Automatic Reader of Recording Strips

V. Delcourt[1], C. Machy[2], C. Mancas-Thillou[2], X. Desurmont[2],
[1]SNCF - 75008 Paris, 45 Rue de Londres, France - vincent.delcourt@sncf.fr
[2]Multitel Research Center – Faculté Polytechnique de Mons , 7000 Mons, Belgium - name@multitel.be,

Abstract

Even if the number of accidents involving the railway system is decreasing due to the technical progress, the statistics are still too high. For instance in 2004 in EU 25, 9309 accidents were reported, including 142 in France [ERA]. For each of these accidents in France, one element can be used as evidence in the eyes of the law: the recording strip and its associated filling-card or the so-called ATESS file recorded by digital Juridical Recording Units (*JRU*) introduced in the mid 80s. The strip contains all the information concerning the journey of the train, speed and time recording and all the driving events (such as emergency breaking). The card features additional information on train's driver, departure/arrival stations, number of trains, etc. These two elements are presently checked manually. The idea of this project is to simplify the procedure and to perform the checking as automatically as possible. This paper then aims at presenting a whole system for the Automatic Read of Recording Strips (*ARRS*).

Introduction

Specific train performance parameters are registered thanks to a system of driving follow-up which is the equivalent of flight recorders in planes. Nowadays, on French railways trains, there are two types of recorders: digital data recorders providing numerical files, and paper recorders, named "Flaman", "Teloc", "Tachro" or "Atec", providing rolled up recording strips *(RS)* which are graphic bands of paper associated with filling-cards *(FC)* filled by drivers and which contain a list of the successive journeys plotted on the recording strip.

Figure 1. Filling card (*FC*)

Figure 2. Part of a recorded strip (*RS*)

The only way to check recording strips for the moment is to do it manually. Two procedures are used by operators: the complete checking, in which each detail of the run is scrutinized, and the partial checking which means that operators only pay attention to the compliance of main security rules. Each month, 44000 recording strips are processed in these conditions, at the rate of five or six minutes per strip, by almost a hundred operators in four centers in France.
The main goal of the project is to convert these numerous recording strips into numerical data comparable with data issued from digital recorders, making use of an automatic reading system. Thus, each file resulting from conversion could undergo a complete checking by AIDA, a post-processing software previously developed by SNCF Innovation and Research Department, and currently used to analyze numerical recordings. Figure 3 shows a typical use of the system.

There are three different types of information plotted by recording pens to convert from strips: speed and time recordings, which are curves, and all the Driving Events *(DE)* which appear as a sequence of various patterns on the recording strip. As far as *FC* are concerned, dates, numbers of trains, departure stations, arrival stations or even the name of the drivers are examples of which information has to be converted.

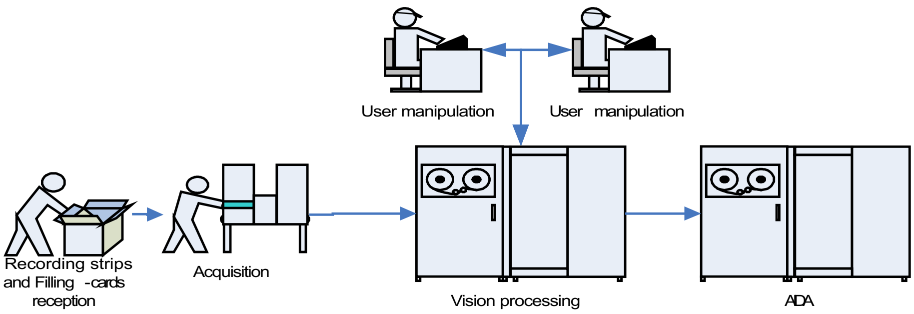

Figure 3.Global presentation of the ARSS system.

