Using Clustering to Assist Understanding of Digital Ink in Low Attention Environments

Craig Prince
University of Washington
cmprince@cs.washington.edu

ABSTRACT

Understanding large amounts of multimedia information is a challenge in many domains – as technology makes its way into the classroom, the classroom becomes one such domain. An instructor has little attention to focus on the technology and is instead focused on teaching. We examine several techniques for clustering digital ink diagrams drawn by students during in-class activities. These diagrams are submitted electronically to the instructor in real-time. The goal of clustering is to allow the instructor to gain an overview of the responses submitted, to quickly assess the level of understanding of the students, and to select "interesting" responses to display and discuss further. We have found that an instructor has difficult achieving these goals even in small classes of size 15 to 20. As the class size increases this task becomes impossible. We find that for some exercises our clustering works surprisingly well and should help to reduce the cognitive load on an instructor.

1. INTRODUCTION

Computers today are dealing with richer and broader sets of data than ever before. The most significant reason for this is the increase in raw computing power and storage resulting from cheaper, more powerful hardware. This increase not only enables the analysis of more information, but also allows us to capture and digitize a richer and more complete set of media. Image, audio, and video analysis is becoming commonplace on personal computers and even the real time capture of handwriting has become possible with the development of tablet computing technologies. With all these new streams of media comes the new problem of how we can effectively understand, assimilate, and use all this information.

While it is very easy for a human to understand and process multimedia data, as the volume of data increases this task becomes increasingly difficult. Furthermore, in many of the domains where multimedia is most useful the user is unable to focus all his/her attention on the technology. In fact, in most cases the technology is just a tool in performing some higher-level task. Not only is the focus not on the technology, but also time is a critical factor in these domains and analysis of the data must be done in near real-time. It is under these low-attention circumstances that being able to automatically understand multimedia content and reduce the cognitive load on the user becomes most important.

As technology and classroom networks become an integral part of education, the classroom becomes one domain that exhibits all of the characteristics described above and is the focus of this research. Specifically, we examine the scenario where an instructor must assimilate digital ink data from numerous classroom participants in real-time. During a class session the instructor may give one or more exercises to class participants to complete. Upon completion, the participants submit their responses to the instructor, who can then use the responses to achieve numerous goals and aid his/her instruction. This is a difficult domain because within the classroom the technology is not the primary focus of an instructor but is simply a tool used to aid instruction. As a result, an instructor can not be expected to perform deep analysis of any multimedia as his/her attention will instead be focused on the students. Furthermore, the classroom domain is one where real-time interaction is imperative. Class time is limited and any multi-media data must be used within a short time span – there cannot be long pauses in the instruction as the instructor attempts to understand any digital ink data he or she is receiving.

1.1 Ink Clustering Problem

The problem faced by an instructor in the classroom domain is one of cognitive load and the goal of this work is to reduce this load. Because data is being generated by classroom participants, there is also an issue of scaling. That is, the problems only become worse as the number of students in a classroom increases. There are several techniques that can be used in order to reduce the cognitive load associated with understanding any multimedia data. First, novel visualization techniques can be leveraged in order to allow a user to more efficiently see a larger amount of data. Second, data summarization can be used to remove non-relevant information, leaving only important aspects of the data. Lastly, various structuring techniques can be used to help organize the data and make it easier to understand and navigate. These three axes are not mutually exclusive and can all be used in conjunction to improve the understanding of rich sets of media. Our approach to solving this problem focuses on automatically clustering the digital ink in order to assist cognition in this low-attention environment.

In this work we have chosen to solve our problem by imposing some structuring on the data to allow a viewer to quickly and efficiently make sense of large data sets. The most obvious solution would be to simply remove duplicate answers or answers with the same semantic meaning. This is basically the approach we take – that is, we seek to cluster semantically "similar" responses together in order to reduce the cognitive load on the instructor by reducing the number of responses that need to be assimilated.

For this approach to work, we must make several assumptions about out data. First, we assume that the data itself has some interesting underlying structure to cluster. We must also assume that we can filter out other non-relevant structure that exists in the data and instead focus only on the useful structure that we are concerned about. Lastly, we must believe that the types of exercises that instructors conduct and the goals that instructors have for their exercises lend themselves to clustering. Through observation of our data we have found that digital ink often falls

into clusters that have semantic similarities. Therefore, it is plausible to believe that given the correct similarity metrics it should be possible and useful to cluster our data.

