The Role of Grouping in Sketched Diagram Recognition

Amirhossein Ghodrati
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
a.ghodrati@massey.ac.nz

Rachel Blagojevic
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
r.v.blagojevic@massey.ac.nz

Hans W. Guesgen
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
h.w.guesgen@massey.ac.nz

Stephen Marsland
School of Mathematics and Statistics
Victoria University of Wellington
Wellington, New Zealand
stephen.marsland@vuw.ac.nz

Beryl Plimmer
Department of Computer Science
University of Auckland
Auckland, New Zealand
b.plimmer@auckland.ac.nz

ABSTRACT

An early step in bottom-up diagram recognition systems is grouping ink strokes into shapes. This paper gives an overview of the key literature on automatic grouping techniques in sketch recognition. In addition, we identify the major challenges in grouping ink into identifiable shapes, discuss the common solutions to these challenges based on current research, and highlight areas for future work.

CCS CONCEPTS

•**Human-centered computing** → *Human computer interaction (HCI);*

KEYWORDS

Sketch recognition; digital ink recognition; digital ink grouping; digital ink segmentation

ACM Reference format:

Amirhossein Ghodrati, Rachel Blagojevic, Hans W. Guesgen, Stephen Marsland, and Beryl Plimmer. 2018. The Role of Grouping in Sketched Diagram Recognition. In *Proceedings of The Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, Victoria, BC, Canada, August 17–19, 2018 (Expressive '18),* 12 pages.

DOI: 10.1145/3229147.3229160

1 INTRODUCTION

Hand-drawn diagrams are frequently used for externalizing ideas and documenting existing phenomena. Pen and paper offers an unconstrained space for diagramming, is quick to use, and allows for ambiguity. The availability of stylus input to the computer can offer similar advantages to pen and paper, with added benefits, such as ease of replication, storage, and sharing. In addition, the automatic recognition of these diagrams to create a computational model can allow for even greater advantages, such as intelligent editing, automated translation to alternative forms, and execution or animation of sketch models.

A diagram recognition engine typically takes primitives (such as lines and curves, or strokes) as input and outputs the higher level meaning of the diagram. Performing recognition often requires a pipeline consisting of some or all of the following steps: separating the sketch into writing and drawing primitives; grouping primitives and recognising basic shapes; identifying domain-specific components; and finally inferring the semantics of the entire sketch. In this paper, we refer to grouping as the process that decides which primitives belong to the same shape. The shape recogniser then identifies a group of primitives as a particular shape.

Grouping primitives into potential shapes is not a trivial problem. The naïve approach of examining all possible combinations of primitives in the diagram, would require exponential time for the number of strokes. To limit the search space of possible combinations, heuristics based on spatial and/or temporal proximity can be used. For a systematic user, these heuristics may be sufficient to identify the individual groups, since objects are often drawn sequentially and in a clearly delineated position. However, this is not always the case, and there may not be clear spatial or temporal boundaries between elements due to a user's interspersed drawing style (where a user starts drawing a new shape before completing the previous one) [Sezgin and Davis 2007b], or highly connected components (e.g., the circuit diagram in Figure 1), where shapes are in very close proximity to one another or even overlapping.

Early methods avoided the problem of grouping completely by placing constraints on how objects should be drawn. For example, users have been restricted to drawing each shape with a single stroke [Herold and Stahovich 2012; Plimmer et al. 2012; Reaver et al. 2011; Rubine 1991; Wobbrock et al. 2007], or asked to provide cues such as clicking a button, or pausing for a period of time, in order to show that the current shape is finished [Hse and Newton 2005]. Other systems require a temporally contiguous sequence of strokes to be drawn for a shape [Gennari et al. 2005]. Although such constraints simplify the grouping process, they do little to

Expressive '18, Victoria, BC, Canada

978-1-4503-5892-7/18/08…$15.00
DOI: 10.1145/3229147.3229160

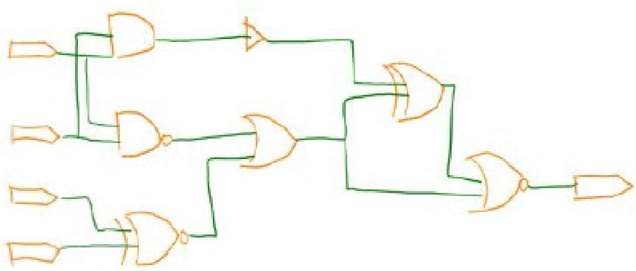


Figure 1: Example of digital circuit diagram [Stevens et al. 2013]

preserve a flexible, free-sketch environment, that follows on from the user experience of diagramming with pen and paper.

Accuracy is the overarching problem at every stage in diagram recognition engines. Given that grouping occurs quite early on in the recognition process, it is even more important that it is performed accurately, as any errors will cause a snowball effect in subsequent parts of the system. There are trade-offs between accuracy, computation time, and providing a free-sketch environment when designing a recognition engine. The effect of these trade-offs on grouping will be discussed further in Section 5.

In this paper we review the approaches to grouping thus far and consider their role in sketched diagram recognition. Our analysis focuses on highlighting the key characteristics of these approaches; where the literature is used to illustrate these characteristics.

There is a potential chicken-and-egg problem with grouping and shape recognition [Arandjelović and Sezgin 2011; Peterson et al. 2010]. Identifying a group of primitives that form a shape requires recognition of the shape, while recognising a shape may require that the correct group of primitives are presented to the recogniser. Therefore, one of the two main approaches identified is to perform grouping and recognition simultaneously. Techniques for simultaneous grouping and recognition are described in Section 2. An alternative approach is to follow a sequence and perform the grouping independently of the shape recognition. This is typically based on either a clustering method that groups primitives together, or a sequential optimisation that optimise a cost function; these methods are reviewed in Section 3. In addition, there are techniques that do not follow these pipelines which are discussed in Section 4. In our analysis of these approaches, we focus on the characteristics of grouping, rather than details on shape recognition approaches used with grouping. In Section 5 we discuss the aspects of grouping that have been successful and identify future challenges. To the best of our knowledge, this is the first paper to provide an in-depth analysis of grouping techniques applied to sketched diagram recognition.

2 SIMULTANEOUS GROUPING AND RECOGNITION

Many groupers use a shape recogniser as a way of guiding the search process to only group viable shape candidates. We have identified three conceptual approaches from the literature, based on how they use the information from the recogniser. The first of these are negative example methods that use the recogniser to reject potential shapes that are not recognised. The second approach is based on grammars and languages, which work by defining a logical grammar to describe acceptable shapes. The third technique is optimisation, where the temporal order of shapes are captured for grouping primitives. We have also included a section on other simultaneous methods that do not fall into the previous categories. The general process of simultaneous grouping and recognition is shown in Figure 2. Domain knowledge (displayed in dashed box) is mainly used in the grammar and language based approaches.

