20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Using Agglomerative Clustering of Strokes to Perform Symbols Over-segmentation within a Diagram Recognition System

Martin Bresler, Daniel Průša, Václav Hlaváč
Czech Technical University in Prague, Faculty of Electrical Engineering
Department of Cybernetics, Center for Machine Perception
166 27, Praha 6, Technická 2, Czech Republic
{breslmar, prusapa1, hlavac}@cmp.felk.cvut.cz

Abstract. *Symbol segmentation is a critical part of handwriting recognition. Any mistake done in this step is propagating further through the recognition pipeline. It forces researchers to consider methods generating multiple hypotheses for symbol segmentation – over-segmentation. Simple approaches which takes all reasonable combinations of strokes are applied very often, because they allow to achieve high recall rates very easily. However, they generate too much hypotheses. It makes a recognizer considerably slow. This paper presents our experimentation with an alternative method based on a single linkage agglomerative clustering of strokes with trainable distance metric. We embed the method into the state-of-the-art recognizer for on-line sketched diagrams. We show that it results in a decrease in the number of generated hypotheses while still reaching high recall rates. A problem emerges, since the number of bad hypotheses is still significantly higher than the number of symbols and it leads to unbalanced training datasets. To deal with it, we propose to train symbol classifiers with synthesized artificial samples. We show that the combination of these two improvements make the recognizer significantly faster and very precise.*

1. Introduction

Free hand writing and especially drawing are very natural ways how people express their thoughts. Devices allowing users to write and draw with a stylus directly on the surface of a displaying unit became very common. This functionality is in tablets, tablet PCs, or smart white boards. There is a great interest in systems capable to recognize this so called *ink input*. It is also called an on-line input and it is considered to be a sequence of handwritten strokes, where a stroke is a sequence of points captured by a device. Every point is always defined by its coordinates in the plane (drawing canvas). Additional data like a time stamp and a pressure value is usually provided as well. An output of a recognizer is a formal description of the input.

The research in handwritten document analysis and processing has moved from recognition of plain text to recognition of more structured inputs such as mathematical formulas, chemical formulas, music scores, or diagrams. Several recognizers of e.g. mathematical formulas with a good precision were presented in recent years [1, 9, 14]. Moreover, there is a contest in recognition of mathematical expressions [12]. In contrast, availability of diagram recognizers is still limited. The reason might be that there exists a vast of different diagrammatic domains, some of them not being well specified as mathematical formulas. However, there has been an effort to develop recognizers for electric circuits [8], chemical drawings [13], or flowcharts [5].

Although we showed that there exist numerous recognition systems specialized on various domains, they all face a common problem of symbols identification. Symbol segmentation is a crucial part of handwriting recognition where symbols are located in the input so they can be classified later. Ideally, the segmentation output would be disjoint subsets of strokes covering all the strokes. However, the segmentation can be barely done properly without knowledge of the whole structure. In practice, it is not wise to make hard decisions in this early step of the pipeline. A better approach is to perform so called *over-segmentation*. It supplies a larger number of subsets which typically share some strokes.

The final decision, which subsets fits the structure of the input diagram best, is left to the later phases performed by a structural analyzer.

It is important to achieve a high recall rate by the segmentation, which means that there are subsets of strokes representing ideally all of the symbols. Usually, simple over-segmentation methods based on intuitive assumptions that symbols comprise of strokes which are spatially and temporarily close are used. It considers all possible sequences of strokes up to some size. The segments are created iteratively and their number is limited by a maximal number of strokes and also by thresholds saying what is the maximal allowed distance or time difference between strokes in a segment. Variants of described approach are used in all the systems we introduced. Although it can achieve a high recall rate, it usually induces a very poor precision, because it simply generate too many bad hypotheses. Their consideration followed by rejection makes the whole recognition process time consuming.