1.2 Contributions

It is important to clarify that we do not view the specific clustering algorithms and metrics presented in this work as being the major contribution of this work. On the contrary, the algorithms used are simple and well-known. We feel that a lack of complicated application-specific algorithms helps our work to maintain some generality and allows us to focus on the more important contributions.

One important contribution of this work is that we are focusing on a domain (i.e. the classroom) where the end user has little attention to focus on the technology. Within the classroom environment the instructor is focused primarily on teaching the students. In addition, the instructor is usually nervous or excited. These factors make is difficult to use technology in the classroom. These issues are not just relevant to the classroom domain either, but are also generally applicable to many other domains in which a user's attention is divided. Consider for example the surgical domain. The surgeon's attention is focused on the patient; however, there are numerous streams of multimedia including output from various monitors, dialog with nurses and other specialists, patient history information, X-ray/MRI data, etc. Having methods to allow quick assimilation of this data is vital and our work, while focused on the classroom domain, is a step toward solving the issue in general.

Another important contribution of this work is that we focus on helping to use multimedia to improve the educational experience. There is a tremendous opportunity to leverage technology in the classroom to improve the overall educational experience [6][7][20]. Previous work has shown that technology can promote classroom interaction by enabling active learning [16][21][23] and peer instruction [10], two techniques which have a positive impact on education. In general the types of classroom networks enabled by technology help to keep students engaged and have been very successful [2]. Our work also has a direct relation to previous work on in-class assessment techniques [4][12][18]. However, the previous classroom response and assessment technologies have not been based on digital ink and thus do not allow for the same richness of response.

Finally, our work is specifically focused on the digital ink medium. This is a relatively uncommon form of media, but is extremely expressive as a digital medium. Handwriting/Drawing and thus digital ink is very easy for humans to understand and is a natural input modality. However, this modality is inherently "analog" which makes it difficult for computers to work with. Digital ink is very different than other forms of media like text and images. While much previous work has been done in clustering these other forms of media [9][22][24], we should not expect all the previous results and techniques to be directly applicable.

2. Classroom Technology

There are many systems that have been designed to assist in improving education [2]. Some of these systems are designed to simply allow the capture and replay of educational material [17]. While this use of technology provides some benefit – it is interactive systems that have the most potential. There are several axes of interaction as well – for example, the authors of [13] provide a system for student collaboration. This promotes peer instruction, but because the interaction is many-to-many there is less of a burden on each individual to assimilate large amounts of multimedia. The second axis of interaction is the instructor-class interaction. This interaction is most effective when the instructor is able to directly engage the students. Several systems seek to provide technological means to lower this barrier to in-class communication [5][11][19].

We have developed Classroom Presenter, an application for the Tablet PC, designed for use in the classroom [1]. The basic functionality of Classroom Presenter is to allow for the display and annotation of PowerPoint-style slide decks. In addition, our system has the ability for student devices to be synchronized with the instructor's presentation and also allows for note-taking on the student device. One of the most compelling features of Classroom Presenter is its ability to allow the instructor to give exercises/activities to the students in a classroom in real-time. That is, the instructor can give an in-class exercise/activity to the students and the students can then complete it and submit their responses electronically to the instructor. The instructor can then display these responses on a public display in order to comment on and discuss the exercise/activity.

There are three different interfaces for classroom presenter depending on the user of the system. Figure 1 shows the instructors interface, on the left is a filmstrip view showing all the slides in the current deck. The current slide is shown in the center and the system allows the instructor to write directly on this slide. A public display will always display the current slide. In this example, we see a blank slide containing an exercise for the students to complete. The students are to draw out the appropriate Huffman tree as given in the exercise. Figure 2 shows the student's view of the exercise on his/her device. Because the student is using a Tablet PC he/she is able to write directly on the application to solve the exercise. The figure shows an actual completed student response. After the student is finished he or she can submit the response back to the instructor. Figure 3, shows how these responses (also referred to as student submissions) are displayed to the instructor. All the submissions are placed into their own deck and the instructor can select which of the various responses to display on the public display allowing the instructor to discuss and annotate various student submissions.

