A semi-automatic groundtruthing framework for performance evaluation of symbol recognition & spotting systems

Mathieu Delalandre, Jean-Yves Ramel and Nicolas Sidere
Laboratory of Computer Science
François Rabelais University, Tours city, France
firstname.lastname@univ-tours.fr

***Abstract*—In this paper, we are interested with the groundtruthing problem for performance evaluation of symbol recognition & spotting systems. We propose a complete framework based on user interaction scheme through a tactile device, exploiting image processing components to achieve groundtruthing of real-life documents in an semi-automatic way. It is based on a top-down matching algorithm, to make the recognition process less sensitive to context information. We have developed a specific architecture to address the recognition problem in constraint time, working with a sub-linear complexity and with an extra memory cost.**

***Keywords*-symbol recognition & spotting, performance evaluation, semi-automatic groundtruthing**

I. Introduction

This paper deals with the the performance evaluation topic. Performance evaluation is a particular cross-disciplinary research field in a variety of domains such as Information Retrieval, Computer Vision, CBIR, etc. Its purpose is to develop full frameworks in order to evaluate, to compare and to select the best-suited methods for a given application. Two main tasks are usually identified: groundtruthing, which provides the reference data to be used in the evaluation, and performance characterization, which determines how to match the results of the system with the groundtruth to give different measures of the performance.

In this work, we are interested with the groundtruthing problem for performance evaluation of symbol recognition & spotting systems. We propose a complete framework based on user interaction scheme through a tactile device, exploiting image processing components to achieve groundtruthing of real-life documents in an semi-automatic way. In the rest of the paper, section 2 will present related work on this topic. Then, in section 3 we will introduce our approach.

II. Related works

Groundtruthing systems can be considered according three main approaches: automatic (i.e. synthetic), manual and semi-automatic. Concerning performance evaluation of symbol recognition & spotting, most of the proposed systems are automatic [1]. In these systems, the test documents are generated by a generation methods which combines pre-defined models of document components in a pseudo-random way. Performance evaluation is then defined in terms of generation methods and degradation models to apply. The automatic systems present several interesting properties for performance evaluation (reliability, high semantic content, complete control of content, short delay and low cost, etc.). However, the data generated by these systems still appears quite artificial. Final evaluation of systems should be completed by the use of real data to proof, disprove and complete conclusions obtained from synthetic documents.

Semi-automatic and manual systems deal with the groundtruth extraction from real-life documents. At best of our knowledge only the systems described in [2], [3] have been proposed to date for performance evaluation of symbol recognition & spotting, and both of these systems are manual. In [2], the authors employ an annotation tool to groundtruth floorplan images. The groundtruth is defined in terms of RoI[1] and class names. Such an approach remains quite subjective and few reliable due to image ambiguities and errors introduced by human operators. In addition, the obtained groundtruth is defined "a minima" i.e. only rough localization and class names are considered.

The EPEIRES[2] platform [3] is a manual groundtruthing framework working in a collaborative fashion. It is based on on two components: a GUI to edit the groundtruth connected to an information system. The operators obtain from the system the images to annotate and the associated symbol models. The groundtruthing is performed by mapping (moving, rotating and scaling) transparent bounded models on the document using the GUI. The information system allows to collaboratively validate the groundtruth. Experts check the groundtruth generated by the operator by emitting alerts in the case of errors. The major challenge of this platform is to federate a community. Indeed, the groundtruthing process is time consuming due to the user-interaction with the GUI and the additional validation steps. Due to these constraints, no "significant" datasets have been constituted to date using this platform [1].

A way to solve the limitation of manual systems is semi-automatic groundtruthing [4]. This approach is popular in the