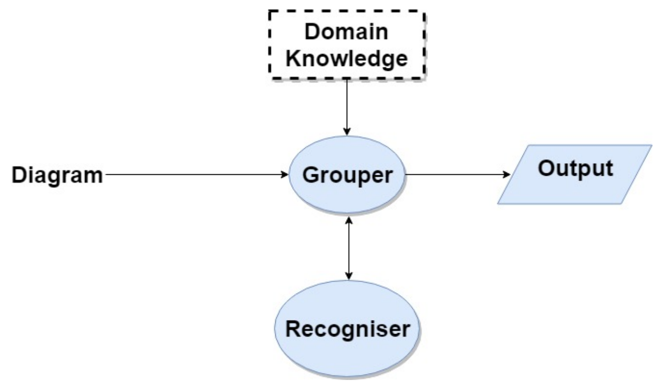


Figure 2: Simultaneous grouping and recognition process

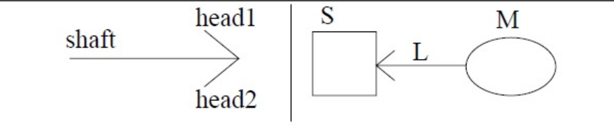
2.1 Negative Examples

The most common method of enabling a classifier to reject shape candidates is to include the sub-shapes that should be ignored in the training set, but with a 'no-match' label given to them. This label can be applied to several sub-classes of shapes. The simultaneous grouping and recognition process then becomes a method for selecting candidate shapes using a classifier that returns either a class label representing a shape, or the 'no-match' label.

Candidate shapes were selected using all possible sets of up to five spatially proximal primitives in [Bresler et al. 2013b] for the flowchart dataset. To reduce the number of shape candidates produced in [Bresler et al. 2013b], [Bresler et al. 2015b] later used the Single Linkage Agglomerative Clustering (SLAC) algorithm [Delaye and Lee 2015] for grouping. In all cases, a multiclass Support Vector Machine (SVM) was used as the recogniser, although in [Bresler et al. 2015a] a recurrent neural network (LSTM) was used as a post-processing step for arrow detection.

Since the negative examples would not equip a recogniser to accurately reject invalid candidates, context information was used along with the recogniser's output to select a final set of shape candidates by optimizing a cost function [Bresler 2016; Bresler et al. 2013a, 2014; Ouyang and Davis 2007] or maximizing the joint probability [Ouyang and Davis 2009]. In addition, domain knowledge was used in [Ouyang and Davis 2007] to check whether the structure is chemically sound for the domain of molecular compounds.

As these methods consider all subsets of spatially related primitives up to some maximum size, they have a large search space. In addition, the recognisers need to learn about very large 'no-match' classes, which requires a large amount of data, since there are many ways that negative examples can be presented. Selecting the final



DEFINE ARROW
(**Subshapes**
L1,L2,L3: (Line *shaft head1 head2*))
(**Constraints**
C1: (coincident *shaft.p1 head1.p1*)
C2: (coincident *shaft.p1 head2.p2*)
C3: (equal-length *head1 head2*)
C4: (shorter *head1 shaft*)
C5: (acute-angle *head1 shaft*)
C6: (acute-angle *head2 shaft*)
DEFINE MOTHER-SON
(**Subshapes**
M,S,L: (Female *M*) (Male *S*) (Child-link *L*)
(**Constraints**
C1: (touches *L.head S*)
C2: (touches *L.tail M*))

Figure 3: The description of the shape "arrow" in family tree domain [Alvarado and Davis 2004]

set of shape candidates through optimization techniques is a computationally expensive task, which is another downside of these approaches.

A summary of work using negative examples can be found in Table 1. The accuracy column in this table reports the statistics from each paper. Some papers report the accuracy for grouping alone and others are for grouping and recognition. The dataset column shows the datasets that are used for training and testing. The interspersed column shows whether interspersed is allowed or not. The context column summarizes the information about how neighbouring strokes/shapes influence the classification/interpretation of a shape. The same scheme is used for the next tables. The domain independent column gives the information whether the approach is applicable to a specific domain or is a domain agnostic approach.

2.2 Grammar and Language

Grammar and language techniques have been explored in a wide range of sketch recognition tasks including grouping of ink. These approaches typically define a language to describe shapes and their constraints, and use this language as a key part to grouping primitives and recognising the groups (see the summary in Table 1). Variations on the form of languages exist, such as in [Alvarado and Davis 2004] where a hierarchical shape language is used (see Figure 3 for the definition of an arrow in the family tree domain), while in [Julca-Aguilar et al. 2017] a graph grammar generates potential shape hypotheses and their possible relations. Groups are constructed based on how well they conform to the shape descriptions, where the descriptions rely heavily on spatial and temporal proximity.

Groups are further evaluated in various ways. In [Costagliola et al. 2005] a parser generates multiple parse trees, each providing a possible interpretation of the sketch. Each tree is assigned a probability based on the fitting error of strokes and the accuracy of shape relationships. The best tree with the highest probability is chosen as the output. In SketchREAD [Alvarado and Davis 2004] a Bayesian Network evaluates the strongest interpretation of the diagram. Given the descriptions and constraints, the system also tries to find the missing parts from partially drawn shapes by checking spatially and temporally close strokes. Further improvements to SketchREAD have been made [Alvarado and Davis 2006] using dynamically constructed Bayesian networks to determine how well each shape hypothesis fits the data. In [Julca-Aguilar et al. 2017] a classifier first prunes the search space by rejecting groups using it's level of confidence, this may include groups representing negative example classes. The remaining shape hypotheses are then chosen by optimising a cost function that considers the likelihood score of shapes and their relationships.

Computational time is still a significant issue for these approaches, particularly if interspersed drawing is allowed. Hammond and Davis explore these issues in [Hammond and Davis 2009], where they propose a grouping method that examines all possible shape combinations, but use an indexing technique to reduce computation time. They perform a stress test to analyse the running time of the system, and the results indicate that it runs close to real-time with 100% recognition accuracy. However, the authors mention that in the worst case, the algorithm is still exponential in terms of computation time. Also, the recognition is severely limited to the constraints defined by the language. Further details of the nature of the dataset used, and what constitutes the worst case scenario for computation time is not clear in the paper.

The main limitation of these methods are that the language must be defined by an expert, for each domain. It is also difficult to encode the levels of ambiguity in these languages that we know are present in sketched diagrams. The accuracy of these approaches is therefore limited by the difficulties in defining a language.

2.3 Optimisation

In optimisation approaches, grouping and recognition are performed simultaneously, and a model of the primitives is optimised to produce the groups and labels for those groups. The main technique demonstrated in the literature is to use time-based models [Sezgin and Davis 2005, 2007a, 2008] to group and recognise shapes i.e. those that only consider temporal information.

