Width Function of the presence mask, which calculates an image of gray levels where each horizontal segment has a gray level indicating its width (see figure 5-b).

The separation process is divided into the following steps:

- Side contacts detection/separation: When a side contact happens a sudden increase in the width occurs. When a side contact finishes a sudden decrease of the width occurs. See figure 5-b, blob 2. Then, if we compute the vertical gradient of the width function an indication of side contacts can be easily got. Once a contact is detected, we proceed to separate the individual (side by side) prints by introducing artificial background points. Finally a shrinking of the resulting horizontal segments is performed to achieve complete separation (see figure 5-c, blob 2).

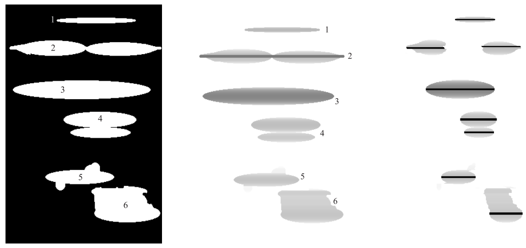
- Markers extraction: after side contact separation, only longitudinal separation remains. The information that we use to perform this separation is that the people prints have an *approximately* convex shape. That would suggest the use of the maxima of the width function as markers. In order to obtain only one marker for print we have computed the contrast of each maximum of the Width Function [2].The contrast of a maximum of the width function has the following meanings:
  - If there is a single maximum of width per blob, the contrast is the same as the width at the widest point.
  - If there are more than one maxima of width per blob (see figure 6), the contrast will coincide with the width for the highest maximum ($W1$ in figure 6 ) and will be the difference $W2 - W3$ for the other width maxima.
- Segmentation: If we were only interested in finding out the number of people, the process would have ended with the marker extraction. However in order to determine the direction in a robust manner, it is interesting to segment the presence image in order to integrate all the information concerning each person. We have used the watershed segmentation to the negative of the width function [3] using like markers the maxima of the Width Function with high Extinction Value. This splits the blob at the narrowest point between each pair of markers.

### 3.3 Direction Determination

We have used an algorithm based on optical flow [4] to estimate the direction of crossing the door. The speed obtained from the optical flow equation gives one value per pixel of the stack. For each label of the segmentation we compute the weighted average value of speed.

## 4 Summary and Conclusions

We have performed a large number of tests (149 train stops) corresponding to images taken at different times of different days in a railway line near Madrid. The test sequences include both indoor and outdoor stations. No parameter tuning has been made for the whole set of test sequences. The intensity of people passing varies from no one passing at certain stations, to an average intensity of more than one person per second. The results have an average of error of less than 2%.



(a) (b) (c)

**Figure 5. (a) Synthetic image used to explain the different separation techniques. (b) Markers (in black) and image to which we will apply the watershed. (c) Markers and input for watershed.**

## References


[1] P. Salembier and J. Serra, “Flat zones filtering, connected operators and filters by reconstruction,” *IEEE Trans. on Image Processing*, vol. 3, pp. 1153–1160, august 1995.

[2] C. Vachier, *Extraction de caracteristiques, segmentation d’image et morphologie mathematique*. PhD thesis, Ecole Nationale Supériore des Mines, Paris, Décembre 1995.

[3] S. Beucher and F. Meyer, *Matehmatical Morphology in Image Processing*, ch. 12. The morphological Approach to Segmentation: The Watershed Transformation, pp. 433–481. Marcel Dekker Inc., 1993.

[4] A. M. Tekalp, *Digital Video Processing*, ch. 5. Optical Flow Methods, pp. 72–116. Prentice Hall, 1995.

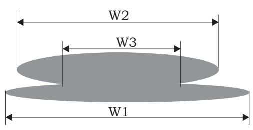




**Figure 6. Widths involved in the contrast of the width function computation.**