We designed a diagram recognition system which uses exactly this approach to achieve the over-segmentation [4]. In this paper, we investigate different options which would allow to achieve still high recall rates and generate significantly less segmentation hypotheses and thus to increase the precision. Delaye and Lee [7] showed that objects of interest may be found using Single-Linkage Agglomerative Clustering (SLAC). It is a hierarchical bottom-up clustering where two closest clusters are merged together in each step until there is only one cluster remaining. Singleton clusters consisting of a single stroke are created first and bigger clusters are created iteratively. A link is created at each merging step and it contains information about two clusters it links and a distance between them. The resulting tree structure is called dendrogram and we can get the desired clusters by defining a suitable threshold to cut the tree. For illustration see Figure 1. In case of single-linkage the distance between two clusters is given by the distance between their two closest elements. The tricky part is to find a suitable measure defining a distance between two strokes. The authors use a set of simple features which basically express differences in geometric, spatial, and temporal characteristics of two strokes. The distance between two strokes is given by a weighted sum of these features. Obviously, each feature has different importance and thus it is necessary to find a suitable weights. They proposed an algorithm which is able to train the weights automatically from annotated data. The algorithm finds the best threshold to cut the dendrogram as well. They tested the whole approach on several domains and showed that this approach can find well defined symbols in flowcharts (FC), finite automata (FA), or mathematical expressions as well as loosely defined text blocks and figures in free hand sketches.

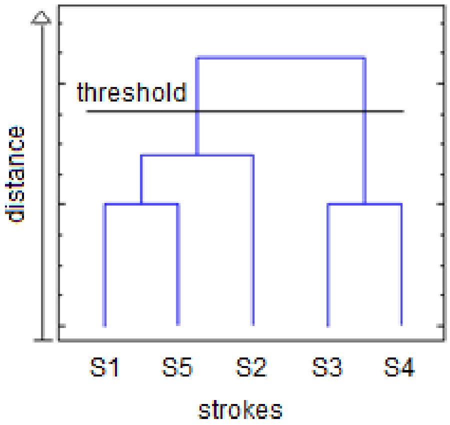
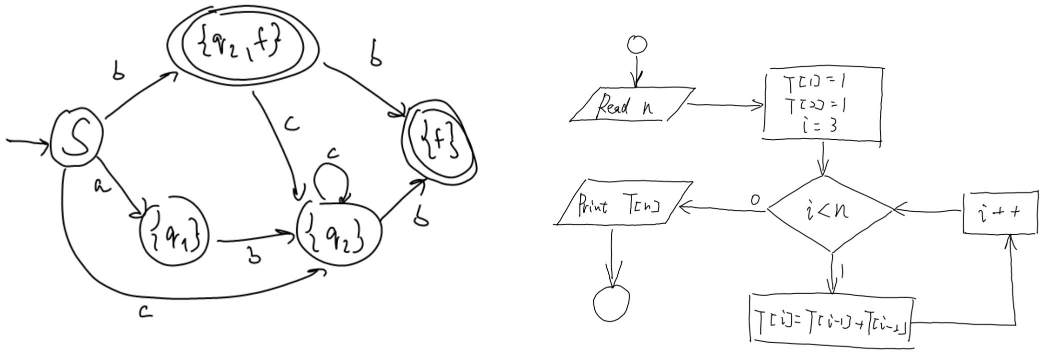


Figure 1: Illustrative example of a dendrogram and its cutting.

Delaye [6] later tried to classify the segmented clusters into corresponding symbol classes. He upgraded the proposed segmentation tool into a diagram recognizer. It is based on Conditional Random Fields (CRF), where the created clusters represent nodes of the graph. The author creates hierarchical model by applying several values of the threshold. The created graphs have a tree structure and thus the problem can be solved efficiently by the Belief Propagation algorithm. It makes the system extremely fast. However, it is a purely statistical approach which gives no further information about the diagram structure and it may produce inconsistent labelings.

There exist benchmark databases for FC [2] and FA [4] domains. We embed the SLAC method proposed by Delaye and Lee into our recognition system and compare the new results with our previous version of the system. We compare it with other two systems – the statistical recognizer by Delaye [6] and the grammar base recognizer by Carton e al. [5]. Examples of diagrams from the two mentioned database are shown in Figure 2.

The rest of the paper is organized as follows. We briefly describe our diagram recognizer in Section 2.



(a)

(b)

Figure 2: Examples of diagrams from the two domains – (a) finite automata (FA) and (b) flowcharts (FC).

The proposed improvements are presented in Section 3. Experiments with the improved recognizer follow in Section 4. Finally, we make a conclusion in Section 5.