In [Sezgin and Davis 2005] the temporal order of shapes are modelled with different Hidden Markov Models (HMMs). To interpret a diagram, a graph showing the temporal order of primitives is produced, with the addition of edges that represent possible shape candidates (weighted with the log-likelihood of it matching a known shape). The shortest path between the first and last primitive (node) of the diagram is used to determine the optimal grouping and recognition of shapes, where each edge in the shortest path represents a valid shape. In later work [Arandjelović and Sezgin 2011], a similar approach was used to construct the graph. However in this work, the optimal shape candidates are determined using dynamic programming, with information combined from time-based and image-based recognizers. In addition, a one-class SVM classifier is used to reject invalid shape candidates. Sezgin and Davis also

Table 1: Summary of simultaneous grouping and recognition techniques - Int. = Interspersed strokes allowed, D.I. = Domain Independent

	Proximity	Int.	Dataset	Accuracy	Context	Limitations	D.I.
[Bresler et al. 2013b]	Spatial	Yes	Flowchart [Ahmad-Montaser Awal 2011]	91.9% in grouping/recognition	N/A	Limited size of strokes per shape	Yes
[Bresler et al. 2013a]	Spatial, temporal	Yes	Flowchart [Ahmad-Montaser Awal 2011]	88.7% in grouping	Uses relations between shapes		Yes
[Bresler et al. 2014]	Spatial, temporal	Yes	Flowchart dataset [Ahmad-Montaser Awal 2011], FA (Finite Automata)	82.8% in grouping/recognition of flowchart dataset; 94% in grouping/recognition of FA dataset	Uses relations between shapes	Limited to arrow connected diagrams	Yes
[Bresler et al. 2015b]	Spatial, temporal	Yes	flowchart dataset [Ahmad-Montaser Awal 2011], FA [Bresler et al. 2014]	95.33% in grouping and 84.2% in grouping/recognition of flowchart dataset; 99.39% in grouping and 95.5% in grouping/recognition of FA dataset	N/A	Computation time is quadratic in the number of strokes - Requires fine tuning the negative class	Yes
[Ouyang and Davis 2007]	Temporal	No	Molecular compounds	85% in grouping/recognition without domain knowledge, 89% in grouping/recognition with domain knowledge	Examines nearby shapes and applies domain specific constraints	Relatively low recognition rate without domain knowledge	Yes
[Ouyang and Davis 2009]	Spatial, temporal	Yes	Molecular diagrams, Electrical circuit diagrams	97% grouping/recognition in Molecular diagrams; 91% grouping/recognition in Electrical circuit diagrams	Using graphical models to capture shape relations	Computationally expensive training and recognition process	Yes
Language and Grammar							
[Costagliola et al. 2005]	Spatial, temporal	Yes	Not tested	Not tested	Shape relations are used in the grammar	Requires shape definition, hard to extend	No
[Alvarado and Davis 2004]	Spatial, temporal	Yes	Family tree, digital circuit	77% grouping/recognition for family tree; 62% grouping/recognition for digital circuit diagram	The likelihood of an interpretation of a stroke is influenced by surrounding shapes	Requires structural definition of shapes and their relationships for a specific domain	No
[Hammond and Davis 2009]	Spatial, temporal	Yes	Japanese Kanji, Military course of action, Biology diagrams	100% for in all diagrams	Shape relations are defined in the language	Requires shape definition	No
[Julca-Aguilar et al. 2017]	Spatial, temporal	Yes	Mathematical expression, Flowchart dataset [Ahmad-Montaser Awal 2011]	85.5% accuracy in grouping/recognition on flowchart dataset	Relations between sub-components or shapes are modeled in the grammar	Relies on the grammar	No
Optimisation							
[Sezgin and Davis 2005]	Temporal	No	88 objects from different domains including military course of action, stick-figures, and mechanical engineering	96.5% recognition rate	None	Relies on temporal pattern of how objects are drawn	Yes

Table 1: Summary of simultaneous grouping and recognition techniques (continued)

[Arandjelović and Sezgin 2011]	Temporal	No	Course of Action full diagrams	36.3% accuracy in grouping and recognition of full diagram (partial recognition of a diagram is not counted)	None	Relies on temporal pattern of how objects are drawn - Limited temporal window	Yes
[Sezgin and Davis 2007a]	Temporal	No	Circuit diagrams	Ranging from 77.4% to 93% for different users	By capturing object-level pattern	Relies on temporal pattern of how objects are drawn	Yes
[Sezgin and Davis 2008]	Temporal	Yes	Circuit diagrams	Ranging from 87.7% to 95.6% for different users	By capturing object-level pattern	Computationally expensive	Yes
Others							
[Hammond and Paulson 2011]	Spatial	Yes	Military course of action, circuit diagrams, chemistry diagrams, mechanical engineering diagrams	89.7% weighted accuracy in grouping/recognition	None	Computationally expensive	Yes
[Johnston and Alvarado 2013]	Spatial, temporal	Yes	Digital circuit diagrams	94.2% accuracy in grouping	Domain specific checks are required	Limited to spatial bounding box size	Yes

extended the work of [Sezgin and Davis 2005] in [Sezgin and Davis 2007a], but in this approach a dynamic Bayesian network is used where object-level patterns (the sequence of drawn objects) are considered as well as the temporal ordering of primitives. An extra node is added for each primitive to denote if the shape is complete at that point. The model then tries to maximize the joint likelihood of stroke-level and object-level patterns, resulting in an optimal grouping and recognition of the sketch. Since these approaches only rely on temporal order, interspersed drawing cannot be supported. This issue has been resolved for grouping in [Sezgin and Davis 2008] by considering an additional "switching node" (MUX) for each observation which indicates whether the user has interspersed any two objects, but drawing order is still considered when recognising shapes.

2.4 Other Simultaneous Methods

There are also other approaches that perform grouping and recognition simultaneously (see Table 1).

In [Hammond and Paulson 2011] all strokes are first given to PaleoSketch [Hammond and Davis 2009], a low-level recognizer capable of recognizing nine primitives (e.g. lines, arcs and curves). From the recognized primitives, a neighbourhood graph is constructed and searched for connected components using Tarjan's algorithm. To find valid shapes, all sub-graphs of each connected component are recognised using the neural network version of PaleoSketch [Paulson and Hammond 2008]. The group must then pass a false-positive removal stage in order to be deemed a correct grouping. The main limitation of this approach is that it is computationally expensive and has a grouping and recognition accuracy of 89.7%.

In [Johnston and Alvarado 2013] a neighbourhood graph is constructed by placing a bounding box around each stroke to locate spatially close strokes. Connected components of the graph are found by performing a series of breadth-first-searches. All possible subsets of each connected component (up to size 5) are evaluated by calculating the Hausdorff distance of the subset to template shapes. Domain-specific checks are carried out on the best candidate using an encoding language before finalizing a shape. This approach handles interspersed drawings up to a certain window size, and is also limited by the use of a bounding box in the initial grouping stages. However, it has performed well on digital circuit diagrams, producing a recognition rate of 94%.

3 SEQUENTIAL GROUPING AND RECOGNITION

These approaches follow a pipeline of grouping and recognition as separate tasks occurring sequentially and independently. We have further categorised these sequential approaches into optimisation and clustering techniques. The general processes of sequential grouping and recognition are shown in Figure 4 and Figure 5. Figure 4 shows the pipeline of Sequential optimisation and Hard-clustering approaches. Hard-clustering approaches do not have the optimisation task (showed in dashed box). Figure 5 also shows the pipeline of the PGM-based recognition system.

3.1 Sequential Optimisation

Sequential optimisation approaches generally follow a sequence of group-recognise-optimise. Spatial/temporal information about the primitives are used to generate possible shape candidates, each candidate is recognised, and finally the best set of shape candidates is chosen by optimising a given cost function (see Table 2 for a summary of work using sequential optimisation).