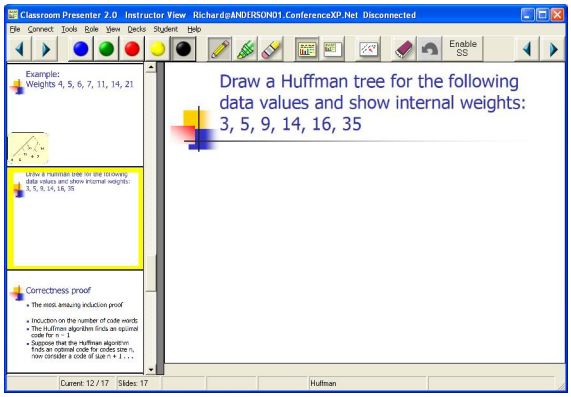

Figure 1 Instructor interface showing a slide for an exercise.

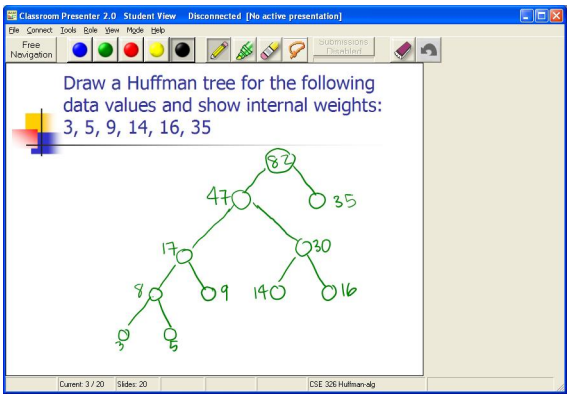

Figure 2 Student interface after the student has drawn a tree.

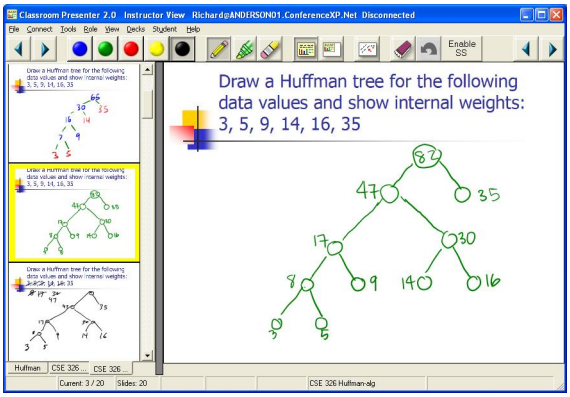

Figure 3 Instructor view after slides have been received from the students. The student submissions are placed in a film strip so the instructor can preview and select them to be shown on the public display. Submissions from three different students appear in the film strip.

2.1 Classroom Experience

Classroom Presenter has been in use on an experimental basis since 2004. We have had 4 instructors using the system in over 6 courses with a total of 28 sessions. Up to this point we have had 87 in class exercises conducted. In addition, we have collected data from several demos. Through this experience we have been able to collect a large corpus of real data. We have over 1100 individual student submissions. We stress that all of the data used in this research is from real classroom experience and is not fabricated. These are actual submissions from university level students. The setup for our system is usually in a classroom with a single instructor computer plus numerous student machines. The computers are connected via either a wireless or wired network.

There has been a large variety of activities conducted by instructors using our system. For the most part they fall into three categories: textual responses, numerical responses, and diagrammatical responses. In this work we focus primarily on the diagrammatical exercises. We believe that textual and numerical will require different techniques than presented here. We also focus on diagrammatical responses because it is these exercises where the digital ink is most beneficial for.

To make the type of data we're dealing with more concrete we present several examples of real exercises. Figure 4 illustrates an exercise where the student is asked to draw out the appropriate Huffman tree for the given set of numbers. This example shows the benefit of using digital ink because the results not only include the answer, but also a derivation of the answer. This is a strength of the student submission paradigm because it allows for more expressive answers, but on the other hand it also makes automatic recognition more difficult. Figure 5 is a simpler exercise that involves drawing graphs. Because of the regularity of the responses this should be a good candidate for automatic methods. Finally, Figure 6 is an exercise that asked the students to brainstorm a user interface design. This exercises allowed the students to be more expressive and thus the responses are more visually diverse.