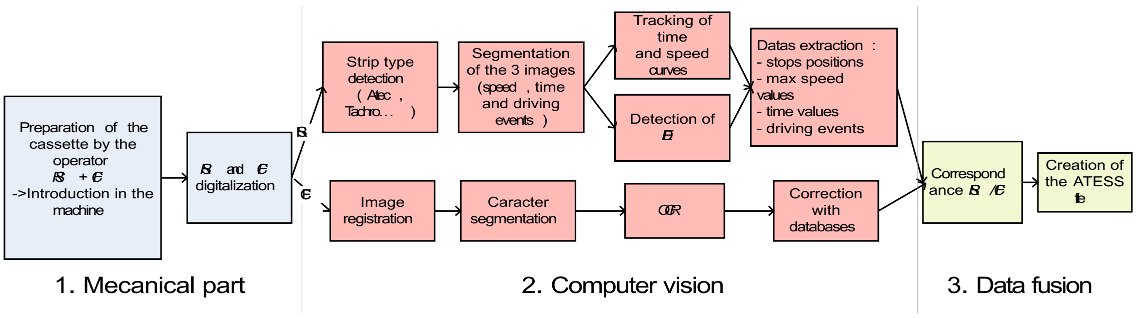

Figure 4. Technical presentation of the *ARSS* system.

The present paper is divided into four sections. The mechanical system is described with its main functions in the first part. The methodology for the extraction of *RS* and *FC* information is presented in the second part. Next, the data fusion between RS and FC is explained in the third part. Afterwards, an evaluation of the method with the first tests is presented.

1. Presentation of the mechanical system

In order to make the checking fully automatic, a machine has been developed (Figure 5). Cassettes of adequate size have been created especially to be able to receive any type of the recording strips issued from the different speed recorders currently used in trains. The strips have to be introduced into cassettes with their associated filling-cards before being processed. Thanks to the system which is able to pleat the extremity of the bands and unwind them, using the machine is convenient because the role of the operator is deeply simplified. Otherwise, these special clips have been designed to keep recording strips intact despite of the flimsiness of the paper. Moreover, as the paper is thin and fragile, the machine mechanism was optimized in order to avoid the bands being ripped while unwinding. The system enables the loading of several tapes in a compartment (Figure 5 on the left) where they are stocked till they are read one after one by a vision system composed of a linear camera (i.e. a camera that can see only a line) in charge of reading the strips (Figure 6) and a scanner for reading both sides of the associated filling-card. Then, information is taken from the numerical images acquired by the vision system. Figure 7 and 8 show the loading process of the strip on the paper roller capstan.

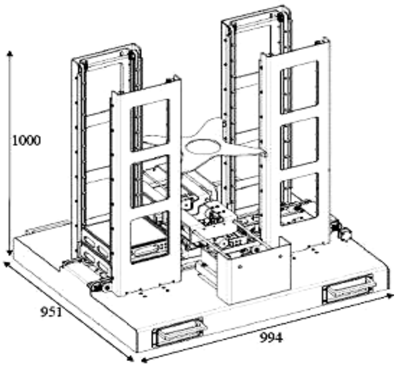

Figure 5. Mechanical part of the system.

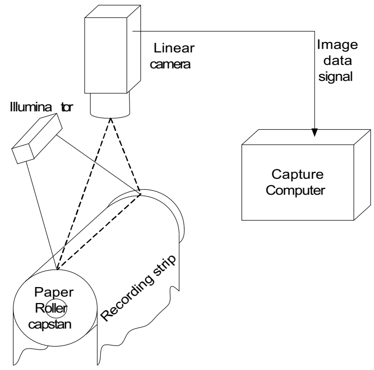

Figure 6. Optical acquisition of *RS*.

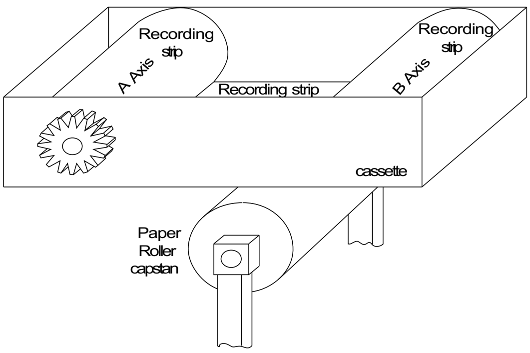

Figure 7. Cassette with low position capstan.

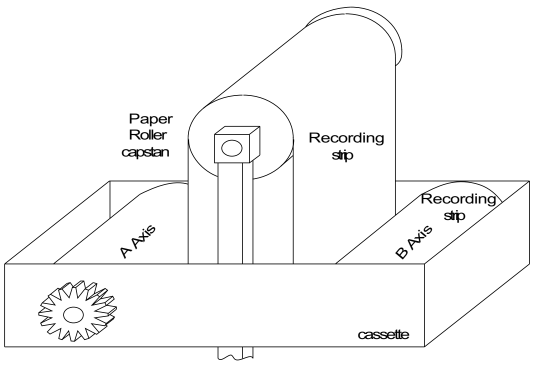

Figure 8. Cassette with high position capstan.

