16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

# AUTOMATIC EXTRACTION OF INFORMATION FOR CATENARY SCENE ANALYSIS

*Florent Montreuil*[1,2], *Régis Kouadio*[1,2], *Caroline Petitjean*[1], *Laurent Heutte*[1], *Vincent Delcourt*[2]

[1]Université de Rouen, LITIS, EA 4108
BP 12, 76801 Saint-Etienne-du-Rouvray, France
[2]Direction de l'Innovation et de la Recherche, SNCF,
45 rue de Londres, 75379 Paris Cedex 8, France
florent.montreuil@gmail.com, regis.kouadio@sncf.fr, caroline.petitjean@univ-rouen.fr,
laurent.heutte@univ-rouen.fr, vincent.delcourt@sncf.fr

**ABSTRACT**

This paper presents an original system for the automatic recognition of catenary elements. Based on a bottom-up approach, our analysis is composed of two stages: the first stage provides recognition hypothesis regarding previously segmented elements, with a precision rate of 91.7 %. The second stage allows to correct some hypothesis using a Markovian model which analyzes the catenary element sequence. It allows to improve precision up to 98.1%.

## 1. INTRODUCTION

### 1.1 Background

Overhead wires that are used to transmit electrical energy to trains are called the catenary. The contact wire (CoW) provides electricity and the carrying wire (CaW) supports the contact wire. Three vertical elements support the two wires: supporting arms (SA), droppers (D) and droppers with electrical connection (DEC). The part of the catenary that is delimited by two supporting arms is called a catenary stave (Figure 1). The number of elements and their location inside a catenary stave are specified in the so-called mounting rules.

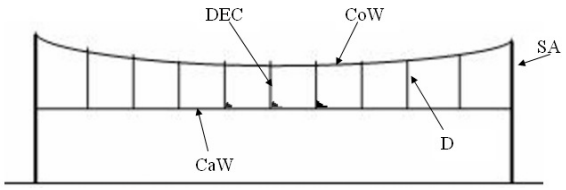


Figure 1: Perpendicular view of a catenary stave

Maintenance of railway infrastructure consists in checking the presence and the integrity of each element. Today maintenance is carried out by visual inspection. The Innovation and Research Department of the French railways (SNCF) plans to automate this long and fastidious task. A dedicated acquisition system has thus been embedded inside a TGV coach [4]. The speed of the TGV being 320km/h, a high frame acquisition rate is required (53kHz). Thanks to a regulation of the obturation rate by the train speed, images are not fuzzy. Moreover, filters compensate for bad weather conditions. Images are acquired perpendicularly to the catenary. Their size is 1024x768 pixels and each pixel is coded on an 8-bit-gray level scale. For each image, the acquisition position on the line is provided. Horizontal resolution is 1.8 mm per pixel when the train speed is constant, but the resolution may vary during acceleration or deceleration.

Images represent adjacent segments of the catenary stave. A stave is made of about 40 images. A catenary stave can be either single, with one pair of contact wire and carrying wire (Figure 2) or double, *i.e.* with two pairs of contact wires and carrying wires (Figure 3), or even triple.

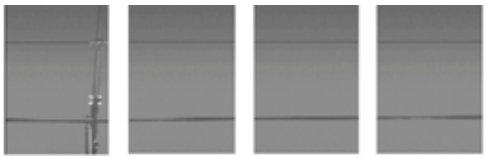

Figure 2: Successive images of a single catenary stave

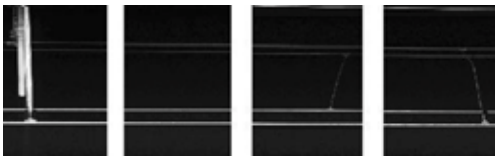

Figure 3: Successive images of a double catenary stave

The aim of the study is to conceive and develop an automatic image processing system that allows to identify the catenary elements as described earlier, using the mounting rules. The ultimate goal will then be to detect defaults on identified elements. In this paper we will solely focus on the element recognition part, as being a preliminary task to the default detection process. Such an analysis of the catenary elements is an original application which has no equivalent today.

### 1.2 State of the art and adopted approach

Identifying the catenary elements can be seen as a scene analysis problem, for which two main approaches may be used [1]:

- a bottom-up approach: simple, element-independent features are extracted from the image. These features are gathered and knowledge is incorporated into the recognition process in order to finally identify the objects.
- a top-down approach: this approach relies on the hypothesis that the image contains a particular object and thus consists in predicting the presence of features in the image, using high-level *a priori* knowledge.