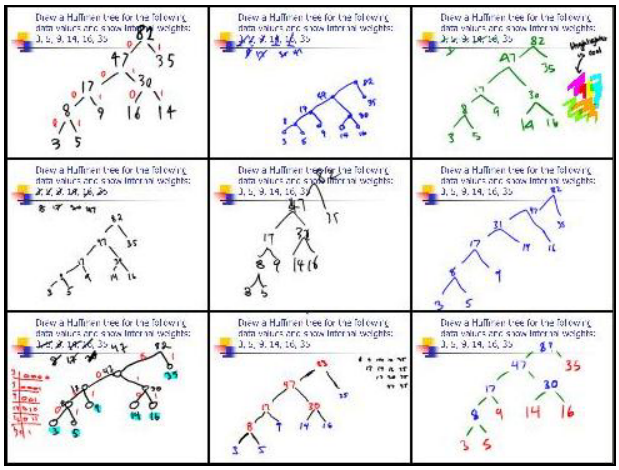

Figure 4 Student submission exercises showing trees constructed by students.

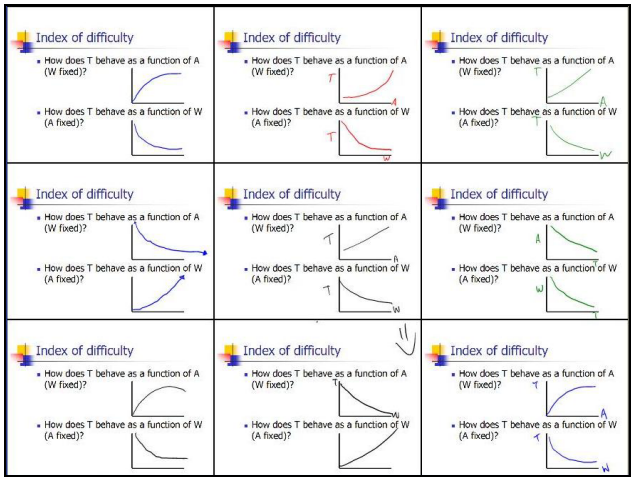

Figure 5 Exercise involving drawing of graphs on the provided graph axes.

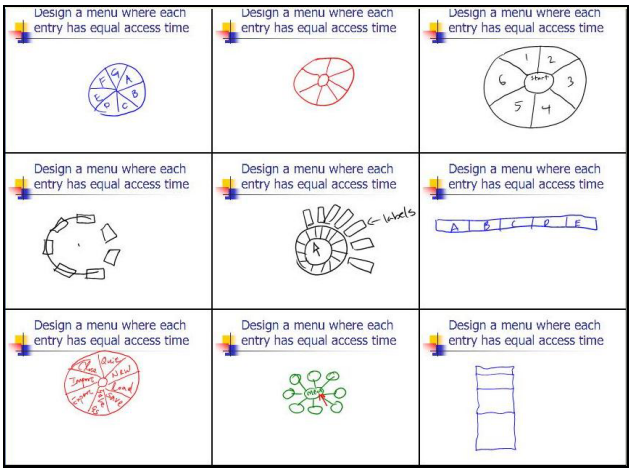

Figure 6 Diagrammatic exercise where students were asked to draw an example menu with a specified property.

Given that the instructors receive a vast array of responses there are several goals in mind for analyzing student responses. For many exercises the instructor simply wants to get a distribution of how many students got the exercise correct and how many got it incorrect. As the size of the class increases it would be sufficient to simply perform random sampling in order to get this distribution. This is not the only goal however, so random sampling is not a viable solution. Another goal of instructors is to find a wide variety of answers – this includes rare, but interesting responses. Random sampling can miss these rare responses, yet these same responses often lead to the most in-class discussion.

Even in classes with as few at 15 student devices and on problems with relatively simple responses, the instructors find it difficult to achieve the goals listed above. This has a direct impact on the scalability of our system and as class sizes increase it will become increasingly difficult to deal with the digital ink responses in class. The ultimate goal of our system would be to scale to large lecture style classrooms with over 100 students. Our work looks to enable this scenario without diminishing the richness that digital ink provides as an expressive medium.

3. Automatic Clustering

The purpose of using clustering to automatically group student submissions is to reduce the cognitive load placed on an instructor during class. By grouping student submissions that are similar, in principle an instructor can simply look at a single instance from each cluster to get an overview of all the different submissions that have been made. Furthermore, by looking at and comparing the size of the clusters created it is easy to get a general sense of how common a particular solution is. As a result, clustering submissions is the perfect solution for completing the two tasks described in section 2.1 above.

