Automatic Static/Variable Content Separation in Administrative Document Images

David Aldavert, Marçal Rusiñol, Ricardo Toledo[†]
Computer Vision Center
Dept. Ciències de la Computació
Universitat Autònoma de Barcelona
Email: {aldavert, marcal}@cvc.uab.cat

***Abstract*—In this paper we present an automatic method for separating static and variable content from administrative document images. An alignment approach is able to unsupervisedly build probabilistic templates from a set of examples of the same document kind. Such templates define which is the likelihood of every pixel of being either static or variable content. In the extraction step, the same alignment technique is used to match an incoming image with the template and to locate the positions where variable fields appear. We validate our approach on the public NIST Structured Tax Forms Dataset.**

I. Introduction

Many efforts have been made by companies and institutions in the digital age to get rid of processing information stored in physical paper by shifting their workflows towards electronic information. A paperless working environment present several advantages such as important economic and storage savings, remote accessibility to information, security and environmental progress. However, the fact is that nowadays most of entities still need to process an important portion of their incoming data in paper, image or in the best case scenario, in electronic document formats. In most of the cases, such information comes in an unstructured way, so that an interpretation step is still needed in order to extract in a structured way such data. Manually processing the bulk of incoming documents is a really costly task and the industry and the market needs have led an important amount of research and development in the context of automatic processing such administrative documents.

In the field of Document Image Analysis, many tasks that fall within the digital mailroom paradigm have been addressed, from document classification [1], [2], [3], document flow segmentation [4], routing [5], and information extraction [6], [7], [8], [9]. In particular, the information extraction step, is often based on the definition of templates that help to point out the locations in which an OCR engine has to read the particular fields to extract. Such templates are usually based on the detection of anchor elements that can be easily and steadily extracted from different instances of the same document kind, i.e. they are based on the detection of *static* content that always appear within the documents under study and that can help to locate the position of *variable* information. Prior work such as the methods proposed by Ishitani [7], [8], Peanho et al. [6], Rusiñol et al. [9], Schuster et al. [10] or Santosh et al. [11] often involve a manual intervention of an expert user that assist the system in building such templates.

In this paper we focus in a particular step of the broader information extraction problem which is the task of segmenting which portions of a document image might contain relevant content since they are the ones that change from instance to instance of the same document type. We propose an unsupervised and fully automated process that given several examples of the same document kind is able to produce a probabilistic template that will indicate the likelihood of every pixel of being either static or variable content. In order to do so, we rely on an image alignment algorithm [12], [13] that is able to cope with the deformations that we can find across different document instances. Once this probabilistic template is build, new incoming document images are processed in order to detect the variable zones of those unseen documents. The main advantage of the system is obviously its ability of producing those templates in an automatic fashion, thus eliminating the need of manual intervention. This specially useful in historical collections where the template of the form is usually not available and therefore has to be generated.

Such approach is mainly interesting when dealing with highly structured documents that have a predefined static layout that is later filled by the users with the relevant information. We carried our tests using the public NIST Structured Tax Forms Dataset (SPDB2) [14], but such method could also be applied to other document kinds that also present this particularity of mixing static and variable content such as invoices, contracts, identification documents, and so on...

The rest of this paper is organized as follows. In Section II we present the overview of the problem. Section III is devoted to present the algorithm for aligning two document images from the same class. Then, in Section IV we present our system pipeline. Section V presents the experimental results that we carried and finally conclusions are drawn in Section VI.

II. Overview

We show in Figure 1 a schematic overview of the proposed system. Two different stages are considered. In an offline step, several instances of the same document kind have to be provided to the system. A pairwise alignment step is

[†]Regrettably, Dr. Toledo passed away after the manuscript submission.