Once the RS and FC components have been processed by the vision system, the strip is rewound in order to allow an operator to check it again if necessary. The operator can pick up the recording strips which have been treated. If any incident has been detected, it can be archived.

2. Computer vision part: Data extraction

a. Strip data extraction

Detection of the type of band

The three different sections of the strip are shown in Figure 9. The characteristics of the recorded data (position of time and speed curve, higher speed, recorded type of *DE*) depend on the type of strip. For example, on a TELOC strip, the *DE* lines are located on the bottom, whereas they are on the top of the strip for the other types. The first difficulty consists in recognizing what kind of tape is to be treated in order to deduce the position of the different curves, the max speed and the kind of *DE*. One element can be used to differentiate the 8 strips: the template of speed/time lines. Eight template images are then created, one for each strip, they represent in black/white the background of the strips.

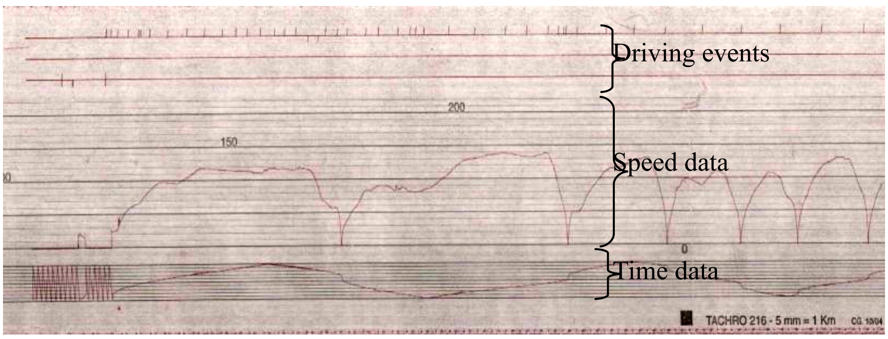

Figure 9. Example of recording strip.

Figure 10. Template image for ATEC 280

After an Otsu segmentation [Otsu], downsizing of the image, a convolution is performed with each of the eight template images. The best score is kept to determine the type of strip. For each template, a corresponding file gives the position of the min/max speed/time position. With the knowledge of the type of strip, we can crop the image into three sections separating time, speed and the *DE*.

Segmentation of the image

The next step consists in extracting the relevant information of these three images, i.e. the curve from the background with the grey template. As one can notice in the following images of Figure 11, the pen draws red curves on the paper. So, this color information will be used to segment the curves. The segmentation is done here using a clustering algorithm called the k-means [Hartigan]. A specific color space is created, mixing two components from the HSV (Hue Saturation Value) and CIELab color spaces. The k-means algorithm performs the classification of objects into different groups. As far as our application is concerned, there are two subsets: pixels belonging to the useful data traced by recording pens and background pixels.

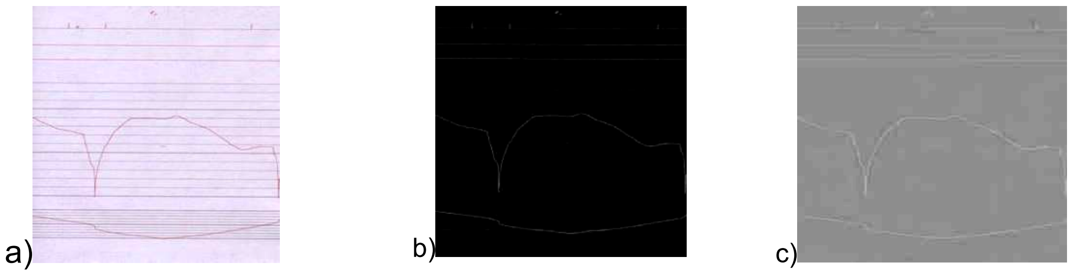

Figure 11. Creation of the a*/s space : a) original image, b) saturation image, c) a* image.