The model relies here on the mounting rules and a description of the different types of catenary staves. It indi-

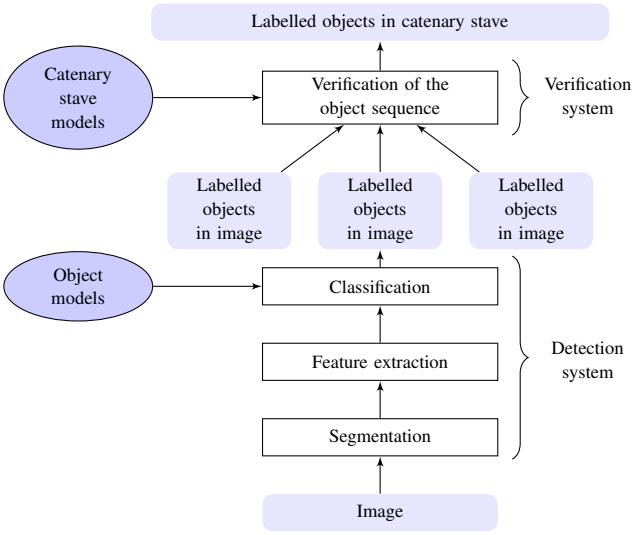


Figure 4: Overview of the system

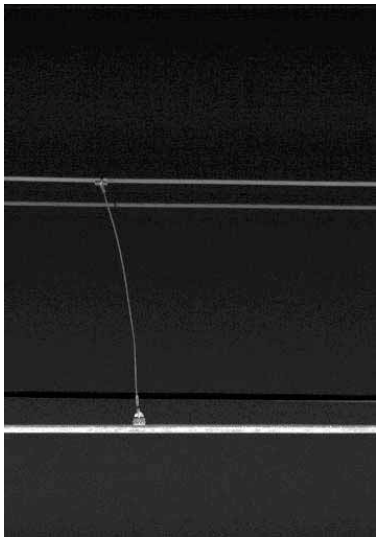

Figure 5: Dark and light objects in a catenary stave

cates, for each type of catenary stave, the number of droppers and droppers with electrical connection and the space between them. The model is not fully applicable here because horizontal resolution varies.

Thus, in order to use the system on all images, we chose a bottom-up approach. In a first step, we propose a method based on a classical scene analyse scheme, which is done image by image: segmentation in vertical and horizontal components, feature extraction and classification after a learning phase. In a second step, the labels which are attributed by the classifier are checked and corrected, by considering the whole stave for analysis: the sequence of vertical elements is checked using *a priori* knowledge obtained from the mounting rules. An overview of the system is shown on Figure 4.

The remaining of the article is organized as follows: Section 2 presents the three steps of the detection of catenary elements in each image independently, and Section 3 presents the verification step (Figure 4). The last part concludes and draws some perspectives for this work.

## 2. ANALYSIS IN THE CONTEXT OF AN IMAGE

The goal of this section is to segment the main elements of the catenary (contact wire, carrying wire, supporting arms, droppers, droppers with electrical connection), to perform feature extraction and to classify the type of each element, in each image independently. Therefore, the input of this detection step is a single, 1024x768 pixel image (typically 1.38m of the catenary) and the output is the presence and location of elements of interest.

### 2.1 Segmentation

The elements of the catenary are quite linear and may thus be detected by thresholding horizontal and vertical projections of the image. But images are not as straightforward to process as they seem. Indeed, they show inhomogeneous noisy background, that has a poor contrast with the droppers. Droppers are also very thin, only 2 to 3 pixels wide. Furthermore, some objects are darker and some other are lighter than the background (Figure 5). Note that for visualization purposes, all images presented in this paper have been manually contrast-enhanced.

In order to simplify the segmentation process, the image is binarized. Because of the background inhomogeneity, binarization is performed using TopHat and BotHat morphological operators (Equation 1). The TopHat (respectively BotHat) operator allows to segment elements which are lighter (respectively darker) than the background [5].

$$\begin{aligned} \text{TopHat} &= \text{Image} - \text{Opening(Image)} \\ \text{BotHat} &= \text{Closing(Image)} - \text{Image} \end{aligned} \tag{1}$$

The two structuring elements for these operators are constructed according to the shape and the size of the elements to be detected. They are 10-pixel long horizontal and vertical lines.