[1]Region of Interest
[2]http://www.epeires.org/

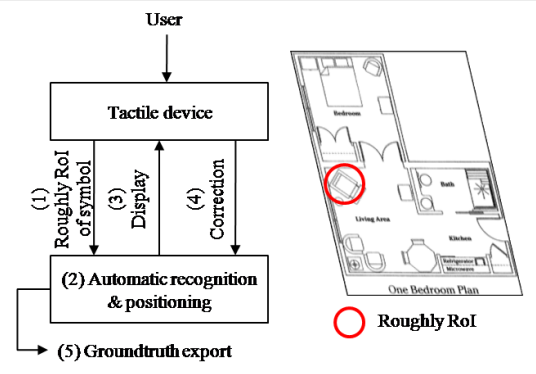

Figure 1. Overview of our system

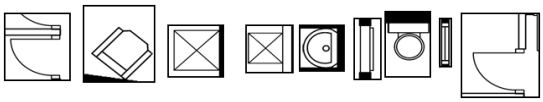

Figure 2. Some examples of roughly RoIs

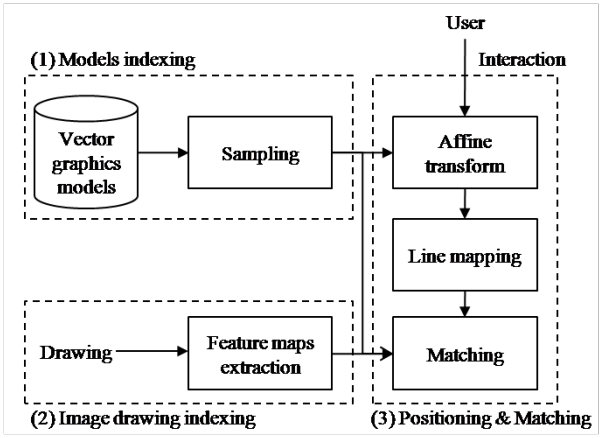

Figure 3. Architecture of our system

field of DIA[3], systems have been proposed for performance evaluation of chart recognition [4], handwriting recognition [5], layout analysis [6], etc. Major challenge of these systems is the design of image processing components able to support the groundtruthing process and the user-interaction. Such components are application dependent, and at best of our knowledge none has been proposed to date to support performance evaluation symbol recognition & spotting. This paper presents a first contribution on this topic, the next section will introduce our approach.

III. Our approach

A. Introduction

The general overview of our system is presented in Fig 1. This one uses a mixture of auto-processing steps and human inputs. User interaction is done through a tactile device (e.g. smartphone, tablet or tactile screen). Then, for every symbol on the document it is asked to the user to outline it in a roughly way (1). Specific image and recognition processings are then called to recognize & localize the symbol automatically (2). In the case of miss-recognition, the user can correct the result manually (4) based on results display (3). Otherwise, implicit validation is obtained when no correction is observed. At last, groundtruth is exported (5) including the class name, the precise location, the scale factor & orientation of the symbol and its graphics primitives. With this approach, we constraint the user to outline individually each symbol. We didn't consider the automatic spotting methods [2] to gain in robustness.

Regarding the user-interaction scheme defined above, auto-processing for semi-automatic groundtruthing must deal automatic recognition and positioning of symbols in context (Fig. 2). These symbols are obtained following roughly outlines of users. To support the production of groundtruth, the auto-processing must be robust enough and work in constraint time to allow a fluent user-interaction. We propose here a specific system with algorithms that support these constraints. Our recognition & positioning approach is top-down i.e. symbol models will be matched to the RoIs describing symbols for better robustness to context elements. In addition, we define it as partially invariant to scale and rotation change and constraint users on providing rough approximation of scale and rotation parameters (i.e. size and direction of RoI). The full process works with a sub-linear complexity and with an extra memory cost. The Fig. 3 presents the general architecture our system. This one is composed of three main blocks: indexing of models (1), indexing of the drawings (2), and then positioning & matching process (3). We will briefly present each of them in next subsections B, C and D.

[3]Document Image Analysis

B. Indexing of symbol models

To support our matching and positioning algorithm (see section D), our models are given in a vector graphics form. We complete this representation by applying a sampling process in order to extract a set of representative points of symbol models (Fig. 4). We set this sampling process with sampling frequency f_s. This frequency fixes the number of points n to extract and their inter-distance gap T. The parameter L corresponds to the sum of lengths of vector graphics primitives composing the symbol. Like this, this process will respect an unique inter-distance gap T for all the symbol models. The number of points n will change regarding the number and length of primitives composing the symbol. The frequency parameter is limited at minimum to a value $\frac{1}{L}$ (i.e. two points at least for a line).