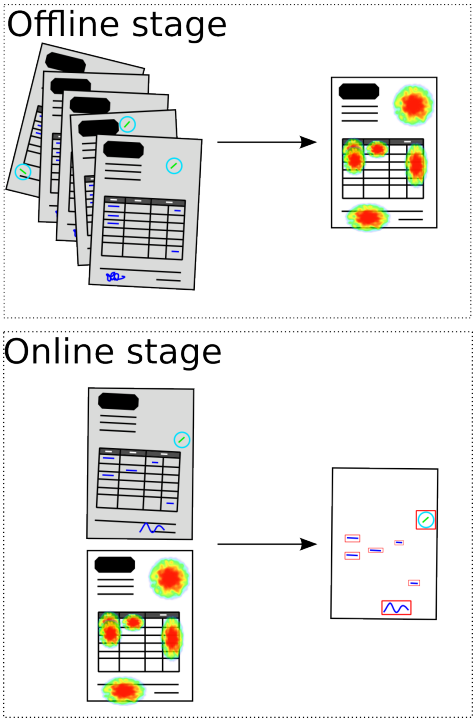


Fig. 1. System Overview

applied in order to cope with distortions that might appear in the digitization process so that all the images are registered together. Once aligned, pixels that are steadily activated as foreground are considered as most probable to be static content whereas pixels that are foreground in some images but not in the others are considered more likely to be variable content. From this "voting" step, a probabilistic template image is automatically produced.

When a new incoming image arrives, it is then aligned with the probabilistic template which is also used in order to weight the foreground pixels. A simple post-processing step is then used in order to end up either with a binary image that only contains variable content or with a set of bounding-boxes in the original image that locate such variable fields.

Let us continue with the details on how the alignment step is performed.

III. Document Alignment

In order to align two document images $\mathbf{I}$ and $\mathbf{T}$, we use the algorithm proposed by Lucas and Kanade [12] to compute the optical flow of the image. This method however has been extensively used in image registration problems, specifically on face registration [15]. The aim of the algorithm is to find the parameters $\mathbf{p}$ that wrap the image $\mathbf{I}$ so it minimizes the differences with the image $\mathbf{T}$,

$$\arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \|\mathbf{I}\left(\mathcal{W}(\mathbf{x},\mathbf{p})\right) - \mathbf{T}(\mathbf{x})\|^2, \tag{1}$$

where $\mathcal{W}(\mathbf{x},\mathbf{p})$ is the wrap function that converts the coordinates $\mathbf{x}$ from the template to the original image reference frame. The complexity of transform $\mathbf{p}$ depends on the model used to relate both images. For example, for face registration problems a non-rigid model like Active Appearance Models [15] is used. However, in our case document images are related at most by an affine transform so $\mathbf{p} = [t_x, t_y, s_x, s_y, s_k, \alpha]$, where t_x and t_y are the translation parameters, s_x and s_y are the scale parameters, s_k is the skew parameter and α is the rotation parameter. Then, the wrap function $\{x', y'\} = \mathcal{W}(\{x, y\}, \mathbf{p})$ becomes,

$$\begin{aligned} x' &= s_x \cos(\alpha)\, x + (s_k \cos(\alpha) + s_y \sin(\alpha))\, y + t_x \\ y' &= -s_x \sin(\alpha)\, x + (-s_k \sin(\alpha) + s_y \cos(\alpha))\, y + t_y. \end{aligned}$$

The algorithm finds the solution to Eq. 1 by iteratively computing the parameters increments $\Delta\mathbf{p}$ that solve

$$\arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \|\mathbf{I}\left(\mathcal{W}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p})\right) - \mathbf{T}(\mathbf{x})\|^2, \tag{2}$$

and updating the parameter $\mathbf{p} = \mathbf{p} + \Delta\mathbf{p}$ until the parameters estimate $\mathbf{p}$ converges. The non-linear expression $\mathbf{I}(\mathcal{W}(\mathbf{x},\mathbf{p}+\Delta\mathbf{p}))$ on Eq. 2 is linearized with its first order Taylor expansion,