A binarization result is shown on Figure 6.

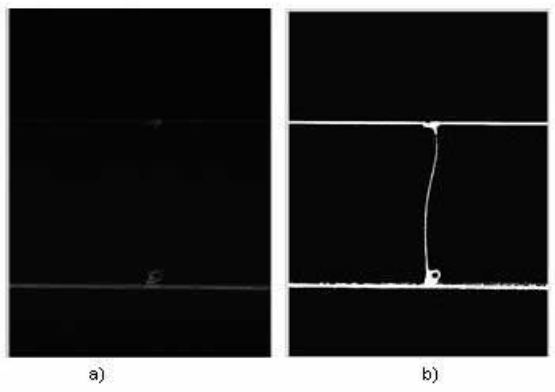


Figure 6: a) Original image, b) Binarized image

Segmentation is then performed by thresholding the projections of the binarized image along both directions:

- vertically, in order to detect vertical elements (supporting arms, droppers and droppers with electrical connection) ;
- horizontally, in order to detect horizontal elements (contact wire and carrying wire). Since the wire horizontality is not always maintained, the rotation angle is taken into account before projecting the image, by means of a Radon transform [3].

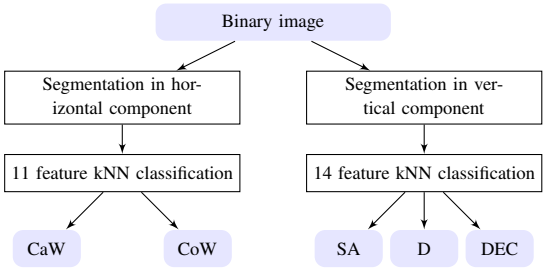


Figure 7: Segmentation, feature extraction and classification

### 2.2 Feature extraction and classification

We now aim at recognizing the segmented objects, by implementing a feature extraction process and a $K$-nearest neighbour (kNN) classifier, as described in Figure 7. The features to be extracted, computed from the bounding box, have been chosen to describe the shape of segmented objects. They include 11 global features:

- pixel density,
- height-to-width ratio,
- Zernike's eight first moments [7],
- mean of vertical projection,

and 3 local features:

- variance of vertical projection,
- size of the biggest square surface,
- sum of vertical profile.

Local features are used mainly to discriminate vertical elements, that can be very similar (Figure 8).

### 2.3 Segmentation and classification results

Our segmentation method has been tested on an initial database of 603 images (see Table 1), representing 1741 objects to be segmented, for which the ground truth has been obtained by manually tracing a bounding box around the objects.

Table 1: Learning base and test base

| # | Learning base | Test base |
|---|---|---|
| Images | 1160 | 603 |
| Elements | 2989 | 1741 |
| Staves | 24 | 14 |
| of which: | | |
| - single stave | 9 | 7 |
| - double stave | 12 | 6 |
| - triple stave | 3 | 1 |
| CoW | 1392 | 816 |
| CaW | 1316 | 746 |
| D | 211 | 120 |
| DEC | 33 | 21 |
| SA | 37 | 38 |

The segmentation performance is assessed by computing a distance $d$ between the manually obtained bounding box (denoted by $BB_m$) and the one obtained automatically (denoted by $BB_a$). The distance $d$ is defined as the maximum of absolute differences between heights and widths of the bounding boxes. The segmentation results can then be of four types:

- correct segmentation, *i.e.* such as $d(BB_a, BB_m) \leq S$,
- incorrect segmentation, *i.e.* such as $d(BB_a, BB_m) > S$,
- undetection, *i.e.* $BB_a$ does not exist whereas $BB_m$ exists,
- false alarm, *i.e.* $BB_a$ exists whereas $BB_m$ does not exist.

where $S$ is a threshold (typically 20 pixels) allowing to prevent from the approximation of manual labelling.

Some segmentation examples are shown on droppers and droppers with electrical connection on Figure 8.