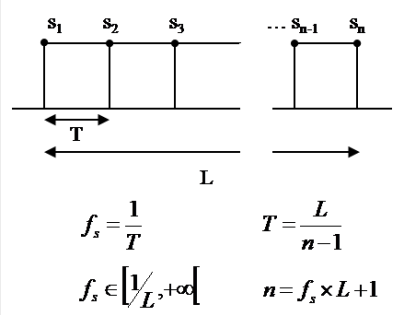

Figure 4. Sampling process

Model	p_i	is a sampled point of a model to fir with the feature map
	α	is the local orientation of the sampled model stroke with $\alpha \in [\gamma_u, \gamma_v]$
Features map	$[\gamma_u, \gamma_v]$	is the orientation gap of the map
	q_k	is the nearest foreground point to p_i in the map $[\gamma_u, \gamma_v]$
	β_i	is the orientation of the line p_i, q_k
	d_i	is the length of the line p_i, q_k
	γ_k	is the local orientation estimation of the skeleton stroke, with $\gamma_k \in [\gamma_u, \gamma_v]$

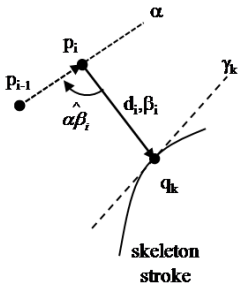

Figure 5. Extracted features

C. Indexing of drawing images

Our matching and positioning process (see section D) will exploit on one side the sampled models, and in the other side the neighbourhood information available on the images. In order to reduce the complexity, we extract previously some features maps with pre-computed information to be use in the positioning & matching. The Fig 5. details the organization of these features.

For a given sampled point p_i of a symbol model, to fit with the features maps, we exploit the α value corresponding to the local orientation of the model stroke that it composes. This value α drives the selection of a features map $[\gamma_u, \gamma_v]$, such as $\alpha \in [\gamma_u, \gamma_v]$. The reading of the pixel p_i will provide directly the features $\{d_i, \beta_i, \gamma_k\}$, corresponding respectively to the distance, the direction and the local orientation estimation of the nearest foreground point q_k in this map.

To extract these features maps, we employ the image processing chain presented in Fig. 7. This chain is executed off-line. It is composed of five main steps:

1) The first step is a skeletonization. The key goal is to adapt the drawing image to the sampled representation of our models. We use the algorithm detailed in [7], as it is well adapted for scaling and rotation variations.
2) In this step, we detect the chain-points composing the skeleton's strokes. We chains and separates them from the junction and end points composing the rest of the skeleton. It is achieved using the method described in [8]. Chain-point are stored as Freeman code for further processing in steps 3 and 4.
3) For every chain-point, we compute a local direction estimation. This estimation is done using the chain code of a local neighborhood within a $m \times m$ mask. Local tangent values are computed within the mask from the central pixel to the "up" and "down" chains. The direction estimation is the average of these values.
4) In the step 4, we process the chain points with their direction estimations by a n-bins separation algorithm. This algorithm aims to build-up the orientation maps, that are root versions of our features maps. It stores every point q_k of local orientation estimation γ_k in the map $[\gamma_u, \gamma_v]$, such as $\gamma_k \in [\gamma_u, \gamma_v]$. The parameter n controls the number of maps, and then fixes the extra memory cost of our approach.
5) In a final step, we apply a Distance Transform (DT) on each orientation map. The DT algorithm is applied on the background part, in order to propagate the d_i features (Fig. 5) to every background pixels. We have "tuned" this algorithm to propagate the β_i and γ_k values to each foreground point.

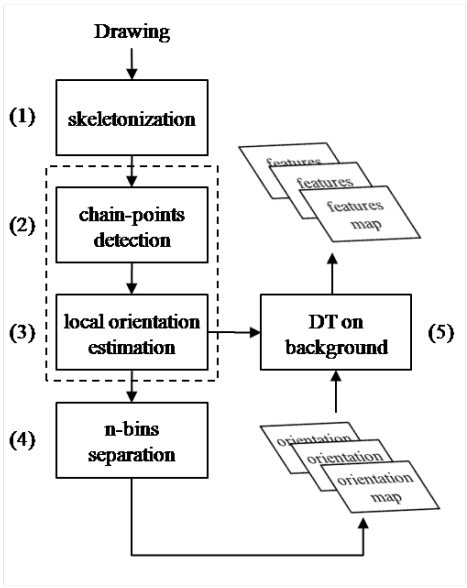

Figure 6. Computation of features maps

D. Positioning & Matching

In a final last step, we exploit the indexed model database and the features maps to achieve the matching & positioning of symbols. As presented in Fig 3. this process relies on three main steps: affine transform, line mapping and matching. We will present each of them in next subsections 1, 2 and 3.