2. Diagram Recognition System

We have developed a recognition system for online sketched diagrams [4]. It is a general system so far adopted for domains of flowcharts and finite automata. The system consists of several steps of the recognition pipeline, which is depicted in Figure 3. The first one is the input normalization where the points are resampled to remove those points that are too close to each other. Text strokes are removed from the input by the text separator, which is based on the algorithm for mode detection by Phan and Nakagawa [16]. The removed text strokes are going to be put back later when the diagram structure is recognized to form text blocks attached to symbols. Next step is the symbols candidates detection. It is done by over-segmentation and classification of the created groups of strokes. Symbols are divided into two types: *uniform symbols* with relatively stable appearance and *non-uniform arrows* with significantly varying appearance. The recognition is done in two steps. Uniform symbols are found first and arrows are detected afterwards as connectors linking pairs of uniform symbols. Uniform symbols are classified by an SVM classifier based on the trajectory based normalization and direction features proposed by Liu and Zhou [10]. We detect arrows with recently proposed arrow detector based on LSTM RNN classifier [3]. The core of the recognition pipeline is the structural analysis phase. Individual symbol candidates have a score assigned saying how good the hypothesis is without considering any context. Some of the symbol candidates are in relations. Binary predicates are defined to indicate if two symbol candidates can coexist in the solution together or if one symbol candidate can be a part of the solution without the other one, etc. The selection of the best subset of symbol candidates is cast as an optimization problem where the goal is to maximize the sum of scores of selected symbol candidates that fulfil all the constraints given by the predicates. We model this framework as a pairwise max-sum labeling problem. Finally, remaining unused text strokes form text block which can be easily found with the knowledge of the diagram structure.

3. Proposed System Improvements

We propose two improvements of the recognition system. First, we replace the naive strokes grouping by the SLAC. Second, we improve the symbol classifiers by using synthesized samples.

3.1. Over-segmentation Improvement

The old method works with an important assumption that symbols are formed of strokes which are spatially and temporarily close. Strokes grouping is done iteratively. Within the first iteration, every single stroke forms a subset of size 1. Further, subsets of size k are created by adding a single spatially and temporarily close stroke to subsets of size $k-1$. Maximal size of a strokes group k must be derived from knowledge of a domain and it affects a number of generated groups. Threshold used to determine if two strokes are spatially and temporarily close must be derived form data. The advantage of this approach is its simplicity and possibility to achieve 100 % recall using the right parameters. The disadvantage is

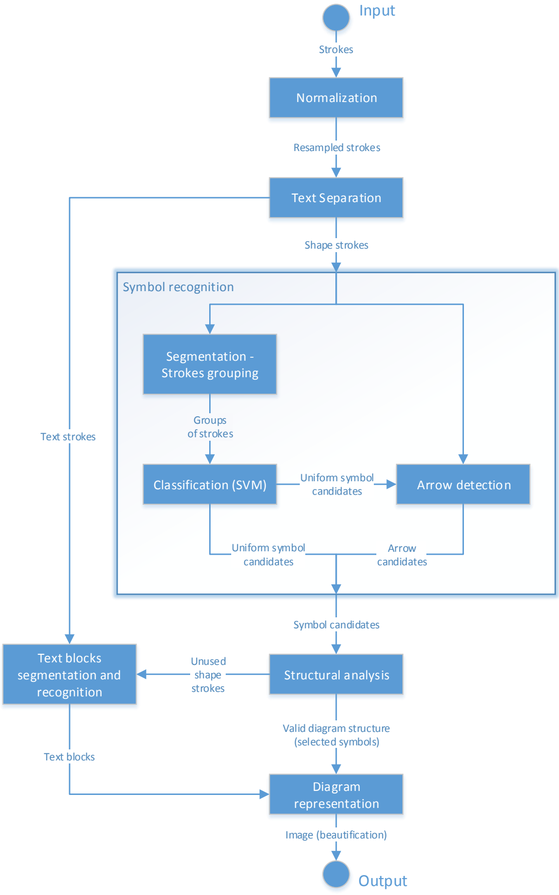


Figure 3: The pipeline of our recognition system.

the fact that the method considers too many combinations of strokes, which are not important, because they can never form a symbol. This inefficiency led us into experimentation with other possibilities.

Single-linkage clustering based on weighted combination of several features with trainable parameters proposed by Delaye and Lee [7] reaches very high precision. Its advantage is that it uses more features combined together and can express more complex relations between strokes than just the Euclidean distance. Another advantage is that single-linkage is a fast clustering algorithm. Time complexity is quadratic in the number of strokes. It is only needed to compute distance between individual strokes once and then the distance between two clusters is given by the distance of their two closest strokes. We reimplemented the method and trained the feature weights and the threshold as is described in the work by Delaye and Lee. We achieved a bit worse precision (cca. 3 % less) on both, FC and FA, databases. We believe that it is caused by slight differences in the input normalization. However, the method is powerful and the result is satisfactory for our purposes.