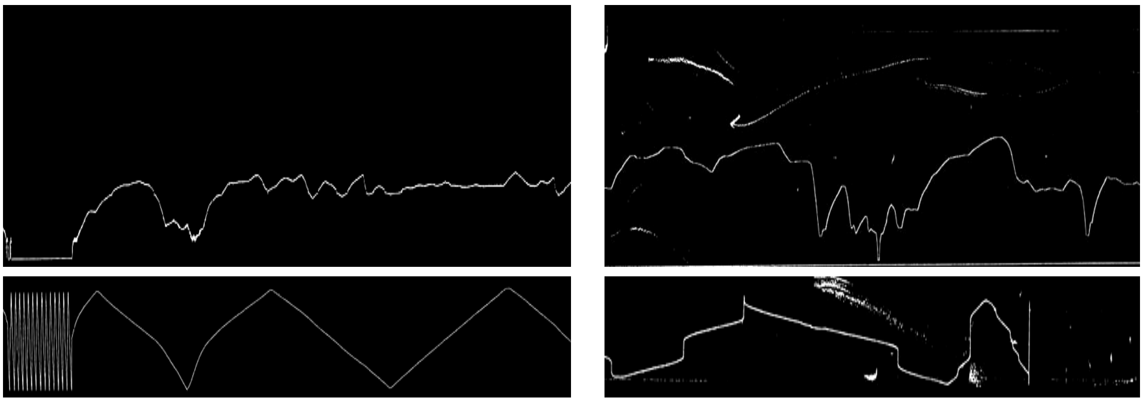

Figure 12. Examples of image segmentation (time and speed).

Time and speed curves tracking

Then, once the beginning of the signals has been detected, the graphs of speed and time are recovered step by step by determining the position of the curves on arcs of circles.

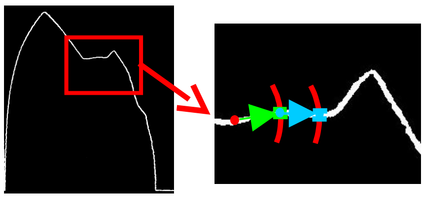

Figure 13. Illustration of speed curve tracking.

In order to simplify the treatment and to avoid errors, the graph of time is broken up into sections where variations are strictly monotonous, and the graph of speed is treated by sections delimited by two consecutive stops as it can be seen on these acquisitions (fig 14).

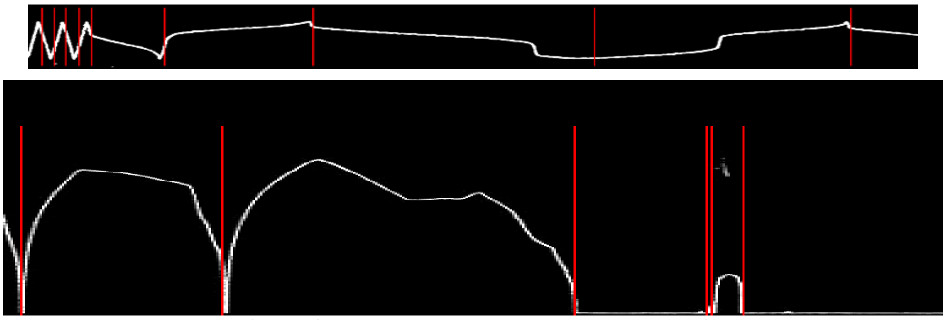

Figure 14. Illustration of time/speed sections created to simplify the tracking process.

Events/signals extraction

The first step for the *DE* extraction is to improve the contrast/quality of the *DE* images. Consequently, a dilatation and a contrast enhancement are performed.

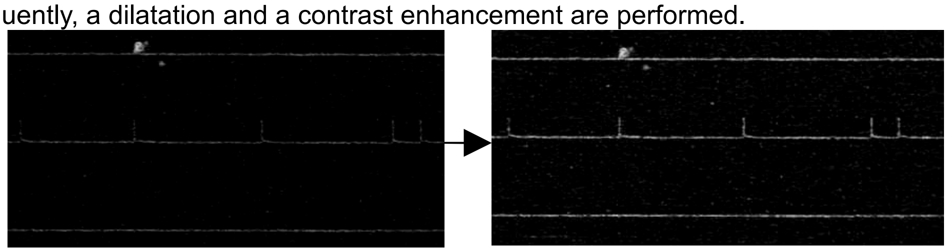

Figure 15. Preprocessing for *DE* extraction.