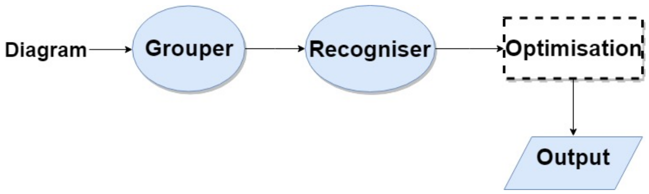


Figure 4: Sequential optimisatoin based and hard-clustering based grouping and recognition process

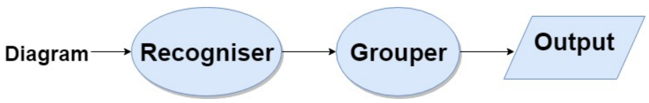


Figure 5: Sequential PGM-based grouping and recognition process

One common way of representing the spatial proximity heuristic is to construct a neighbourhood graph of strokes, with the strokes being nodes, and edges between them representing spatial contiguity (within some pre-defined threshold). Using this graph it is possible to perform grouping by searching over the set of all connected components of the graph of size at most k, where each component is a possible shape candidate. [Shilman et al. 2004] used this approach and defined a cost function for optimization based on the shape recognition cost (which is itself made up of the classification confidence and some context information), the combination cost of including two subsets of the graph, and the constraint cost that checks the connectivity of subgraphs. This approach was demonstrated on a mathematical expression dataset in [Shilman et al. 2004] and extended to recognize flowcharts and mixtures of text and graphics in [Shilman and Viola 2004]. In both cases boosting was used to classify each potential object, but in [Shilman and Viola 2004] A^* search was used to prune away parts of the search space that could not lead to a viable solution. The experimental results demonstrate that the proposed system can achieve 97% grouping/recognition for 13 shape classes collected from 19 different users, although these are isolated shapes rather than full diagrams. On full flowchart diagrams [Shilman and Viola 2004] reports an 85% grouping/recognition rate.

In a similar manner to [Shilman and Viola 2004; Shilman et al. 2004], [Feng et al. 2009] generates shape candidates based on spatial proximity and maximum size, but with two additional constraints. Temporal proximity is considered by allowing a maximum number of time jumps between primitives within a group, to account for the interspersed nature of sketching. Also, a maximum overlap ratio is used as an extra spatial constraint, where groups with significant overlap of primitives are merged as they are considered part of the same shape. Each shape hypothesis is first checked to see if it is a connector using Least Square Fitting Error (LSFE). If the candidate is not a connector then a Neural Network is used for the classification. A cost function based on the shape resemblance and the connectivity requirements of the shape is then optimized. This approach is specifically designed for circuit diagrams.

In [Ouyang and Davis 2011], candidate shapes are again generated using temporal and spatial proximity. Recognition and optimisation of the solution is performed using Conditional Random Field (CRF). The CRF model used here constructs a hierarchical model that combines features measured from three levels: raw ink points, primitives and candidate shapes. The joint model captures the relationship between these levels. Each primitive has a shape candidate node which represents all possible shape candidates that includes that primitive. The inference process selects the best shape candidate (with the maximum likelihood) for each primitive. The results show that the proposed algorithm is able to detect and classify 97.4% of shapes in a chemical drawing dataset. However, the inference process is computationally expensive.

Although optimisation approaches have yielded reasonable recognition rates, the search and optimisation process is still computationally expensive.

3.2 Clustering-Based Grouper

Clustering-based groupers typically take every pair of primitives in a diagram and capture pairwise features of these primitives to form shape candidates, regardless of what the shape is. Then each shape candidate is given to a recognizer for identification. We further separate these into those that use hard clustering methods, and those that use Probabilistic Graphical Models (PGMs).

3.2.1 Hard clustering. Clustering is the task of grouping objects that have common characteristics [Tan et al. 2005]. In [Delaye and Lee 2015] a grouping method based on the Single Linkage Agglomerative Clustering (SLAC) algorithm is proposed. The SLAC algorithm successively merges the two closest clusters unless the distance between the closest clusters is more than a threshold. The closeness of the clusters is calculated through a distance function that captures the pairwise stroke spatial and temporal distances. The proposed algorithm is evaluated on various datasets including flowcharts, finite automata, mathematical notation, synthesized diagram from these datasets and free-form documents with the accuracy ranging from 75% to 95.55% for different datasets.

In order to solve the problem of exhaustive search in traditional methods, a grouping technique has been designed with two levels of classification [Peterson et al. 2010]. This strategy reduces the problem of exhaustive search into a classification problem. The initial step is the single stroke classification that classifies strokes into coarse classes depending on the context. For example, in [Peterson et al. 2010] each stroke is classified as wire, gate, or text for the domain of digital circuit diagrams. The next step is to perform the grouping using the coarse classifications. [Peterson et al. 2010] compares two methods of grouping, one uses a simple threshold based on the spatial distance and the elapsed time between two strokes, and the other approach uses the AdaBoost classifier trained with 13 pairwise features to classify each pair as: "don't join", "far join" or "near join". In an iterative process, pairs of strokes with "far join" or "near join" are clustered together. In [Stevens et al. 2013] 11 new pairwise features are added which has led to an increase in the grouping accuracy. However, in [Lee et al. 2012] authors found only 6 features that are introduced in [Peterson et al. 2010] were

Table 2: Summary of sequential grouping and recognition techniques - Int. = Interspersed strokes allowed, D.I. = Domain Independent

	Proximity	Int.	Dataset	Accuracy	Context	Limitations	D.I.
[Shilman et al. 2004]	Spatial	Yes	Mathematical expression	94% in grouping/recognition	The neighbouring strokes of a shape candidate to help the recognizer spot invalid shape candidates	Computationally expensive	Yes
[Shilman and Viola 2004]	Spatial	Yes	Isolated shapes of HHreco [Hse and Newton 2004], synthesized flowchart diagrams	90% on grouping and 85% on grouping/recognition of flowchart diagrams, 97% in grouping/recognition of HHReco	The neighbouring strokes of a shape candidate to help the recognizer spot invalid shape candidates	Computationally expensive	Yes
[Feng et al. 2009]	Spatial, temporal - constrained to max # of time jumps	Yes	Digital circuit, the Neural Network is separately trained on 451 collected isolate shapes	90.29% accuracy in class-level recognition	The connectivity constraints are employed to aid the rejection of pseudo-shapes (either containing strokes from multiple shapes or an incomplete shape)	Computationally expensive and limited to circuit diagrams	No
[Ouyang and Davis 2011]	Temporal, spatial	Yes	Molecular diagram	97.4% in grouping and recognition	Modeling the spatial compatibility between pairs of segments allows the neighbouring interpretations to influence each other	Employs domain knowledge about chemical structures. Computationally expensive	Yes
Clustering: Hard Clustering							
[Delaye and Lee 2015]	Spatial, temporal features	Yes	Flowchart dataset [Ahmad-Montaser Awal 2011], FA [Bresler et al. 2014], free-form online document, mathematical expressions	88% in grouping for flowchart dataset - 98% for FA dataset	None	Requires fine-tuning the threshold for different datasets	Yes
[Peterson et al. 2010]	Spatial and temporal features	Yes	Circuit diagrams - Family tree diagram	91.4% grouping accuracy for circuit diagrams - 86% grouping accuracy for family tree diagrams	None	Requires expensive training process - Accuracy	Yes
[Stevens et al. 2013]	Spatial and temporal features	Yes	Class diagram - Digital circuit diagram - Family tree - Flowchart	Overall 80.96% accuracy in grouping	None	Requires expensive training process - low accuracy	Yes
[Stahovich et al. 2014]	Spatial and temporal features	Yes	Digital circuits - Family tree	91% grouping accuracy in digital circuits - 86% grouping accuracy in family tree	None	Requires expensive training process - Accuracy	Yes
[Alvarado 2007]	Spatial	Yes	Digital circuit	77% in grouping	Using contextual information for single stroke classification	Low accuracy	Yes