We perform the clustering with the trained parameters and several values of the threshold to perform an over-segmentation to increase the recall. We obtain various values of the threshold by multiplication of the original threshold h by a changing coefficient c_i: $h_i = h \cdot c_i$. We use various values of c_i from the interval $[c_{min}, c_{max}]$ with step 0.01, where the bounds c_{min} and c_{max} must by found in a validation step. Only uniform symbols are our objects of interest, because our recognition system deals with text and arrows separately. We used the validation dataset of the FA database and train dataset of the FC database to find the bounds of the coefficient. We tried all combinations of c_{min} from the range $[0.1, 1.0]$ and c_{max} from the range $[1.0, 2.0]$. The best combination of the bounds is that one which gives the highest recall. In case that more combinations give the same recall, a combination giving higher precision is taken. We found out that for both domains the best values are $c_{min} = 0.5$ and $c_{max} = 1.2$.

3.2. Improvement of the Symbol Classifier

As we care for the greatest possible universality of our system, we used the most general approach and combined trajectory based normalization and direction features proposed by Liu and Zhou [10] as a descriptor with multiclass classifier implemented as an instance of a structured output SVM learned by BMRM algorithm [15]. We trained the classifier with negative examples to obtain the rejection ability. Dataset of symbols for training has been obtained by applying the over-segmentation implemented as the multi-threshold SLAC. If a group of strokes is annotated as a uniform symbol in the database, it is labeled by that symbol. Otherwise it is labeled as *no_match* which denotes a negative example. Arrows as well as incomplete parts of symbols are labeled as negative examples.

The number of negative examples is much higher than the number of uniform symbols. Moreover, they are greatly inhomogeneous. It is thus necessary to cluster them into several subclasses. We employed k-means base on the descriptor to create m *no_match* subclasses, where m is domain dependent ($m = 30$ for flowcharts, $m = 20$ for finite automata). A greater amount of symbol classes in the flowchart domain results in a greater m. This brings a need

Database – Method	Retrieved	Relevant	Matched	Recall	Precision	F-measure
FC – grouping	19 714	921	878	95.33 %	4.45 %	0.085
FC – clustering	5 245	921	876	95.11 %	16.70 %	0.284
FA – grouping	6 095	488	485	99.39 %	7.96 %	0.147
FA – clustering	1 838	488	487	99.78 %	26.50 %	0.419

Table 1: The results of strokes grouping and clustering on the test datasets of the FC and the FA databases.

for a modified loss function which gives zero penalty when a negative example is classified into a different *no_match* subclass. Additionally, a greater penalty is required for misclassification of a uniform symbol as a negative example than vice versa. The ratio between these two penalties depends also on the ratio between the number of uniform symbols and negative examples. A properly chosen loss function can overcome the problem with unbalanced database, as we showed in [4]. However, our current implementation uses artificially synthesized samples to balance the database. The samples were synthesized using the approach proposed by Martín-Albo et al. [11]. It is based on Kinematic Theory and the distortion of the Sigma-Lognormal parameters in order to generate human-like synthetic samples. We generated up to 20 artificial samples from each uniform symbol taken from the training dataset. From all the synthesized samples of one class we randomly chose a subset to get the desired number of symbols for training. This approach does not only help to balance the dataset, it also supplies additional information on handwriting and makes the classifier more robust. Therefore, we empirically set the smaller penalty to 1 and the bigger penalty to 2 just to increase recall in cost of very small decrease of precision. Unfortunately, the FC database does not contain any information about time – points forming strokes are defined by coordinates only. Since time information is crucial for the synthesization, artificial samples could not be obtained for this database.

4. Experiments

We performed two types of experiments. We report a comparison of the results of the naive strokes grouping and more sophisticated strokes clustering first. We show how the clustering method allows to increase the precision significantly while the recall changes minimally. Later we show how this improvement affects the overall performance of the system. It turns out that time complexity is significantly lowered.