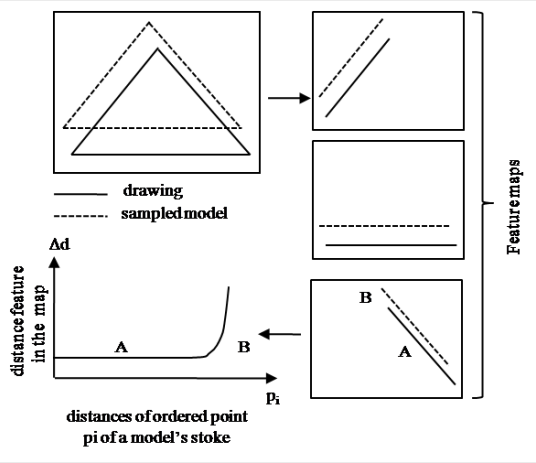

Figure 7. Line mapping

1) Affine transform: Affine transform is the basic operation to take benefit of localization information provided by the users. When a user defines a RoI, the sampled models are fit within that RoI using some affine transform based operations. These operations exploit standard computational geometry methods resulting in shifting, scaling and then orientation change of symbol models with their sampled points.

2) Line mapping: In a next step we achieved a line mapping process Fig. 7. The key goal is to map the strokes composing the model with pixels on the image corresponding to straight lines. This process exploits the features maps computed previously, the local orientations α of the models' strokes are used to drive their selection Fig. 7. In order to be less sensitive to the quantification of features maps, we employ in addition a parameter ϱ such as $[\alpha-\varrho, \alpha+\varrho] \in [\gamma_u, \gamma_v]$. When a multiple selection of maps is observed, the nearest Euclidean distances d_i are considered for selection of q_k.

The sampled points of models are discretized to obtain coordinates and then access the features $\{d_i, \beta_i, \gamma_k\}$ stored in the maps, with an access cost of $o(1)$. Then, we compute for every pair of points p_i the $\triangle d_i$ value Eq. (1). In this equation, i and $i+1$ are the indexes of two successive sampled points p_i, p_{i+1} of the model stroke, and $\triangle d_i$ the difference between their d_i and d_{i+1} features.

As shown in Fig. 7, shifting between model and image lines will result in increasing values of $\triangle d_i$. Here, the area B corresponds to increasing distances whereas the area A remains constant. To solve this problem, we tuned the computation of $\triangle d_i$ into $\triangle_\beta d_i$ Eq. (2). This equation combines the distance d_i and the line orientation β_i in such a namer that $\triangle_\beta d_i$ will will not be impacted by shifting. To do it, we compute direct angle value $\widehat{\alpha\beta_i}$ between vector $\overrightarrow{p_{i-1}, p_i}$ and $\overrightarrow{p_i, q_k}$ Fig. 5. Direct angle takes into consideration the left and right positions between $\overrightarrow{p_{i-1}, p_i}$, $\overrightarrow{p_i, q_k}$ with $\widehat{\alpha\beta_i} \in [0, 2\pi]$. We exploit the $\widehat{\alpha\beta_i}$ value through a φ function Eq. (3) to support the opposite detection cases (i.e. parallel lines at a same distance of the stroke, but on the left and right sides). At the end, the $\triangle_\beta d_i$ curve will present the following properties:

- strict parallel lines,
  $\forall i \ \triangle_\beta d_i \to 0$
- slightly orientation gap between the lines,
  $\forall i \ \triangle_\beta d_i \to K$
- local curvature modification on the image line,
  the $\triangle_\beta d_i$ curve will have a non null tangent
- one-to-many mapping,
  the $\triangle_\beta d_i$ curve will present pick values

$$\triangle d_i = d_i - d_{i+1} \tag{1}$$

$$\triangle_\beta d_i = d_i \sin(\varphi(\widehat{\alpha\beta_i})) - d_{i+1} \sin(\varphi(\widehat{\alpha\beta_{i+1}})) \tag{2}$$

$$\widehat{\alpha\beta_i} < pi \qquad \varphi(\widehat{\alpha\beta_i}) = \widehat{\alpha\beta_i} \tag{3}$$

$$\widehat{\alpha\beta_i} > pi \qquad \varphi(\widehat{\alpha\beta_i}) = -(2\pi - \widehat{\alpha\beta_i})$$

Following the computation of $\triangle_\beta d_i$ for a given model stroke, we perform a mathematical analysis on the obtained curve to determinate the mapping hypothesis. The key objective is to detect the tangent variations in the curve, every mapping hypothesis will correspond to a zone of the $\triangle_\beta d_i$ curve where no tangent variations will be observed. To do it, we compute second derivate $\triangle''_\beta d_i$ and look for the non-null and zero-crossing values. We uses these value as cutting points in the curve. The Fig. 8 presents our mapping model. Every model stroke L_k will result in a set of mapping hypothesis $\bigcup_{\forall p} Mh_p$. Each of these mapping hypothesis Mh_p corresponds to subset of points $\bigcup_{\forall j} p_j$, such as $\bigcup_{\forall j} p_j \in \bigcup_{\forall i} p_i$ with $\bigcup_{\forall i} p_i$ the sampled points of L_k.

In addition, we complete our mapping model with $\overline{\beta_p}, \overline{d_p}, \overline{\gamma_p}$ features, corresponding respectively to the orientation and distance between L_k and the detected line on the drawing, and its local orientation estimation. These features are based on the computation of the $\overline{\triangle_\beta d_j p}$ value of the mapping hypothesis Mh_p, as detailed in Eq. (4). Then, this value $\overline{\triangle_\beta d_j p}$ allows to extract the $\overline{\varepsilon_{\alpha_p}}$ corresponding to the direction gap between L_k and the detected line on drawing as shown in Fig. 9. It is computed as detailed in Eq. (5), using the inter-distance gap T parameter of the sampling process (see section B). Then, $\overline{\beta_p}$ and $\overline{\gamma_p}$ are obtained from $\overline{\varepsilon_{\alpha_p}}$ as detailed in Eq. (6). At last, $\overline{d_p}$ is obtained from Eq. (7), with $\widehat{D}$ the estimation of mean distance between L_k and the detected line Fig. 9.

$$\overline{\triangle_\beta d_j p} = \frac{1}{n} \sum_{j=1}^{n} \triangle_\beta d_j \tag{4}$$

$$\overline{\varepsilon_{\alpha_p}} = \arctan\left(\frac{T}{\overline{\triangle_\beta d_i p}}\right) \tag{5}$$

Symbol model	$S_M = \bigcup_{\forall k}\left(L_k, \alpha_k, \bigcup_{\forall i} p_i\right)$	a symbol model
	L_k , α_k	line (or stroke) model k and its local orientation
	$\bigcup_{\forall i} p_i$	the sampled points composing L_k
Line mapping	$S_M = \left(L_k, \bigcup_{\forall p} Mh_p\right)$	S_M is the mapping set of L_k with Mh_p the mapping hypothesis
	$Mh_p = \bigcup_{\forall j} p_j, \overline{\beta_p}, \overline{d_p}, \overline{\gamma_p}$	a mapping hypothesis
	$\bigcup_{\forall j} p_j$	the sampled points composing Mh_p
	$\overline{\beta_p}, \overline{d_p}, \overline{\gamma_p}$	The "mean" orientation and distance between L_k and the detected line on drawing, and its local orientation estimation