Table 2: Summary of sequential grouping and recognition techniques (continued)

Clustering: PGM-based							
[Delaye 2014]	Spatial and temporal features	Yes	flowchart diagram [Ahmad-Montaser Awal 2011], FA [Bresler et al. 2014]	80.56% in grouping and 75.50% in grouping/recognition of flowchart dataset - 98.15% in grouping and 97.12% in grouping/recognition of FA dataset	The neighboring strokes affect the label prediction of each other	Computationally expensive inference process - Low accuracy for flowchart dataset	Yes
[Deufemia et al. 2014]	Spatial, temporal	Yes	Electric circuit diagrams	Ranging between 81.3% and 91% in grouping and recognition	The spatial and temporal relationships among shapes	Computationally expensive inference process	Yes
[Wang et al. 2016]	Spatial, temporal	Yes	flowchart dataset [Ahmad-Montaser Awal 2011], FA [Bresler et al. 2014]	85.2% in grouping and 84.3% in grouping/recognition of flowchart dataset - 95.8% in grouping/ recognition of FA dataset	Strokefis relations are modeled in the MRF	Computationally expensive inference process - Low accuracy for flowchart dataset	Yes

useful for training a C4.5 decision tree to determine if two strokes belong to the same character for equilibrium equation recognition.

[Stahovich et al. 2014] point out that their original grouping classifier [Peterson et al. 2010] often confuses the "far join" class with "near join" and "no join" classes. To achieve a higher accuracy in grouping, they propose two new training approaches that improve the classifier's performance. One is the Minimum Distance (IPC-MD) approach that uses numerical thresholds to assign stroke-pair labels to the training data. Through this method, strokes from different objects are labelled as "no join" and strokes from the same object are labelled as "near join" if some numerical thresholds are met. The remaining pairs are labelled as "far join". The other approach, Iterative Relabelling (IPC-IR), uses cluster accuracy to iteratively re-label the training data. The iterative labelling uses the IPC-MD method to assign label to pairs of strokes and iteratively optimizes the accuracy of the classifier. The experiments show that the difference between IPC-MD and IPC-IR is not significant for digital circuit diagrams, but both approaches outperform the thresholding method proposed in [Peterson et al. 2010]. However, in the domain of family tree diagrams, the thresholding method outperforms both the IPC-MD and IPC-IR methods. For engineering statics problems the IPC-MD technique significantly outperforms the thresholding method when equations are included, but the thresholding technique achieves a higher accuracy when equations are excluded from the dataset. This grouping approach is employed in a language and grammars based sketch recognition system [Costagliola et al. 2014, 2015] that uses local context for recognition.

In a similar manner of two level classification, [Alvarado 2007] first labels strokes as wire or gate (in the digital circuit domain) using Conditional Random Fields (CRF). A neighbourhood graph is then constructed on gate strokes where each connected component of the graph represents a shape. This algorithm is tested on 51 circuit diagrams with 5 different classes (wire, AND, OR, XOR and NOT gates), achieving 77% accuracy in grouping. Although these approaches avoid the exhaustive search and do not hypothesize many shape candidates, the accuracy of these approaches is relatively low.

3.2.2 *Probabilistic Graphical Model based.*

The following work differs from the hard clustering approaches as recognition of primitives as their corresponding shapes is performed before clustering primitives into groups. These approaches use Probabilistic Graphical Models (PGMs) for the prediction of primitive labels. [Deufemia et al. 2014] propose a sketch recognition system based on Latent-Dynamic CRF (LDCRF) to model the substructure of a shape by learning relationships between shape classes. In this approach, recognition and grouping is performed in two stages; first classifying each primitive using the trained LDCRF, and then applying unsupervised geometric distance-based clustering algorithm to group the primitives belonging to the same shape. The algorithm is evaluated on the domain of electric circuit diagrams, achieving accuracy values between 81.3% and 91%. In [Wang et al. 2016] a Markov Random Field (MRF) is deployed at the primitive level to capture the local context (i.e. relationships between primitives). The stroke relationships are defined as 'same symbol' or 'others'. Having the inferred primitive labels and relationships from the model, grouping is done by merging neighbouring strokes that have the same label and the 'same symbol' relationship label. The label of the shape is defined by the label that all strokes share for the symbol strokes. In [Wang et al. 2017] object relations are added to the system using grammatical descriptions of the domain to ensure the global consistency of the recognition. The grammar also forces the explanation of all available strokes; hence, if there are leftover strokes, a backtracking process explores another solution that can explain all the strokes.

In [Delaye 2014] two tree-structured CRFs are compared for predicting stroke labels; Minimum Spanning Tree (MST-CRF) and Hierarchical clustering tree model (H-CRF). After getting the labels for each stroke through the MST-CRF, groupings of the sketch are obtained by cutting the tree where the weights of the edges are higher than a threshold, and where connected nodes have different shape interpretations. The structure used in the H-CRF is a dendrogram that is obtained by applying SLAC algorithm [Delaye and Lee 2015] on the sketch. The leaves of the dendrogram model the labels of strokes and the nodes model the clusters of strokes. Cutting the dendrogram on edges with a weight less than a threshold produce the groupings of strokes. The experimental results show that the H-CRF outperforms the MST-CRF. These approaches usually have a computationally expensive inference process.

4 OTHER METHODS

In addition to the described approaches, there are techniques that perform grouping from a top-down perspective, where the entire diagram is considered as a whole and the broken down into pieces sequentially. In [Kara and Stahovich 2007], a mark-group recognition technique is proposed. This technique relies on "marker symbols" that can be accurately and inexpensively extracted from the input data. The set of strokes coming before both ends of the marker are considered to be shapes. These techniques rely on the existence of markers which cannot be scaled to the domains that do not contain a marker. [Saund and Lank 2003] decompose a sketch into sequences of contiguous line segments corresponding to line art, and "blobs" of dense ink corresponding to text. They use Gestalt principles to group these objects into larger structures. The approach is computationally expensive for dense diagrams, and is intended to produce groupings suitable for interactive manipulation rather than object recognition [Lin 2014].

In [Chao et al. 2017] a novel gaze-aided grouping method is presented for flowchart diagrams. Gaze data is collected during the drawing of diagrams and is used to assign a heat value to every pixel in the sketching canvas. Since the heat values of arrow regions are low, as the eye spends less time there, the regions with high heat values are the potential parts to be searched for closed shapes. The summary of these methods can be seen in Table 3.

5 DISCUSSION

We believe an ideal grouping strategy should have the following characteristics:

- High accuracy
- Computationally inexpensive (for training and testing)
- Domain independent
- Supportive of a free sketch environment

In fact, these characteristics extend to all parts of a sketch recognition engine. Using our analysis of grouping approaches presented in this paper, we have summarised the success each approach has had in exhibiting these characteristics (see Table 4).