$$\arg\min_{\mathbf{p}} \sum_{\mathbf{x}} \|\mathbf{I}\left(\mathcal{W}(\mathbf{x},\mathbf{p})\right) + \nabla\mathbf{I}\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\Delta\mathbf{p} - \mathbf{T}(\mathbf{x})\|^2, \tag{3}$$

where $\nabla\mathbf{I} = (\partial\mathbf{I}/\partial x, \partial\mathbf{I}/\partial y)$ is the gradient of $\mathbf{I}$ evaluated at $\mathcal{W}(\mathbf{x},\mathbf{p})$ and the term $\partial\mathcal{W}/\partial\mathbf{p}$ is the Jacobian of the wrap function which in our case is

$$\frac{\mathcal{W}}{\partial\mathbf{p}} = \begin{bmatrix} 1 & 0 & -xc_\alpha & -ys_\alpha & -yc_\alpha & (y\,s_k + x\,s_x)s_\alpha - ys_yc_\alpha \\ 0 & 1 & xs_\alpha & -yc_\alpha & ys_\alpha & (y\,s_k + x\,s_x)c_\alpha + ys_ys_\alpha \end{bmatrix},$$

where $s_\alpha = \sin(\alpha)$ and $c_\alpha = \cos(\alpha)$. Then, Eq. 3 is solved by computing its partial derivatives and solving the resulting equation, that gives

$$\Delta\mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} \left[\nabla\mathbf{I}\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right]^\top \left[\mathbf{T}(\mathbf{x}) - \mathbf{I}(\mathcal{W}(\mathbf{x},\mathbf{p}))\right], \tag{4}$$

where $\mathbf{H}$ is the Gaussian-Newton approximation to the Hessian matrix

$$\mathbf{H} = \sum_{\mathbf{x}} \left[\nabla\mathbf{I}\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right]^\top \left[\nabla\mathbf{I}\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right]. \tag{5}$$

Summarizing, the algorithm starts with an initial estimation of the parameters, in our case $\mathbf{p} = [0, 0, 1, 1, 0, 0]$ and it keeps updating the parameter estimation $\mathbf{p}$ with Eq. 4 until it converges, i.e. until $|\Delta\mathbf{p}| < \epsilon$, where ϵ is a small value.

This algorithm is sensible to local minima, so several authors replaced the sum of squared errors in Eq. 1 by a more robust estimate of the registration error between the two images like the enhanced correlation coefficient [16], Gabor filters computed in the Fourier domain [17], image gradient [18] or feature-based methods [19]. In our case, we use the original algorithm but we take several pre-processing steps before aligning the document images, so the registration algorithm attains a certain degree of robustness. First, the

images are binarized using the adaptive binarization algorithm proposed by Sauvola and Pietikäinen [20]. The advantages of working with binarized images are twofold: it lessens the effects of illumination and degradation problems in the document images, and it allows to accelerate the registration algorithm by reducing the amount of points that we need to wrap while estimating $\Delta\mathbf{p}$. Then, a connected component analysis step is used to remove regions which are too small to be relevant and large regions which are touching the image margins and are likely to be marginal artifacts. Finally, the module of the gradient is computed over the image in order to reduce the effects of large *foreground* regions. These regions provide a large contribution to the estimation of $\Delta\mathbf{p}$ in Eq. 4 and they can lead to a misalignment between the images when they belong to a dynamic document structure (e.g. a text written with a larger font or a stamp graphic). Using the module of the gradient, we only keep the contour of the *foreground* elements so the influence of large structures on the $\Delta\mathbf{p}$ is reduced. The obtained image is re-binarized with the Otsu algorithm [21].

As we pointed out before, by using binarized images we only have to take into account the *foreground* pixels while wrapping the original image. Therefore, we can reduce the computational cost of the algorithm by only wrapping those pixels. Moreover, the algorithm can be further speeded up by selecting an small random sub-set of the *foreground* pixels. In our case, images are correctly registered using only a 5% of the *foreground* pixels.