There has been a resurgence of interest in the problem of ink understanding, with various projects developing recognizers for different classes of diagrams [3][14]. Most of the work in ink understanding is very domain specific and involves determining the abstract representation of the digital ink. Our case is somewhat easier since we do not need to understand the digital ink, but simply need to look for similarity amongst the submissions. It could be argued that understanding what the digital ink represents is vital to effective clustering since two drawings can be different yet still represent the same solution. However, in the teaching domain it is often helpful to show the same solution presented in two different ways and so it may be useful for an instructor to be given both solutions.

Our approach to clustering is not novel and much previous work has been done in this area [15]. What is new is that we are dealing with the digital ink media in the classroom setting which has not been done before.

3.1 Clustering in the In-Class Domain

While the overall goal is to group similar student responses, there are several factors that are of particular interest in the classroom domain. One challenge for in-class exercises is that there is often no predefined correct solution. Because we want to allow the instructor to cluster results from any type of activity we do not want to train toward a specific type of exercise. As a result, we cannot leverage training examples of correct and incorrect exercises for our clustering. One possible use for training would be to help extract high-level features from the digital ink to be used in clustering; however, we do not take this approach in this work.

A second challenge in this domain is that we are dealing with the digital ink medium. This is a challenge because digital ink is inherently sloppy and it is challenging to come up with good heuristics for recognition. Also the data is vector data, which is different than image data.

A third challenge in clustering is that all the ink on the slide is not always pertinent to the solution. Some of the digital ink is "doodling" (see Figure 7a) and some of the ink is from preliminary work toward solving the exercise (see Figure 7b). This ink is not part of the solution and it is important not to consider it when clustering similar solutions. Non-pertinent ink (which is more prevalent in this domain than others) makes being able to recognize and filter out non-pertinent ink an important part of automatic clustering.

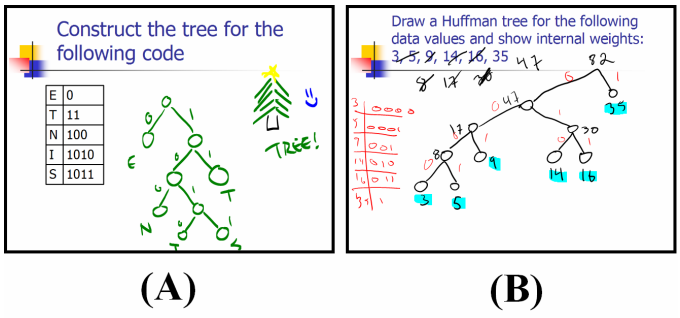

Figure 7 Examples of non-relevant digital ink on submissions.

While we have outlined several dimensions which make the task of automatic clustering difficult, the domain also has several properties that make our task easier. First, the clustering does not have to be perfect. Our goals are simply to allow the instructor to select "interesting" examples in class and to get a general sense of how the class is doing; therefore – it does not matter if a couple examples are incorrectly clustered.

Another advantageous property of this domain is that we only need to cluster similar solutions, not determine if that solution is correct or how it is incorrect. As a result, the most important aspect of the clustering is to have an accurate similarity metric for solutions. The instructor will still need to interpret the solutions; however, we hope to greatly reduce the number of slides that need to be considered.

3.2 Clustering Using an Ink-Based Approach

Our initial approach was to operate directly at the level of individual ink strokes. In this approach we extracted a set of features directly from the strokes themselves and then performed clustering on these features. We calculated these features by first identifying specific strokes, then calculating statistics on these strokes. For example, we might first find the longest stroke and then calculate the average slope. One could then use this statistic to cluster different graphs based on their slope.

Some of the metrics we used to identify strokes were: find the longest stroke, find only strokes that were straight, find strokes that are longer than a certain threshold, etc. Once we had isolated a set of strokes we then could calculate statistics such as: the average curvature, the average length, the number of strokes, the average slope, etc.

Our experience using this technique for clustering was quite poor. For example to cluster various tree structures we tried to first extract all the long straight strokes and then count them. The intuition for this approach was that the straight strokes would correspond to the various branches of the tree and this would be enough to differentiate various tree shapes. Unfortunately, this rather simple approach did not work well for a number of reasons. First, using features at the stroke level is too local. In the previous example, this approach would work fine for trees, but would not generalize to other exercises. Also, this approach is too simple to capture many interesting differences in trees.