Of course without a full comparative study of each grouping technique, it is difficult to judge the relative accuracy of different approaches, particularly when different datasets are used; here we comment on them based on the reported results - which in many cases, indicates their performance in the best case scenario. [Johnson et al. 2009] discusses the accuracy rates that users find acceptable in sketching; they cite [LaLomia 1994] as reporting a 3% error is accepted as minimum level, but 1% is considered "very good" [LaLomia 1994] in handwriting recognition; while in hand gesture recognition error rates of 10% were found to be acceptable [Karam and schraefel 2006]. We regard average accuracy levels of approximately >90% to be acceptable in this paper, but acknowledge that these may still fall short of user expectations.

The characteristics for approaches that use negative examples is largely dependent on the number of primitives allowed per shape. Generally, a limit is chosen that is optimal for the size of valid shapes in the training set. Increasing this limit would increase the number of negative examples that could be included in training exponentially. Therefore, the number of shape candidates (both for training and testing) grow when a larger number of primitives per shape is allowed. A very large training set with a very large invalid shape class has the potential to affect the accuracy of the classifier. A greater number of shape candidates would result in computationally expensive training and testing phases. Generally, these approaches limit the number of primitives allowed per shape, which imposes restrictions on the sketching environment.

An alternative approach for rejecting negative examples is through novelty detection, where the goal is to identify objects that are not similar to the training set [Marsland 2003; Pimentel et al. 2014]. To the best of our knowledge, three papers have used novelty detection in the context of sketch recognition [Arandjelović and Sezgin 2011; Tirkaz et al. 2012; Yesilbek and Sezgin 2017]. As discussed in Section 2.3, in [Arandjelović and Sezgin 2011] a one-class SVM classifier is used to reject invalid shape candidates that are generated using HMM, while [Tirkaz et al. 2012] is aimed at the task of auto-completion, and [Yesilbek and Sezgin 2017] explores the use of few examples for labelling sketch datasets. Incorporating such a method with a highly accurate grouper would be a possible way forward.

Grammar and language based approaches have reported low accuracy, except [Hammond and Davis 2009] which is highly constrained and has an exponential time computational complexity. These approaches are usually computationally expensive as an exhaustive search is required to match the drawn strokes with the shape definitions of a language. They are also domain dependent as the description and constraints are specifically defined by an expert for a particular domain. In addition, the sketching environment is limited by the drawing constraints imposed by the language, where any shape that does not conform to the language cannot be recognised. Defining a language that is robust enough to handle the ambiguities of sketch recognition is a difficult task. Overall, this approach has not been as successful as others in meeting the goals of sketch recognition. However, for domain specific solutions, this may be a suitable approach.

The accuracy of simultaneous optimisation approaches varies (see Table 1), except [Sezgin and Davis 2005] where the conditions of the reported recognition rate is unclear i.e. the nature of datasets (isolated shapes or full diagram). We believe the accuracy is limited because these approaches rely on the temporal ordering of primitives. They typically have a high computation time as they need to maximize a joint probability for the entire diagram. Although [Sezgin and Davis 2008] has attempted to allow interspersed drawing, these approaches have limited support for a free-sketch environment as users need to follow a similar order of drawing to that represented in the dataset.

The sequential optimisation approaches have achieved a reasonable accuracy, and are generally domain independent. However, these approaches are usually computationally expensive as they need to optimise a cost function. In addition, given that these techniques typically limit the number of primitives allowed per shape, their support of a completely free sketch environment is questionable.

Table 3: Summary of other grouping methods - Int. = Interspersed strokes allowed, D.I. = Domain Independent

	Proximity	Int.	Dataset	Accuracy	Context	Limitations	D.I.
[Kara and Stahovich 2007]	Spatial	Yes	Modeling Matlab's Simulink package	70% in recognizing arrows	None	Limited to arrow connected diagrams	No
[Chao et al. 2017]	Spatial	Yes	Flowchart diagrams	71.15% in grouping of shapes (excluding arrows)	None	Requires peripheral devices, low accuracy, limited to closed shapes	No

Table 4: Comparison of grouping approaches. ✓= Exhibits the characteristic, ✗=Does not exhibit the characteristic, ∼=Dependant on other factors

		Accurate	Computationally Inexpensive	Domain Independent	Supports Free Sketch Environment
Simultaneous	Negative Examples	∼	∼	✓	∼
	Grammar & Language	✗	✗	✗	∼
	Optimisation	✗	✗	✓	✗
Sequential	Sequential Optimisation	✓	✗	✓	∼
	Clustering: Hard Clustering	✗	✓	✓	✓
	Clustering: PGM-based	✗	✗	✓	✓

The sequential approaches that use hard clustering for the task of grouping have reported relatively low accuracy rates. It is possible that the accuracy could be improved by designing more discriminating feature sets. The computation time of these approaches is polynomial, although during the classification process, the calculation of pairwise features for each pair of primitives can take some time in practice. However, one advantage is that they avoid generating too many shape candidates in the grouping phase, which maintains low computation time for the recognition step. Given that these methods are trainable, they can be applied to multiple domains. They also support a free sketch environment.

The sequential PGM-based approaches are also domain independent and allow for a free sketching environment, like the hard-clustering methods. However, they have a high computation time as the maximization of joint probability (in the inference process) is computationally expensive. The reported accuracy of these approaches are relatively low.

We believe that the issues considered above for grammar and languages, simultaneous optimisation and PGM-based clustering approaches are difficult to resolve. Hard clustering has obvious computational benefits and we hypothesize that with more discriminatory features it may improve performance.

We believe there are promising ways forward, that have the potential to improve negative example and sequential optimisation techniques. Both of these methods suffer from over-segmentation, which is when a large number of shape candidates are generated in the search process; for these methods an exponential number of shape candidates are produced. The effect of over-segmentation is exponential computation time for the number of shape candidates. Typically, a limit is placed on the number of primitives allowed per shape candidate to limit over-segmentation. However, this constrains the drawing environment, as any shapes drawn with more than this number of primitives cannot be grouped or recognised. In order to preserve accuracy, the number of primitives per shape allowed should be chosen carefully.

A possible solution to these issues is to design a hybrid approach where a hard-clustering algorithm is used to group shapes, as these methods are computationally inexpensive and support free-sketching. However, these methods have a lower accuracy in grouping. There are two ways of hard clustering that have been used previously: SLAC and two-level classification. Two-level classification makes a hard decision on grouping that cannot be altered, whereas SLAC can be tuned to produce a more acceptable number of shape candidates. Using a technique like SLAC with sequential optimisation would solve the problem of supporting a free-sketch environment, and resolve part of the issue of computation time. The remaining computationally expensive part of sequential optimisation techniques, is the optimisation process itself, which cannot be solved in this manner. On the other hand, combining SLAC or similar, with negative example approaches solves the free-sketch issue, and could produce a more computationally inexpensive approach given the reduced number of shape candidates that would be produced.

SLAC has successfully been used in this manner in [Bresler et al. 2015b] for negative examples (see Section 2.1). However, in this paper it is pointed out that in their experiments they never achieved 100% accuracy in grouping. Hence, there is a trade-off between the accuracy of these approaches and the computation time of naïve over-segmentation (exploring all possible shape candidates). In cases where computational time is more important than accuracy this approach could be of benefit. One issue remains for negative examples, which is how to deal with invalid shape candidates. As previously mentioned, novelty detection could be investigated as a way of dealing with this remaining issue.