The algorithm is sensible to local minima, so when images are related by a large transform it is very likely that the registration algorithm gets *stuck* at a local minima before obtaining the actual registration parameters. Therefore, we follow a multiscale approach where the parameters $\mathbf{p}$ are initially computed using a large sigma at $\nabla\mathbf{I}$ (e.g. with $\sigma = 20$). Then, the parameters $\mathbf{p}$ are recomputed using a smaller sigma using the previous estimation as a *warm start* until we reach $\sigma = 1$. Although this should increase the computationally cost of the algorithm, the algorithm has the same runtime as we use spatial pyramids and recursive Gaussian derivatives [22] to obtain the $\nabla\mathbf{I}$ at each scale, the algorithm converges faster at coarser scales as there are less details and the *warm initialization* greatly reduces the number of iterations of finer scales.

IV. System Description

We use the image registration algorithm from Section III both to align the document images while creating the model template and also to register the model template with a document image when filtering the static parts to obtain the dynamic content of the document.

A. Document Static Model Generation

We have a set of document images which have the same layout as the sample shown in Fig. 2 and we want to obtain an image which contains only the document structures which are common in all documents, i.e. the static elements of the document collection. An example of an obtained model template is shown in Fig. 3.

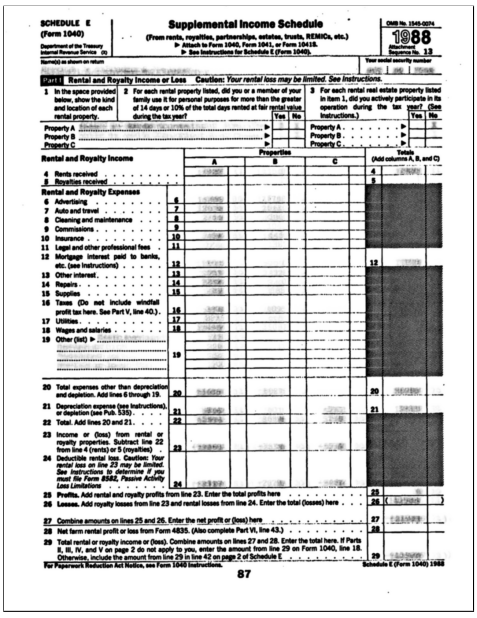


Fig. 3. Probabilistic model template obtained from the document set shown in Fig. 2

In order to obtain such a model, we first have to align all the documents of the collection using the algorithm described in Section III. We can obtain the model by computing all possible image pairs, however following this approach the number of image pairs to be aligned is quadratic respect the number of image of the set. For example, the collection of Fig. 2 has 24 images which means that we have to compute 300 relationships to obtain all the image relationships. Instead, we can randomly select N images which in turn are randomly aligned only with M images of the collection. The resulting model is generated only computing $N \times M$ image pair alignments and although it uses less images, the obtained model does not greatly differ from the obtained using the whole collection. For instance, the model in Fig. 3 has been generated with $N = M = 7$ so only 49 image pairs are aligned.

The document static model $\mathcal{M}$ for the collection of document images $\mathcal{C}$ is generated with two steps: first we generate as set of N partial models $\mathcal{P} = \{\mathcal{P}_k \in \mathcal{C} | k \in \{1, .., N\}\}$ and then we group them into the final document static model $\mathcal{M}$. A partial mode $\mathcal{P}_k$ is created by randomly selecting a base image $\mathcal{X}_k \in \mathcal{C}$ which is then intersected with M randomly selected images $\mathcal{Y}_k = \{\mathcal{Y}_{ki} \in \mathcal{C} | i \in \{1, .., M\}\}$. Like in the previous section, we use the Sauvola and Pietikäinen algorithm [20] to remove illumination and degradation artifacts from the images, so $\mathcal{X}_k$ and $\mathcal{Y}_k$ images are binary. The intersection image $\mathcal{I}_{ki}$ between $\mathcal{X}_k$ and $\mathcal{Y}_{ki}$ is computed as

$$\mathcal{I}_{ki} = \mathbf{G}_\sigma(\mathcal{X}_k) \cdot \mathbf{G}_\sigma(\mathcal{W}(\mathcal{Y}_{ki}, \mathbf{p}_{ki})), \tag{6}$$

where $\mathcal{W}(\mathbf{I}, \mathbf{p})$ applies the affine transform $\mathbf{p}$ to the image