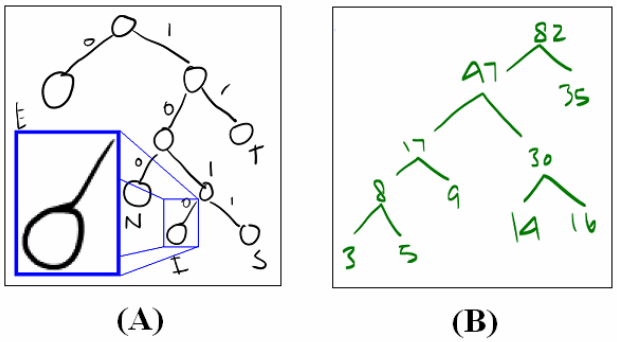

Figure 8 Two cases where more than one object was drawn with a single stroke.

Another reason that our ink-based approach didn't work is that we were naïve in extracting the features and didn't take into account the wide variety of ways that people draw. Consider Figure 8 above. Here we can see that often-times two logical objects in a diagram are actual made up of the same ink stroke. This means that an additional preprocessing of strokes is necessary to account for these behaviors. Additionally, we can see in Figure 9b and Figure 9c below two cases where more than one stroke is used to draw the same logical line. Disambiguating this from two separate lines is very difficult. Similarly in Figure 9a we see that sometimes people overwrite the same line more than once. This is another case where the strokes would need to be combined into a single logical object.

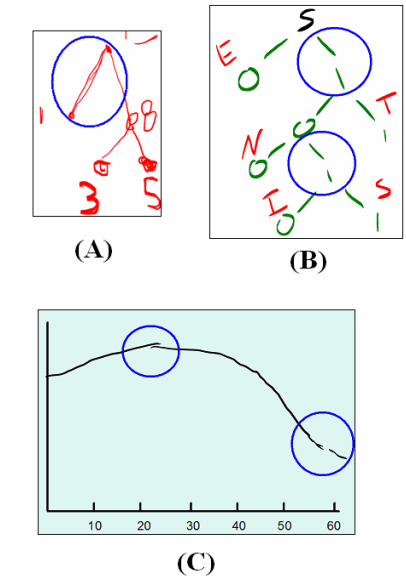

Figure 9 Cases where a single line is made of more than one stroke.

We stress that our inability to get this approach to work is not a flaw in the technique; however, we found that the simple approaches we tried are not viable and more sophisticated techniques for extracting useful features from the ink are needed.

3.3 Clustering Using an Image-Based Approach

Because of our limited success at using stroke-based features for clustering we then attempted to leverage a more standard image-based clustering technique. Treating the digital ink as an image allows us to consider the drawing as one cohesive unit as opposed to a set of individual strokes. Because we are dealing with stroke data, the first necessary step is to convert the ink into a binary image. We do this by overlaying a pixel grid on the ink and assigning pixels based on whether or not a stroke goes through that pixel. This gives us a result similar to Figure 10b below. Once we have a binary image we can then use the Chamfer distance as a simple image comparison metric to gives us distances between images [7].

Our distance metric for images calculates a per-pixel distance between each pixel in an image, A, and the closest pixel in another image, B. Note that this metric is not symmetric – that is, the distance from A to B is not the same as the distance from B to A. Figure 10c below shows an example of a distance image that we calculate to quickly determine the distance between two images. For each pixel in the distance image the number corresponds to the distance to the nearest pixel in the binary image. We use this metric because it 1) is easy to calculate and 2) accentuates large differences between images. This is preferred

since smaller differences are likely to be the result of noise, sloppiness, and/or translations.

Given this distance metric we can now cluster the images (and their corresponding ink) into groups. We use the basic K-means clustering algorithm to perform the clustering [15].

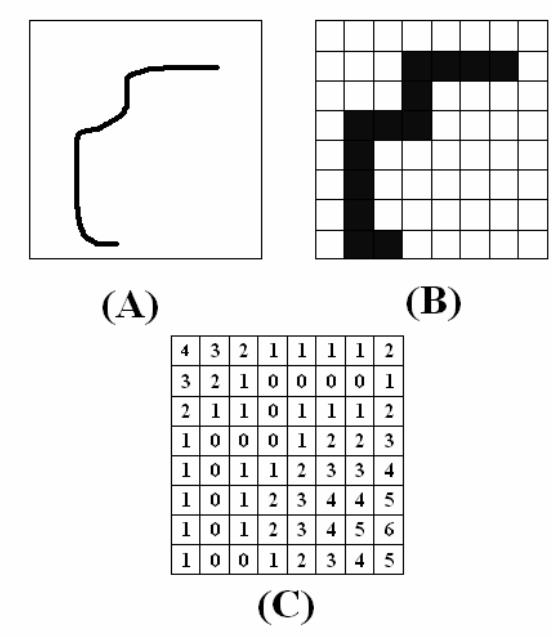

Figure 10 Calculating our distance metric: a) is the ink, b) is the binary image, c) is the distance image.