We believe in the case of over-segmentation, a more intelligent grouping algorithm that is capable of pruning the search space is needed. The addition of a backtracking mechanism, similar to [Wang et al. 2017], to explore other branches of the search space could also assist here in capturing valid shape candidates that are missed initially.

Another possible improvement would be to use context in the grouping/recognition process. This can be achieved by considering relationships between shapes similar to [Feng et al. 2009] or allowing neighbouring interpretations to influence each other similar to [Ouyang and Davis 2011]. We believe that this information adds value to the grouping/recognition process. A formal study to analyse the impact of contextual information would be beneficial. It would be ideal to have this information inferred from the training set to support domain independent systems.

6 CONCLUSION

In this paper we have analysed the approaches to grouping of primitives in sketched diagrams. We categorise existing work into those that perform grouping and shape recognition simultaneously and those that are sequential in their approach. Simultaneous methods are further split into those that use negative examples, grammar and languages, or optimisation techniques. Sequential methods include optimisation, and clustering approaches. We have examined the role of grouping in sketch recognition, by analysing the contribution of each approach to the goals (or ideal characteristics) of sketch recognition. In particular, the goals of designing accurate, computationally inexpensive, domain independent algorithms, that support free-sketch environments have been considered. We have found that there are promising ways forward to solve the remaining problems of high computation time, accuracy and supporting free-sketching. We propose further investigation of approaches using negative examples, sequential optimisation and hard clustering. Improvements that can be gained in the important step of grouping will result in better recognition rates for sketched diagrams.

REFERENCES

Harold Mouchre Christian Viard-Gaudin Ahmad-Montaser Awal, Guihuan Feng. 2011. First experiments on a new online handwritten flowchart database. (2011). DOI: http://dx.doi.org/10.1117/12.876624

Christine Alvarado. 2007. Sketch recognition for digital circuit design in the classroom. In *2007 Invited Workshop on Pen-Centric Computing Research*. Citeseer.

Christine Alvarado and Randall Davis. 2004. SketchREAD: a multi-domain sketch recognition engine. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 23–32.

Christine Alvarado and Randall Davis. 2006. Dynamically constructed bayes nets for multi-domain sketch understanding. In *ACM SIGGRAPH 2006 Courses*. ACM, 32.

Relja Arandjelović and Tevfik Metin Sezgin. 2011. Sketch recognition by fusion of temporal and image-based features. *Pattern Recognition* 44, 6 (2011), 1225–1234.

Martin Bresler. 2016. Online recognition of sketched arrow-connected diagrams. *International Journal on Document Analysis and Recognition (IJDAR)* 19, 3 (2016), 253fi?!267.

Martin Bresler, Daniel Prùša, and Václav Hlavác. 2013a. Modeling Flowchart Structure Recognition As a Max-Sum Problem. In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition (ICDAR '13)*. IEEE Computer Society, Washington, DC, USA, 1215–1219. DOI: http://dx.doi.org/10.1109/ICDAR.2013.246

Martin Bresler, Daniel Průša, and Václav Hlaváč. 2013b. Simultaneous Segmentation and Recognition of Graphical Symbols using a Composite Descriptor. In *CVWW 2013: Proceedings of the 18th Computer Vision Winter Workshop*, Walter G. Kropatsch, Geetha Ramachandran, and Fuensanta Torres (Eds.). Vienna University of Technology, Karlsplatz 13, Vienna, Austria, 16–23.

Martin Bresler, Daniel Prusa, and Václav Hlavác. 2015a. Detection of arrows in on-line sketched diagrams using relative stroke positioning. In *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 610–617.

Martin Bresler, Daniel Pruša, and Václav Hlavác. 2015b. Using agglomerative clustering of strokes to perform symbols over-segmentation within a diagram recognition system *(CVWW)*. 67–74.

Martin Bresler, Truyen Van Phan, Daniel Prusa, Masaki Nakagawa, and Vclav Hlavc. 2014. Recognition system for on-line sketched diagrams. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 563–568.

Beibei Chao, Xiaoyan Zhao, Dapeng Shi, Guihuan Feng, and Bin Luo. 2017. Eyes Understand the Sketch!: Gaze-Aided Stroke Grouping of Hand-Drawn Flowcharts. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces (IUI '17)*. ACM, New York, NY, USA, 79–83.

Gennaro Costagliola, Vincenzo Deufemia, and Michele Risi. 2005. Sketch grammars: A formalism for describing and recognizing diagrammatic sketch languages. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 1226–1230.

Gennaro Costagliola, Mattia De Rosa, and Vittorio Fuccella. 2014. Local context-based recognition of sketched diagrams. *Journal of Visual Languages & Computing* 25, 6 (2014), 955 – 962. DOI: http://dx.doi.org/https://doi.org/10.1016/j.jvlc.2014.10.021 Distributed Multimedia Systems DMS2014 Part I.

Gennaro Costagliola, Mattia De Rosa, and Vittorio Fuccella. 2015. Extending local context-based specifications of visual languages. *Journal of Visual Languages & Computing* 31 (2015), 184 – 195. DOI: http://dx.doi.org/https://doi.org/10.1016/j.jvlc.2015.10.013 Special Issue on DMS2015.

Adrien Delaye. 2014. Structured prediction models for online sketch recognition. *Interpretation* 1, 3 (2014), 4–16.

Adrien Delaye and Kibok Lee. 2015. A flexible framework for online document segmentation by pairwise stroke distance learning. *Pattern Recognition* 48, 4 (2015), 1193–1206.

Vincenzo Deufemia, Michele Risi, and Genoveffa Tortora. 2014. Sketched Symbol Recognition Using Latent-Dynamic Conditional Random Fields and Distance-based Clustering. *Pattern Recogn.* 47, 3 (March 2014), 1159–1171. DOI: http://dx.doi.org/10.1016/j.patcog.2013.09.016

Guihuan Feng, Christian Viard-Gaudin, and Zhengxing Sun. 2009. On-line Hand-drawn Electric Circuit Diagram Recognition Using 2D Dynamic Programming. *Pattern Recogn.* 42, 12 (Dec. 2009), 3215–3223. DOI: http://dx.doi.org/10.1016/j.patcog.2009.01.031

Leslie Gennari, Levent Burak Kara, Thomas F Stahovich, and Kenji Shimada. 2005. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers & Graphics* 29, 4 (2005), 547–562.

Tracy Hammond and Brandon Paulson. 2011. Recognizing sketched multistroke primitives. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 1 (2011), 1–34.

Tracy A Hammond and Randall Davis. 2009. Recognizing interspersed sketches quickly. In *Proceedings of Graphics Interface*. Canadian Information Processing Society, 157–166.

James Herold and Thomas F Stahovich. 2012. The 1 Recognizer: a fast, accurate, and easy-to-implement handwritten gesture recognition technique. In *Proceedings - Sketch-Based Interfaces and Modeling, SBIM*. Eurographics Association, 39–46.