Figure 8: Segmentation results: a) Dropper, b) Dropper with electrical connection

| % | Correct segmentation | Incorrect segmentation | Undetection | False alarm |
|---|---|---|---|---|
| CoW | 99.7 | 0.1 | 0.1 | 0.0 |
| CaW | 87.4 | 0.0 | 12.1 | 0.6 |
| D | 90.0 | 4.5 | 0.0 | 5.5 |
| DEC | 94.7 | 0.0 | 0.0 | 5.3 |
| SA | 96.8 | 3.2 | 0.0 | 5.3 |

Table 2: Segmentation performance

As can be seen in Table 2, segmentation results are quite satisfying. Still, some errors remain, because of:

- the undetection, of the wires mainly. The segmentation fails when wires are too thin or have the same grey level as the background;
- false alarm cases, because of a static threshold used for segmentation of vertical elements.

On the previously cited features a kNN classifier is implemented. The learning base and test base are given in Table 1 and the correct classification rate *vs.* the number of neighbors is presented in Figure 9. We notice that performance are maximum for $K = 3$ and for $K = 13$. In order to better estimate the *a posteriori* probability assigned to each class, we have retained $K = 13$.

Recall and precision rates are also two widely used measures for assessing the quality of results of detection and information extraction tasks. They are defined as follows:

$$\text{Precision} = \frac{\text{Number of correctly recognized objects}}{\text{Number of objects to be recognized}}$$

$$\text{Recall} = \frac{\text{Number of correctly recognized objects}}{\text{Number of objects proposed by the system}}$$

Recall is a measure of the ability of the system to localize and recognize all presented objects. Precision measures the

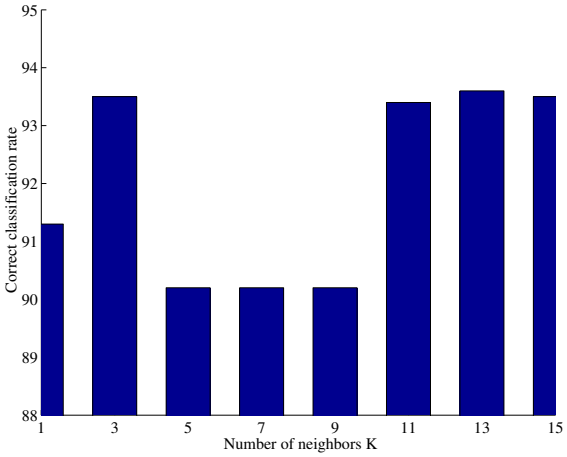


Figure 9: Performance of the classifier for different values of $K$

ability of the system to provide only correct hypothesis and thus to limit the number of false alarms. Both rates, as shown in Table 3, are satisfying and show the good performance of our system, especially on single stave images. Nonetheless they can be improved by adding an analysis at a stave level that includes *a priori* information obtained on catenary stave models.

Table 3: Recall and precision rates of the system

| % | Recall | Precision |
|---|---|---|
| Single catenary stave | 91.7 | 91.7 |
| Single, double, triple, catenary stave | 88.2 | 87.2 |

## 3. ANALYSIS IN THE CONTEXT OF A CATENARY STAVE

Most of the false alarm cases come from vertical elements (see Table 1). In order to check the consistency of the sequence of vertical elements, the sequence is compared to a catenary model, available from the mounting rules. This type of operation is usually performed using Hidden Markov models (HMM) [6]. HMM have been widely used for computer vision applications [2]. As images of double and triple catenary staves are much more complex to analyze, we present here a first attempt to modelize only single staves.

### 3.1 Theoretical framework of the HMM

The analysis of vertical elements using HMM is twofold: (i) verification of the consistency of the vertical elements sequence using the mouting rules, (ii) identification of the catenary stave model, using a database of models. The stave model consists in a description of the element sequence that composes the different types of existing staves. For single staves, there exists 13 types of staves, from the simplest sequence made of 4 droppers to the most complex made of 15 elements combining droppers and droppers with electrical connections (Figure 10).

An HMM is made of a hidden layer and an observable layer. The hidden layer determines the state of the vertical element labels (SA, D, DEC) whereas the observable layer represents the sequence of labels given by the classification step. Let us denote by $N$ the number of elements (here $N = 3$),

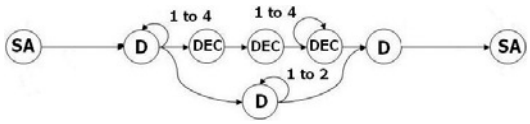


Figure 10: Graphical modeling of a stave

$p_1, ..., p_N$ the elements labels, and $V_1, ..., V_M$ the $M$ observation symbols. Let $Z_t$ be the state at time $t$, the model can be described using a triplet of matrices $(\Pi, A, B)$, where:

- $\Pi = \{\pi_i\} = \{P(Z_1 = p_i)\}_{i=1,...,N}$ is the initial state distribution vector. Since a stave always starts by SA, $\Pi$ is then $\{1\ 0\ 0\}$.
- $A = \{a_{i,j}\} = \{P(Z_t = p_j | Z_{t-1} = p_i)\}_{\substack{i=1,...,N \\ j=1,...,N}}$ is the state transition probabily distribution matrix.
- $B = \{b_i(k)\} = \{P(Z_t = V_k | Z_t = p_i)\}_{\substack{i=1,...,N \\ k=1,...,M}}$ is the observation symbol probability distribution matrix. Since the observation of a vertical element corresponds to its generation by an identical state of the model, matrix $B$ is the 3x3 identity matrix.

The learning of matrix $A$ is done using the Baum-Welch algorithm on manually labelled datasets. These datasets were constructed using the mounting rules, because there were not enough data in the learning base (*cf.* Table 4). Once learning is done, one must decide which state sequence is the most likely, given the observations sequence provided by the kNN classifier. This search is usually performed using Viterbi algorithm [6].

### 3.2 Results of the verification step

The experiments have been conducted on databases described in Table 4, as follows: on each catenary stave, the detection system provides element hypothesis, which are then concatenated into sequence of elements. This sequence is aligned on each of the 13 models of catenary staves. The best alignement found by Viterbi algorithm provides a way to correct some element hypothesis. This corrected sequence is compared to the labelled one in order to compute precision and recall rates. We present in Table 5 the difference between results obtained by the detection system solely and the complete system, in terms of precision and recall rates.

Table 4: Learning and test bases constitution

| # | Learning base | Test base |
|---|---|---|
| Images | 369 | 280 |
| Elements | 876 | 802 |
| Single catenary | 9 | 7 |

Table 5: Recall and precision rates of the system

| % | Recall | Precision |
|---|---|---|
| Single catenary without model | 91.7 | 91.7 |
| Single catenary with model | 95.7 | 98.1 |

This performance highlights the improvement of the first

results thanks to the verification step. Nonetheless these results need to be validated on larger learning and test bases.

Computation time on a DualCore PC with 2Go RAM, without any code optimisation is approximately 2.5 seconds per image in a single catenary. A line of 300km represents approximately 400 000 images, which requires about a hundred hours of processing to list the elements.

## 4. CONCLUSION

The present work deals with the automatic information extraction in a catenary scene. This analysis is made in two steps: a first pattern recognition system allows to identify elements that are part of the scene, using specific features. Next, a second system checks and corrects the first system results by analyzing them in a more global context: the catenary stave. The performance obtained on both systems highlight the benefit of a verification step.

Future work concerns improving the model. The mounting of a catenary stave imposes a certain range for inter-element distance, a range that depends on the stave type. The model could take into account this distance, but would then restrict the application of the system to images with constant horizontal resolution. However it would enable us to compare our bottom-up approach to a top-down one on the same type of images. The model should also be designed so as to take into account double and triple catenary staves.

Computation time could also be improved, by implementing an SVM classifier instead of the kNN classifier.

At last, we recall that this work is just a preliminary step before the object default identification, on which we are currently focusing our investigation.

## REFERENCES

[1] J. Battle, A. Casals, J. Freixenet and J. Martí, "A review on strategies for recognizing natural in colour images of outdoor scenes," *Image and Vision Computing*, vol. 18, pp. 515-530, 2000.

[2] H. Bunke and T. Caelli, "Hidden Markov Models Applications in computer vision," *Machine Perception and Artificial Intelligence*, vol.45, 2001.

[3] S.R. Deans, "The Radon Transform and Some of Its Applications," *John Wiley & Sons, New York*, 1983.

[4] R. Kouadio, V. Delcourt, L. Heutte and C. Petitjean, "Video based catenary inspection for preventive maintenance on Iris 320," *World Congress of Railway Research*, Soul, Korea, May 2008.

[5] C. Lacoste, X. Descombes, J. Zerubia and N. Baghdadi, "Unsupervised line network extraction from remotely sensed images by polyline process," in *Proc. EUSIPCO 2004*, Vienna, Austria, September 2004.

[6] L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-286, 1989.

[7] M. Teague, "Image analysis via the general theory of moments," *Journal of Optical Society of America*, vol. 70, pp. 920-930, 1980.