Thanks to templates of patterns and a method of edge detection, the sequence of driving events can be detected. This method requires a treatment to reduce the noise because some specks on the strips look like the patterns that the system has to detect or conversely. The idea here to remove the noise is to apply filters after a first detection of the *DE*. These filters check the characteristics (width, height, shape, contrast, dispersion) of the *DE* found. If the *DE* does not fit the requirements, it is removed from the list because considered as noise.

b. Filling card (FC) data extraction

An Optical Character Recognition (*OCR*) algorithm is applied on *FC* to translate written text characters into character codes. In order to make possible this optical character recognition, the format of the *FC* has been strictly defined. Each field contains some boxes where operators have to write the adequate information putting one letter or one number per box. The recognition is made more efficient thanks to an editable dictionary, kind of non-exhaustive database of written characters, which has been created from some of these new *FC* filled by drivers during a test period. But, even if this treatment gives very good results for fields with a defined length, it cannot be used as far as the area with the driver's name is concerned. Indeed, it is not possible to reserve some boxes to write it because of the variable length it can have. That is why another method is used. A small numerical image of the handwritten name is saved by the software. It is not translated in the numerical file but can be consulted by the operator if necessary when an anomaly is detected on the run.
We describe now in more details each step of the form recognition of *FC*.

Pre-processing for form understanding

Once filling-cards are scanned, registration of the card to understand with a reference one is needed in order to extract handwritten characters as accurately as possible. Hence, we use a linear pair wise correlation based on a thick line all around the form. Robustness of registration is increased by using the four corners of the form in order to take into account some non-affine transformations. Figure 16 shows the contour of the form with one of the four corners.
After the registration and the affine transformation of the card to be analyzed, boxes around each possible handwritten character have a known location and a straight forward character segmentation is performed.

As the dedicated recognizer we use is based on binary patterns, a binarization of previously extracted characters is mandatory. The global Otsu thresholding [Otsu] is performed for each character of each field.
Due to some slight misregistration errors and the proximity of every box, some box edges may also be segmented and their removal is required to increase character recognition. Based on a directional filtering, box edges are finally removed even if they are connected to the character, which is difficult to handle in a generic application.

Multi-layer-perceptron-based recognition
Two important steps are required in isolated handwritten character recognition: feature extraction to characterize the general shape of the character and the classification itself. Conventional features characterize distribution of points, transformations and series expansions of structural analysis such as moments, n-tuple, characteristic loci and so on, for either binary images, skeletons (thinned characters) or grey-scale images. Here, we used a dedicated code to characterize internal and external edges of characters plus geometrical moments [Thillou].
Then a classification is usually applied based on features and we use a statistical supervised recognition with neural networks, which have proven their efficiency in handwritten character recognition. More particularly, a Multi-Layer Perceptron (*MLP*) with back propagation has been developed.
Briefly, a *MLP* is a network of simple processing units arranged into a hierarchical model of layers. The units (neurons) in the first layer are connected to neurons in the subsequent hidden layer(s), to the final layer. Numerical feature vectors of patterns are presented at the input layer, and activity flows through the network to the output layer. Connections have a numerical weight value associated with them, and the signal transmitted via a connection is multiplied by the weight value. Each unit computes some function of the sum of its weighted inputs, and transmits the result through its output connections by comparing with a threshold. This kind of activation function is generally the sigmoid function. Figure 17 explains the basic principle of a *MLP*.

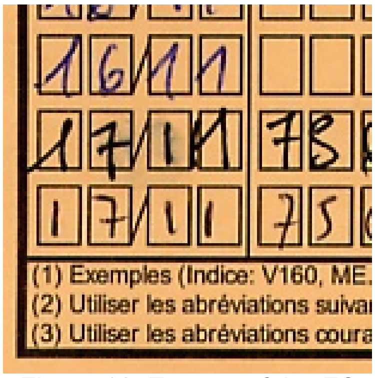

Figure 16. Excerpt of the *FC* showing the thick line used for registration with one of the four corners of the form.