4.1. Strokes Grouping vs. Strokes Clustering

Note that the text separation step precede the over-segmentation step and thus the most of the text strokes are removed. The text separator achieves the precision in *shapes*/*text* class of 99.62 %/94.76 % and 100.00 %/93.31 % for FC and FA, respectively. Since the over-segmentation is used to find uniform symbols only, we do not consider text blocks or arrows as relevant objects. Therefore there are 921 / 488 relevant objects in the test dataset of the FC / FA databases. Results of both over-segmentation methods on both databases are summarized in Table 1. Notice that the clustering method achieved even higher recall than the naive grouping in the case of FA. Obviously, few symbols in the test dataset violated one of the assumptions we use in the process of strokes grouping. Specifically, they comprise of more strokes than is allowed. The advantage of the clustering approach is that we do not need such assumption at all.

4.2. Overall System Performance

We changed the over-segmentation method in the recognition pipeline and made experiments with diagram recognition. We use two standard metrics for system quality assessment – correct strokes labeling and correct symbol segmentation and recognition. We compare the results with the published results of our former system [3], with the grammar based method by Carton et al. [5], and with the purely statistical method by Delaye [6]. Our system achieved the highest precision using both metrics on both domains. For details, see Tables 2, 3. The precision slightly increased in the case of FA and slightly decreased the case of FC. However, the main benefit of the new over-segmentation method is the difference in the performance in the term of the running time. Our system is implemented in C# and we ran the experiments on a standard tablet PC Lenovo X230 (In-

Class	Correct stroke labeling [%]				Correct symbol segmentation and recognition[%]			
	Carton	Delaye	WACV 2015	Proposed	Carton	Delaye	WACV 2015	Proposed
Arrow	83.8	–	88.7	87.5	70.2	–	78.1	76.6
Connection	80.3	–	94.1	94.1	82.4	–	95.1	95.1
Data	84.3	–	96.4	95.3	80.5	–	90.6	90.5
Decision	90.9	–	90.9	88.2	80.6	–	75.3	72.9
Process	90.4	–	95.2	96.3	85.2	–	88.1	88.6
Terminator	69.8	–	90.2	90.7	72.4	–	88.9	89.0
Text	97.2	–	99.3	99.2	74.1	–	89.7	89.5
Total	**92.4**	**93.2**	**96.5**	**96.3**	**75.0**	**75.5**	**84.4**	**84.2**

Table 2: Recognition results for the FC database. We compared the proposed system with the grammar based method by Carton et al. [5], with the statistical method by Delaye [6], and with our previous work presented at WACV 2015 [3].

Class	Correct stroke labeling [%]			Correct symbol segmentation and recognition[%]		
	Delaye	WACV 2015	Proposed	Delaye	WACV 2015	proposed
Arrow	–	94.9	98.0	–	92.8	97.5
Initial arrow	–	85.0	98.6	–	84.0	97.3
Final state	–	99.2	99.2	–	98.4	99.2
State	–	96.9	98.3	–	97.2	98.2
Label	–	99.8	99.7	–	99.1	99.2
Total	**98.4**	**97.4**	**99.0**	**97.1**	**96.4**	**98.5**

Table 3: Recognition results for the FA database. We compared the proposed system with the statistical method by Delaye [6] and with our previous work presented at WACV 2015 [3].

tel Core i5 2.6 GHz, 8GB RAM) with 64-bit Windows 7. Detailed results with performance of all the systems are in Table 4. We reduced the running time significantly and made the system useful for a real-time applications. However, the purely statistical approach by Delaye is much faster. On the other hand, the author probably used more optimized implementation and more powerful machine, because our system spent more time on feature extraction solely than his system did on the whole recognition.

5. Conclusion

Naive over-segmentation approach considering all combination of spatially and temporarily strokes is simple and achieves a very high recall. It is possible to apply several restrictions like maximal number of strokes in a segment to suppress the number of created segmentation hypotheses. However, the number of generated hypotheses is still too big and the precision is limited. Even though the symbol classifier can reject most of the hypotheses in the early stage, it might be still time consuming.

System	FC	FA
Carton [5]	1.94 s	-
Delaye [7]	80.8 ms	52.0 ms
WACV 2015 [3]	1.06 s	2.03 s
proposed	0.78 s	0.69 s

Table 4: Average running time for diagram recognition by different systems.