Figure 8. Mapping model

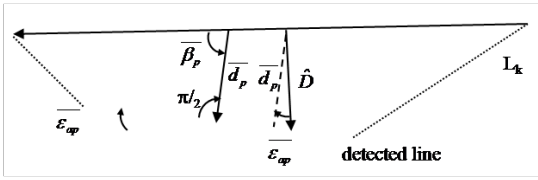

Figure 9. Computation of mapping features $\overline{\beta_p}, \overline{d_p}, \overline{\gamma_p}$

$$\overline{\gamma_p} = \alpha + \overline{\varepsilon_{\alpha_p}} \qquad \overline{\beta_p} = \frac{\pi}{2} + \overline{\varepsilon_{\alpha_p}} \tag{6}$$

$$\overline{d_p} = \widehat{D} \times \cos(\overline{\varepsilon_{\alpha_p}}) \tag{7}$$

$$\widehat{D} = \frac{1}{n}\sum_{j=1}^{n} d_j \sin(\widehat{\alpha\beta_j})$$

3) Matching: Matching is based on the mapping hypothesis and their associated features. The matching algorithm achieves for every symbol model a line mapping, then the best mapping is the one resulting in the smallest set of mapping hypothesis. We determinate final alignment parameters as the scalar product of $\overline{\beta_p}, \overline{d_p}$ features, as defined in Eq (6). The aligned symbol model is displayed to users as shown in Fig 10. The implicit validation of symbol is done when the user releases the tactile screen. Otherwise the matching process is repeated and display results are refreshed.

$$\left(\widetilde{\beta_p}, \widetilde{d_p}\right) = \frac{1}{n \times m}\sum_{k=1}^{n}\sum_{p=1}^{m}\left(\overline{\beta_p}, \overline{d_p}\right) \tag{8}$$

E. Conclusion

In this paper, we have proposed a complete framework for semi-automatic groundtruthing for performance evaluation of symbol recognition & spotting systems. This one uses a mixture of auto-processing steps and human inputs

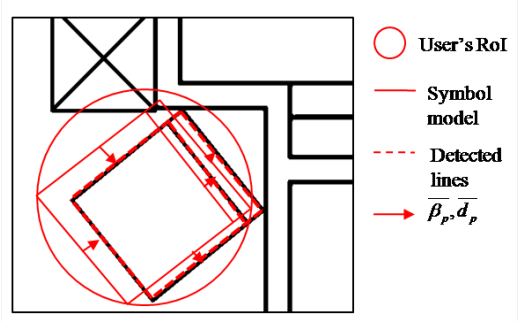

Figure 10. Matching display

based on a tactile device. It employs a top-down matching algorithm, to make the recognition process less sensitive to context information. In addition, it deals with the automatic positioning of symbols to support graphics primitives export. The proposed algorithm is partially invariant to scale and rotation change, constraining users only in rough definition of RoI. The full process works with a sub-linear complexity, allowing like this a fluent user-interaction.

References

[1] M. Delalandre, E. Valveny, and J. Lladós, "Performance evaluation of symbol recognition and spotting systems: An overview," in *Workshop on Document Analysis Systems (DAS)*, 2008, pp. 497–505.

[2] M. Rusiñol and J. Lladós, "A performance evaluation protocol for symbol spotting systems in terms of recognition and location indices," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 12, no. 2, pp. 83–96, 2009.

[3] P. Dosch and E. Valveny, "Report on the second symbol recognition contest," in *Workshop on Graphics Recognition (GREC)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3926, 2006, pp. 381–397.

[4] W. Huang, C. Tan, and J. Zhao, "Generating ground truthed dataset of chart images: Automatic or semi-automatic?" in *Workshop on Graphics Recognition (GREC)*, ser. Lecture Notes in Computer Science (LNCS), vol. 5046, 2007, p. 266 277.

[5] A. Fischer and al, "Ground truth creation for handwriting recognition in historical documents," in *International Workshop on Document Analysis Systems (DAS)*, 2010, pp. 3–10.

[6] M. Okamoto, H. Imai, and K. Takagi, "Performance evaluation of a robust method for mathematical expression recognition," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2001, pp. 121–128.

[7] G. D. Baja, "Well-shaped, stable, and reversible skeletons from the (3,4)-distance transform," *Journal of Visual Communication and Image Representation*, vol. 5, no. 1, pp. 107–115, 1994.

[8] D. Popel, "Compact graph model of handwritten images: Integration into authentification and recognition," in *Conference on Structural and Syntactical Pattern Recognition (SSPR)*, ser. Lecture Notes in Computer Science (LNCS), vol. 2396, 2002, pp. 272–280.