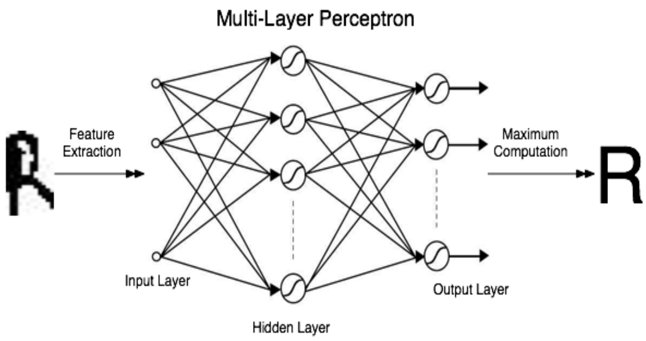

Figure 17. Basic explanation of the multi-layer perceptron used in handwritten character recognition.

Our recognizer has been trained on more than 56000 realistic characters, directly extracted from existing cards filled by drivers on a given period of time. Depending on the field to recognize (for instance, train number, date and so on), several trainings have been performed to handle a different number of outputs. For example, dates are filled with numbers only and a 10-class MLP is sufficient to increase recognition rates whereas train numbers may be filled with letters and numbers, which requires a 36-class MLP.

Lexicon-based correction through finite state machines
Character recognition is error-prone and correction including a priori information is mandatory to increase recognition rates. We perform a lexicon-based correction using finite state machines in order to have quick answers as the search in huge databases is usually very long.

A finite state machine (*FSM*) contains a finite number of states and produces outputs on state transitions based on inputs. *FSM*s are widely used to model systems in diverse areas and have been used in text correction in [Beaufort&Thillou]. They often lead to a compact representation of rules, which can be lexical for example, which is considered as natural by linguists.
Hence in order to correct a date for example, regular expressions have been written to code the grammar for a date e.g. day field is between 1 or 01 and 31, depending on each month.
By using the N-best of *OCR* outputs (here N is experimentally chosen equal to 5), we manage to correct many recognition errors as stated in the Evaluation section. Moreover, this way for encoding lexicon enables to model a huge dictionary (for train numbers, for instance) and to include large flexibility (some boxes may be filled or not depending on the length of the field, all train numbers have different lengths with a known maximum one).

3. Data fusion between the strip and the filling-card

One main job of the people who manually check the strip is to position the departure/arrival stations on the strip from the filling-card information. Thus, the *ARRS* has to automatically find on the strip the position of the departure/arrival station.
To solve this point, the information at our disposal is, on one hand, the stops of the train detected on the *RS* and, on the other hand, the list of stations from the *FC*. A multi-hypothesis algorithm to match Stops/Stations was developed. The idea was to find a mapping that minimizes the sum of the difference between station positions (theoretical positions from the filling-card) and the stops position from *RS*. The main challenge here is that perfect stop detection and a perfect *OCR* with no error are needed.

4. User Interface

After the automatic acquisition and processing of *RS* and *FC*, the user still has the possibility to validate the obtained results. Figure 18 below shows the Graphical User Interface (*GUI*). One can see that the RS is seen with addition of colour information to show the tracking result of the speed (blue), the tracking result of time (red), the detection of *DE* (yellow) and the position of origin/destination with names (green). The operator can type or rectify some information if the system fails in the treatment of a tape, but the aim is to reduce his interventions as much as possible. That is why the system is able to correct some erroneous information with the help of some rules established for certain fields of the *FC*.