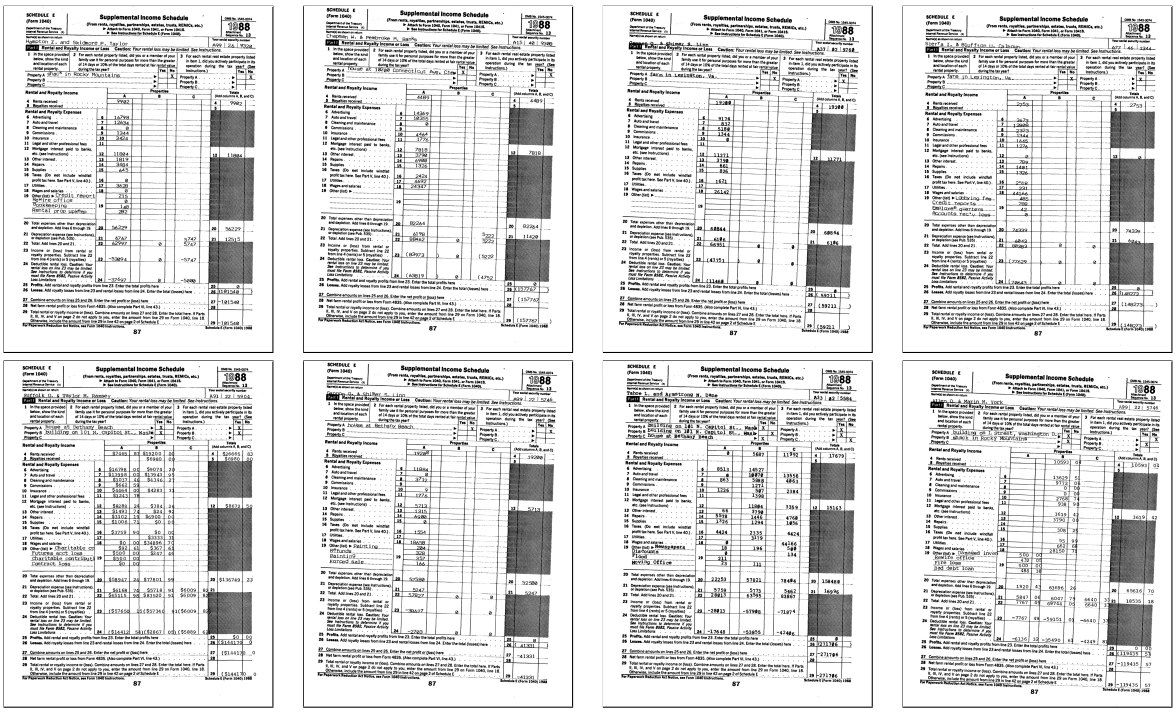


Fig. 2. Sample training documents for a particular class

$\mathbf{I}$, $\mathbf{p}_{ki}$ are the parameters of the transform that align $\mathcal{Y}_{ki}$ to $\mathcal{X}_k$, and $\mathbf{G}_{\sigma=1}$ is a Gaussian filter used to smooth the binary images in order to account for small binarization and misalignment errors. Then, we compute the average between the intersection images as

$$\widetilde{\mathcal{P}_k} = \frac{1}{M} \sum_{i}^{M} \mathcal{I}_{ki}$$

and obtain the partial model $\mathcal{P}_k$ by applying a pixel-wise sigmoid function over $\widetilde{\mathcal{P}_k}$ to increase its contrast. Instead of just accumulating all the intersection among images to generate the model, we apply a sigmoid function which is a non-linear transform that removes low probability contributions that appear in regions where dynamic structures are commonly present. Therefore, these dynamic structures won't receive enough support in the final static model $\mathcal{M}$.