We experimented with over-segmentation method based on agglomerative clustering of strokes. It creates hypotheses in a smarter way, avoiding the consideration of all strokes combinations. We combined clusters obtained by cutting the dendrogram with various thresholds. It allows to increase the recall at the cost of decreased precision. However, the achieved precision is still much higher than in the case of naive strokes grouping and the recall is comparable. This approach generally does not lead to 100 % recall even when all possible values of the threshold are tried. The reason is that all threshold values always produce nested clusters. Their characteristics is given by the set of used distance features, sets of their weights, and the principle of the single-linkage clustering itself. Different clustering methods could be probably combined together to further increase the recall. An intuitive idea is to combine together other agglomerative clustering methods like average or complete linkage. Unfortunately, this methods have higher time complexity than single linkage. However we leave this for a future work.

Acknowledgements

The first author was supported by the Grant Agency of the CTU under the project SGS13/205/OHK3/3T/13. The second and the third authors were supported by the Czech Science Foundation under the project P103/10/0783.

References

[1] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí. Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. *Pattern Recognition Letters*, 35(0):58 – 67, 2014. Frontiers in Handwriting Processing. 1

[2] A.-M. Awal, G. Feng, H. Mouchere, and C. Viard-Gaudin. First experiments on a new online handwritten flowchart database. In *DRR'11*, pages 1–10, 2011. 2

[3] M. Bresler, D. Průša, and V. Hlaváč. Detection of arrows in on-line sketched diagrams using relative stroke positioning. In *WACV 2015: IEEE Winter Conference on Applications of Computer Vision*, pages 610–617. IEEE Computer Society, January 2015. 3, 5, 6

[4] M. Bresler, T. Van Phan, D. Průša, M. Nakagawa, and V. Hlaváč. Recognition system for on-line sketched diagrams. In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 563–568, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 2, 3, 5

[5] C. Carton, A. Lemaitre, and B. Couasnon. Fusion of statistical and structural information for flowchart recognition. In *International Conference on Document Analysis and Recognition (ICDAR), 2013 12th*, pages 1210–1214, 2013. 1, 2, 5, 6

[6] A. Delaye. Structured prediction models for online sketch recognition. Unpublished manuscript, `https://sites.google.com/site/adriendelaye/home/news/unpublishedmanuscriptavailable`, 2014. 2, 5, 6

[7] A. Delaye and K. Lee. A flexible framework for online document segmentation by pairwise stroke distance learning. *Pattern Recognition*, 2014. 2, 4, 6

[8] G. Feng, C. Viard-Gaudin, and Z. Sun. On-line hand-drawn electric circuit diagram recognition using 2D dynamic programming. *Pattern Recogn.*, 42(12):3215–3223, Dec. 2009. 1

[9] A. D. Le, T. Van Phan, and M. Nakagawa. A system for recognizing online handwritten mathematical expressions and improvement of structure analysis. In *11th IAPR International Workshop on Document Analysis Systems (DAS), 2014*, pages 51–55, April 2014. 1

[10] C.-L. Liu and X.-D. Zhou. Online Japanese Character Recognition Using Trajectory-Based Normalization and Direction Feature Extraction. In G. Lorette, editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France), Oct. 2006. Université de Rennes 1, Suvisoft. 3, 4

[11] D. Martín-Albo, R. Plamondon, and E. Vidal. Training of on-line handwriting text recognizers with synthetic text generated using the kinematic theory of rapid human movements. In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 543–548, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 5

[12] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain. ICFHR 2014 Competition on Recognition of On-line Handwritten Mathematical Expressions (CROHME 2014). In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 791–796, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 1

[13] T. Y. Ouyang and R. Davis. Chemink: A natural real-time recognition system for chemical drawings. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 267–276, New York, NY, USA, 2011. ACM. 1

[14] J. Stria, D. Průša, and V. Hlaváč. Combining structural and statistical approach to online recognition of handwritten mathematical formulas. In Z. Kúkelová and J. Heller, editors, *CVWW2014: Proceedings of the 19th Computer Vision Winter Workshop*, pages 103–109, Pod Vodárenskou věží 4, 182 00, Prague, Czech Republic, February 2014. Czech Society for Cybernetics and Informatics, Center of Machine

Percep tion at CTU in Prague, Czech Society for Cybernetics and Informatics. 1

[15] C. H. Teo, A. J. Smola, and Q. V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010. 4

[16] T. Van Phan and M. Nakagawa. Text/non-text classification in online handwritten documents with recurrent neural networks. In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 23–28, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 3