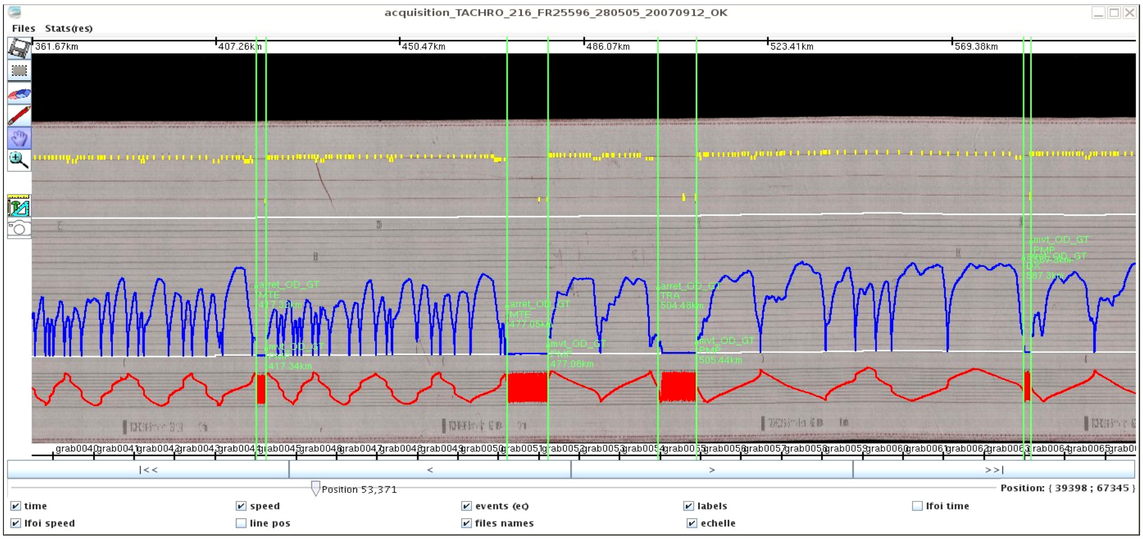

Figure 18. Graphic User Interface (GUI) for results control.

All other data are then gathered into a file readable by AIDA post-processing software. Finally, AIDA can detect suspicious event like over speed or emergency breaking.

5. Performance evaluation and first results

For the tapes, type of strips is 100% well detected, *DE* detection is 99.8% accurate (we use framework of [Desurmont] to evaluate the result). Train stop detection is more than 90% accurate. We are currently improving these results and believe to detect almost 100% of the stops.

Concerning the text recognition for *FC*, each step has been carefully designed to reach satisfying runtimes, and for all steps from registration to recognized character correction, the processing of one mission is less than 200 ms. Each previously defined step contributes to increase recognition rates, but the final correction step enables to drastically increase field recognition rates. By field, we mean the recognition rate of all characters of a given field. Hence, if one error is present in a field, such as train number, the recognition is counted as false. We perform evaluation on more than 960 missions written by many different drivers. The correction step roughly increases the field recognition rate for train number with a factor of 7.6.

We finally estimate that the system allows dividing the human time resources by a factor of 10 when it is used to perform a full check.

6. Conclusion

The Automatic Reader of Recording Strips (ARRS) is an innovating concept which enables to automatically process Recording Strips (RS) and Filling-Cards (FC) in a jointly way, whereas at the moment, those tasks are manually performed by operators.

Cassette of adequate size have been created especially to be able to receive any type of the recording strips issued from the different speed recorders currently used in trains. Any type of recording strip can be recognized and processed by the same machine.

As the aim of the project is to fully check all recordings as quickly as possible, the whole system has been designed in order to improve the processing time. Indeed, loading the strips and the filling-cards in the cassettes has been made very easy, and loading several tapes in the storage compartment of the automatic reader allows the operator to work very fast.

The computer vision algorithms developed for the automatic recognition of data from the strips and the filling-cards, using state of the art technology, were extensively described.

Making automatic the checking of recording strips and filling-cards will deeply modify the organization of checking centers. Thanks to the Automatic Reader of Recording Strips (ARRS), a fully check of all the recordings will be possible to perform in reasonable processing times.

This innovative system is expected to enhance railway operation safety, by making the check of driving of incidents more effective.

References

[Beaufort&Thillou] R. Beaufort, C. Mancas-Thillou, "A Weighted Finite-State Framework for Correcting Errors in Natural Scene OCR", Proc. of IEEE/IAPR International Conference on Document Analysis and Recognition, Curitiba (Brasil), pp 889-893, 2007.

[Desurmont] X. Desurmont, R. Sebbe, F. Martin, C. Machy and J-F. Delaigle, "Performance Evaluation of Frequent Events Detection Systems", Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, New York, 18th June 2006.

[ERA] http://www.era.europa.eu/public/Documents/Safety/Safety_Performance/06-06-01ERA-Report.pdf

[Hartigan] J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.

[Otsu] N. Otsu, “A threshold selection method from grey scale histogram”, IEEE Trans. on Syst. Man and Cyber., vol 1, pp 62-66, 1979.

[Thillou] C. Mancas-Thillou, "Natural Text Scene Understanding", ISBN: 978-2-8746-3059-4, Louvain, 2006.