Heloise Hse and A. Richard Newton. 2004. Sketched Symbol Recognition Using Zernike Moments. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01 (ICPR '04)*. IEEE Computer Society, Washington, DC, USA, 367–370. DOI: http://dx.doi.org/10.1109/ICPR.2004.838

Heloise Hwawen Hse and A Richard Newton. 2005. Recognition and beautification of multi-stroke symbols in digital ink. *Computers & Graphics* 29, 4 (2005), 533–546.

Gabe Johnson, Mark D. Gross, Jason Hong, and Ellen Yi-Luen Do. 2009. Computational Support for Sketching in Design: A Review. *Found. Trends Hum.-Comput. Interact.* 2, 1 (Jan. 2009), 1–93. DOI: http://dx.doi.org/10.1561/1100000013

David Johnston and Christine Alvarado. 2013. Sketch Recognition of Digital Logical Circuits. (2013).

F. Julca-Aguilar, H. Mouchère, C. Viard-Gaudin, and N. S. T. Hirata. 2017. A General Framework for the Recognition of Online Handwritten Graphics. *ArXiv e-prints* (Sept. 2017).

Levent Burak Kara and Thomas F Stahovich. 2007. Hierarchical parsing and recognition of hand-sketched diagrams. In *ACM SIGGRAPH - International Conference on Computer Graphics and Interactive Techniques*. ACM, 17.

Maria Karam and m. c. schraefel. 2006. Investigating User Tolerance for Errors in Vision-enabled Gesture-based Interactions. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '06)*. ACM, New York, NY, USA, 225–232. DOI: http://dx.doi.org/10.1145/1133265.1133309

Mary LaLomia. 1994. User Acceptance of Handwritten Recognition Accuracy. In *Conference Companion on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 107–108. DOI: http://dx.doi.org/10.1145/259963.260086

C Lee, Josiah Jordan, Thomas F Stahovich, and James Herold. 2012. Newtons Pen II: an intelligent, sketch-based tutoring system and its sketch processing techniques. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*. Eurographics Association, 57–65.

Han-Lung Lin. 2014. *Estimating Student Competence in Engineering Statics From a Lexical Analysis of Handwritten Equations.* Thesis.

Stephen Marsland. 2003. Novelty detection in learning systems. *Neural computing surveys* 3, 2 (2003), 157–195.

Tom Ouyang and Randall Davis. 2009. Learning from neighboring strokes: Combining appearance and context for multi-domain sketch recognition. In *Advances in Neural Information Processing Systems 22*. 1401–1409.

Tom Y Ouyang and Randall Davis. 2007. Recognition of Hand Drawn Chemical Diagrams. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 1 (AAAI'07)*. AAAI Press, 846–851.

Tom Y Ouyang and Randall Davis. 2011. ChemInk: a natural real-time recognition system for chemical drawings. In *In International Conference on Intelligent User Interfaces (IUI fi11*. ACM, 267–276.

Brandon Paulson and Tracy Hammond. 2008. PaleoSketch: Accurate Primitive Sketch Recognition and Beautification. In *Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '08)*. ACM, 1–10. DOI: http://dx.doi.org/10.1145/1378773.1378775

Eric Jeffrey Peterson, Thomas F Stahovich, Eric Doi, and Christine Alvarado. 2010. Grouping Strokes into Shapes in Hand-Drawn Diagrams. In *Proceedings of the National Conference on Artificial Intelligence*. 974–979.

Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99 (2014), 215 – 249. DOI: http://dx.doi.org/10.1016/j.sigpro.2013.12.026

Beryl Plimmer, Rachel Blagojevic, Samuel Hsiao-Heng Chang, Paul Schmieder, and Jacky Shunjie Zhen. 2012. Rata: codeless generation of gesture recognizers. In *Proceedings of the 26th Annual BCS Interaction Specialist Group Conference on People and Computers*. British Computer Society, 137–146.

J Reaver, Thomas F Stahovich, and James Herold. 2011. How to make a quick $: Using hierarchical clustering to improve the efficiency of the dollar recognizer. In *Proceedings - SBIM: ACM SIGGRAPH / Eurographics Symposium on Sketch-Based Interfaces and Modeling*. ACM, 103–108.

Dean Rubine. 1991. Specifying Gestures by Example. (1991), 329–337. DOI: http://dx.doi.org/10.1145/122718.122753

Eric Saund and Edward Lank. 2003. Stylus input and editing without prior selection of mode. In *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology*. ACM, 213–216.

Tevfik Metin Sezgin and Randall Davis. 2005. HMM-based efficient sketch recognition. In *Proceedings of the 10th international conference on Intelligent user interfaces*. ACM, 281–283.

Tevfik Metin Sezgin and Randall Davis. 2007a. Sketch interpretation using multiscale models of temporal patterns. *IEEE Computer Graphics and Applications* 27, 1 (2007), 28–37.

Tevfik Metin Sezgin and Randall Davis. 2007b. Temporal Sketch Recognition in Interspersed Drawings. In *Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling (SBIM '07)*. ACM, New York, NY, USA, 15–22. DOI: http://dx.doi.org/10.1145/1384429.1384436

Tevfik Metin Sezgin and Randall Davis. 2008. Sketch recognition in interspersed drawings using time-based graphical models. *Computers & Graphics* 32, 5 (2008), 500–510.

Michael Shilman and Paul Viola. 2004. Spatial recognition and grouping of text and graphics. In *Proceedings of the First Eurographics conference on Sketch-Based Interfaces and Modeling*. Eurographics Association, 91–95.

Michael Shilman, Paul Viola, and Kumar Chellapilla. 2004. Recognition and grouping of handwritten text in diagrams and equations. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*. IEEE, 569–574.

Thomas F Stahovich, Eric J Peterson, and Hanlung Lin. 2014. An efficient, classification-based approach for grouping pen strokes into objects. *Computers & Graphics* 42 (2014), 14–30.

Philip C. Stevens, Rachel Blagojevic, and Beryl Plimmer. 2013. Supervised Machine Learning for Grouping Sketch Diagram Strokes. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (SBIM '13)*. ACM, New York, NY, USA, 43–50. DOI: http://dx.doi.org/10.1145/2487381.2487383

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Caglar Tirkaz, Berrin Yanikoglu, and T. Metin Sezgin. 2012. Sketched symbol recognition with auto-completion. *Pattern Recognition* 45, 11 (2012), 3926 – 3937.

Chengcheng Wang, Harold Mouchère, Aurélie Lemaitre, and Christian Viard-Gaudin. 2017. Online flowchart understanding by combining max-margin Markov random field with grammatical analysis. *International Journal on Document Analysis and Recognition (IJDAR)* 20, 2 (01 Jun 2017), 123–136. DOI: http://dx.doi.org/10.1007/s10032-017-0284-8

C. Wang, H. Mouchre, C. Viard-Gaudin, and L. Jin. 2016. Combined Segmentation and Recognition of Online Handwritten Diagrams with High Order Markov Random Field. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 252–257. DOI: http://dx.doi.org/10.1109/ICFHR.2016.0056

Jacob O Wobbrock, Andrew D Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology*. ACM, 159–168.

Kemal Tugrul Yesilbek and T. Metin Sezgin. 2017. Sketch recognition with few examples. *Computers & Graphics* 69 (2017), 80 – 91. DOI: http://dx.doi.org/https://doi.org/10.1016/j.cag.2017.08.016