Finally, the static document model is obtained by accumulating all the partial models $\mathcal{P}$ by

$$\widetilde{\mathcal{M}} = \frac{1}{N} \sum_{k}^{N} \mathcal{W}\left(\mathcal{P}_k, \overline{\mathbf{p}_k}\right), \tag{7}$$

where $\overline{\mathbf{p}_k}$ are the parameters that align the k-th partial model to $\mathcal{P}_1$. The selection of $\mathcal{P}_1$ as reference frame is arbitrary and we can select any other partial model as reference. Alternatively, we could also estimate the location of the reference frame by averaging the transforms that relate all partial models. However, the main goal is just to accumulate all partial models into the final model, so the reference frame used is not important. Like with the partial models, the static document model $\mathcal{M}$ is obtained applying a pixel-wide sigmoid function over $\widetilde{\mathcal{M}}$.

B. Dynamic Elements Detection

Once we have generated the static model $\mathcal{M}$ for the collection $\mathcal{C}$, we can remove the static parts of the image $\mathcal{Q} \in \mathcal{C}$ by

$$\mathcal{S} = \mathcal{Q} \cdot \mathbf{D}(\mathcal{W}(\mathcal{M}, \widetilde{\mathbf{p}})), \tag{8}$$

where $\mathbf{D}(\cdot)$ is a dilation operator of 5×5 used to widen the wrapped model and $\widetilde{\mathbf{p}}$ are the parameters of the affine transform that aligns the model $\mathcal{M}$ to the query image $\mathcal{Q}$. The contrast of the resulting image is improved by applying a pixel-wise sigmoid function over $\mathcal{I}$.

In order to find the dynamic element regions, we binarize $\mathcal{S}$ by simply applying a low threshold (e.g. activate the pixels that have a probability above 0.25) and merging the detected regions by applying a morphological opening with a rectangular structuring element. The regions which does not have enough support, have unlikely shapes (e.g. are too thin) or have highly intersect with another region are filtered out.

V. Experimental Results

In order to carry out our experiments we used the NIST Structured Tax Forms Dataset (SPDB2) [14]. We have selected a subset of 15 document classes, which are the ones that we have enough document image samples of the same type. For each of those 15 classes, 24 images where selected to build the probabilistic templates in the offline stages. A single example for each class is then used as testing image to assess the quality of the content extraction.

TABLE I
CONTINGENCY MATRIX

		True condition		
		Positive	Negative	Total
Predicted	Positive	466 (TP)	12 (FP)	478
	Negative	17 (FN)	–	17
	Total	483	–	

First, we present in Figure 4 some qualitative results. We show three different test images along with the produced probability maps for the variable field extraction. Here darker values indicate a higher pixel probability of being a variable field and brighter values indicate a higher probability of being static content. Pixels belonging to the static part are almost unperceptible here. We also show the final segmentations of variable content, and we can appreciate that some false alarms appear either due to binarization noise (which in some sense is a variable element) and to highly textured zones.

In order to quantitatively evaluate our system, we manually groundtruthed all the variable fields in the 15 test images to see at which extend the proposed methodology is able to locate such fields and at which extend it also delivers false alarms. In total the ground-truth is composed of 483 fields to extract. We can see the results in form of a contingency matrix in Table I. Here, the true positive condition are all the variable fields, whereas the true negative condition is not quantized since it is the rest of the document (static content). When we run our method, we end up correctly retrieving 466 of the variable fields (true positives) while missing 17 of them (false negatives) and providing 12 erroneous segmentations in zones where there are no variable elements (false positives). In summary, the proposed method yielded a precision of 97.49% and a recall of 96.48% in the task of retrieving variable fields.

VI. CONCLUSIONS

In this paper we have presented an automatic method for separating static and dynamic content that appear in administrative document images. Such method allows to define probabilistic templates aimed at locating locations of relevant fields without the need of an expert user intervention. We have validated our approach on the public NIST Structured Tax Forms Dataset, and we plan to make further tests on other administrative documentation such as invoices, contracts, id cards, etc.

ACKNOWLEDGMENTS

This work was supported by the Spanish project TIN2014-52072-P and by the CERCA Programme / Generalitat de Catalunya. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

REFERENCES

[1] P. Héroux, S. Diana, A. Ribert, and E. Trupin, "Classification method study for automatic form class identification," in *Proceedings of the Fourteenth International Conference on Pattern Recognition*, 1998, pp. 926–928.

[2] N. Chen and D. Blostein, "A survey of document image classification: problem statement, classifier architecture and performance evaluation," *International Journal on Document Analysis and Recognition*, vol. 10, no. 1, pp. 1–16, June 2006.

[3] M. Rusiñol, V. Frinken, D. Karatzas, A. Bagdanov, and J. Lladós., "Multimodal page classification in administrative document image streams," *International Journal on Document Analysis and Recognition*, vol. 17, no. 4, pp. 331–341, December 2014.

[4] M. Rusiñol, D. Karatzas, A. Bagdanov, and J. Lladós, "Multipage document retrieval by textual and visual representations," in *Proceedings of the International Conference on Pattern Recognition*, 2012, pp. 521–524.

[5] P. Viola, J. Rinker, and M. Law, "Automatic fax routing," in *Proceedings of the International Worksop on Document Analysis Systems: Document Analysis Systems VI*, ser. Lecture Notes on Computer Science, 2004, vol. 3163, pp. 484–495.

[6] C. Peanho, H. Stagni, and F. da Silva, "Semantic information extraction from images of complex documents," *Applied Intelligence*, vol. 37, no. 5, pp. 543–557, December 2012.

[7] Y. Ishitani, "Model-based information extraction method tolerant of OCR errors for document images," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2001, pp. 908–915.

[8] ——, "Model-based information extraction and its applications for document images," in *Proceedings of the Workshop on Document Layout Interpretation and its Applications*, 2001.

[9] M. Rusiñol, T. Benkhelfallah, and V. dAndecy, "Field extraction from administrative documents by incremental structural templates," in *Proceedings of the 12th International Conference on Document Analysis and Recognition*, 2013.

[10] D. Schuster, K. Muthmann, D. Esser, A. Schill, M. Berger, C. Weidling, K. Aliyev, and A. Hofmeier, "Intellix – end-user trained information extraction for document archiving," in *Proceedings of the 12th International Conference on Document Analysis and Recognition*, 2013.

[11] K. Santosh and A. Belaïd, "Pattern-based approach to table extraction," in *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, 2013, pp. 766–773.

[12] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, 1981, pp. 674–679.

[13] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, February 2004.

[14] D. Dimmick, M. Garris, and C. L. Wilson, "Structured forms database," National Institutte of Standards and Technology, Tech. Rep., 1991.

[15] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, November 2004.

[16] G. Evangelidis and E. Psarakis, "Parametric image alignment using enhanced correlation coefficient maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, October 2008.

[17] S. Lucey, R. Navarathna, A. B. Ashraf, and S. Sridharan, "Fourier lucas-kanade algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1383–1396, June 2013.

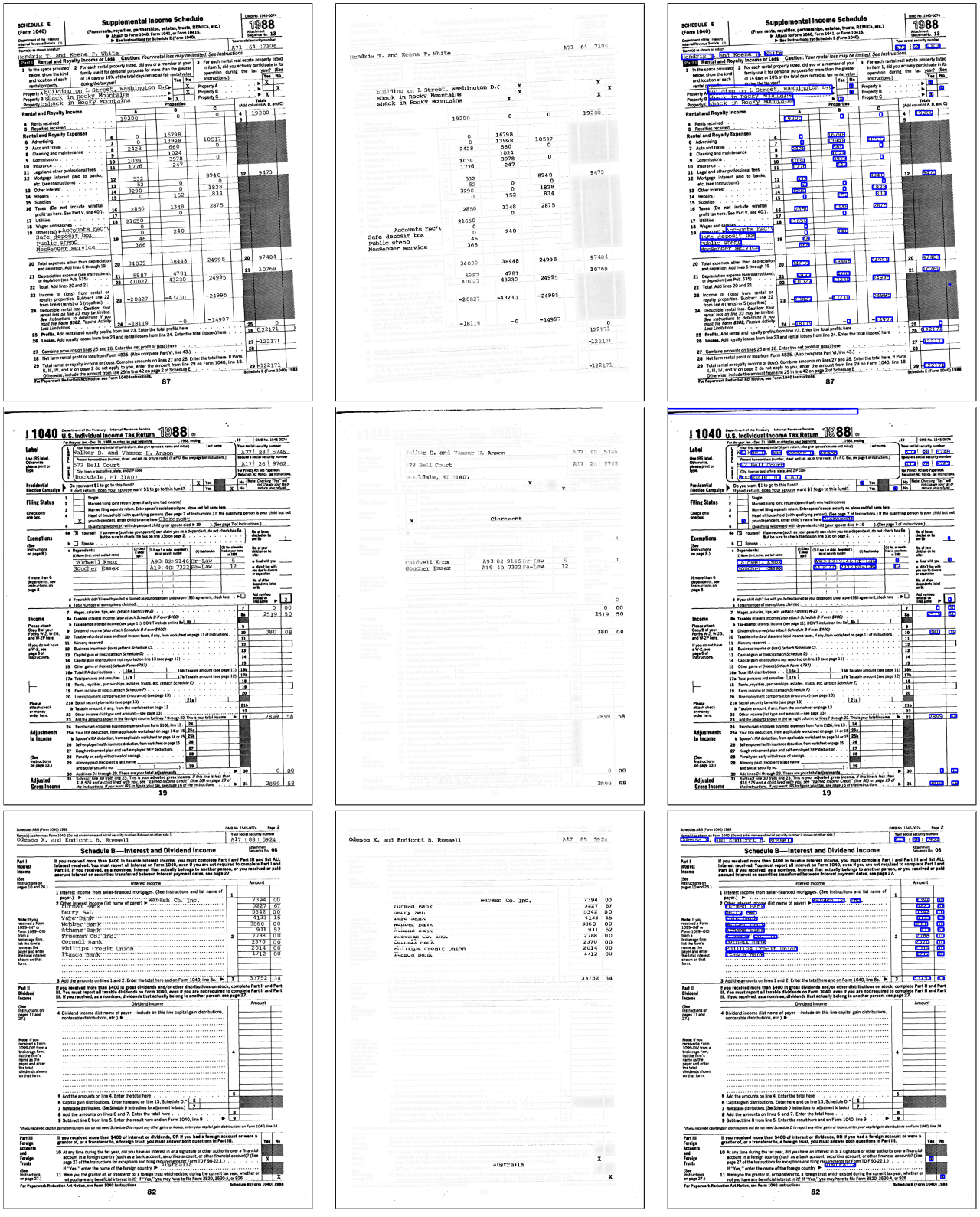
[18] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Robust and efficient parametric face alignment," in *IEEE International Conference on Computer Vision*, 2011, pp. 1847–1854.

[19] E. Antonakos, J. A. i Medina, G. Tzimiropoulos, and S. Zafeiriou, "Feature-based lucas-kanade and active appearance models," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2617–2632, September 2015.

[20] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, February 2000.

[21] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, January 1979.

[22] L. van Vliet, I. Young, and P. Verbeek, "Recursive gaussian derivative filters," in *Proceedings of the International Conference on Pattern Recognition*, 1998, pp. 509–514.



a) b) c)

Fig. 4. Detection results. a) Test images, b) Probability maps of the variable elements after aligning the test documents with their respective templates, c) Final bounding-box extractions.