4. Clustering Results

In order to test the performance of our image-based approach to clustering (see section 3.3) we used four datasets from real in-class exercises. It was imperative to test on real datasets since our end purpose is to deploy our system in a classroom environment. We chose these examples because they are 1) representative of the types of exercises that we have encountered in classes and 2) exercises involving drawing diagrams. We began by removing any non-pertinent ink from the slides in each dataset. While artificial, we wanted to test how well in-general clustering works to group similar solutions to an exercise. Later in this section we will examine what effect this cleaning had on our results.

Table 1 Summary of Datasets.

Dataset Name	# of Slides
Single Graph	65
Dual Graphs	15
UI Layout	14
Tree	20

Table 1 above gives a summary the datasets used for testing and includes the number of slides in each. The Single Graph exercise involved having the students draw a graph of student attention versus time.

Figure 11 below gives some general examples of the types of graphs in this data set. The Dual Graphs exercise is similar except we chose it because the instructors were asked to draw two graphs on the same slide. We wanted to see how our algorithm would perform in this slightly more difficult scenario – again examples of this data can be seen in Figure 5 above. The third data set, UI Layout, involved asking the students to design a user-interface menu; Figure 6 above gives some examples from the dataset. Similarly Figure 4 above shows examples from our Tree dataset involving tree construction.

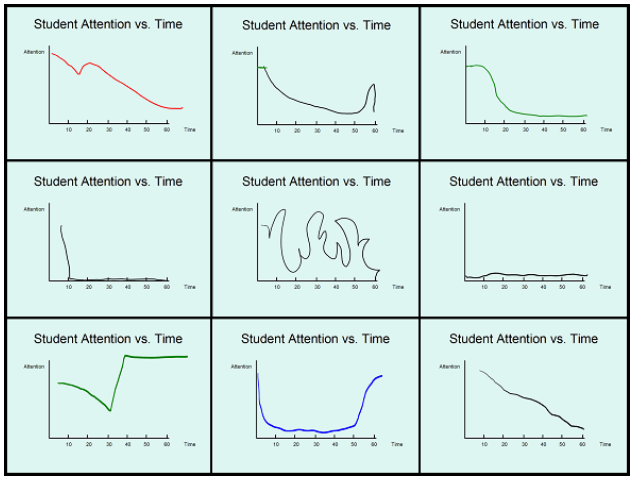

Figure 11 Random examples from the Single Graph dataset.

Table 2 Summary of clusters in each dataset and the size of each cluster.

Dataset Name	# of Clusters	Cluster Sizes						
		1	2	3	4	5	6	7
Single Graph	7	18	11	11	10	8	5	2
Dual Graphs	6	5	4	2	2	1	1	
UI Layout	4	8	3	2	1			
Tree	5	14	3	1	1	1		

As a first step to evaluating our technique we wanted to manually cluster each of the datasets to give us an oracle for testing our accuracy. Table 2 gives a summary for each dataset the number of clusters and the number of solutions in each. For the Tree dataset we defined clusters to be trees with the exact same structure. For UI Layout, we clustered based on the shape of the diagram that was drawn; for example one cluster includes all the circular menus, one cluster includes 4-way radial menus, one cluster includes only rectangular menus, etc. Finally, for the two graph datasets we grouped based on the general shape of the curves.

As might be expected we obtained the best results on the Single Graph dataset. Because our algorithm is suited best for distinguishing gross visual differences, clustering graphs of different shapes seems well suited to our approach. Figure 12

below gives the resulting clustering for this data set. The shaded slides are those slides that are closest to the mean of their corresponding cluster. We can see from this figure that most of the slides in each cluster share the same distinct visual characteristics. In cluster 1, most of the slides slowly descend to the right with a slope of increasing magnitude. Cluster 2 on the other hand is quite different and contains only wavy graphs. Then we have cluster 3, which is again different; it captures those graphs that are flat or ascending. The next cluster captures graphs that drop quickly and rise a little at the end. While clusters 5 and 7 are very similar, cluster 6 again captures a new characteristic – those graphs that have a hump in the middle.

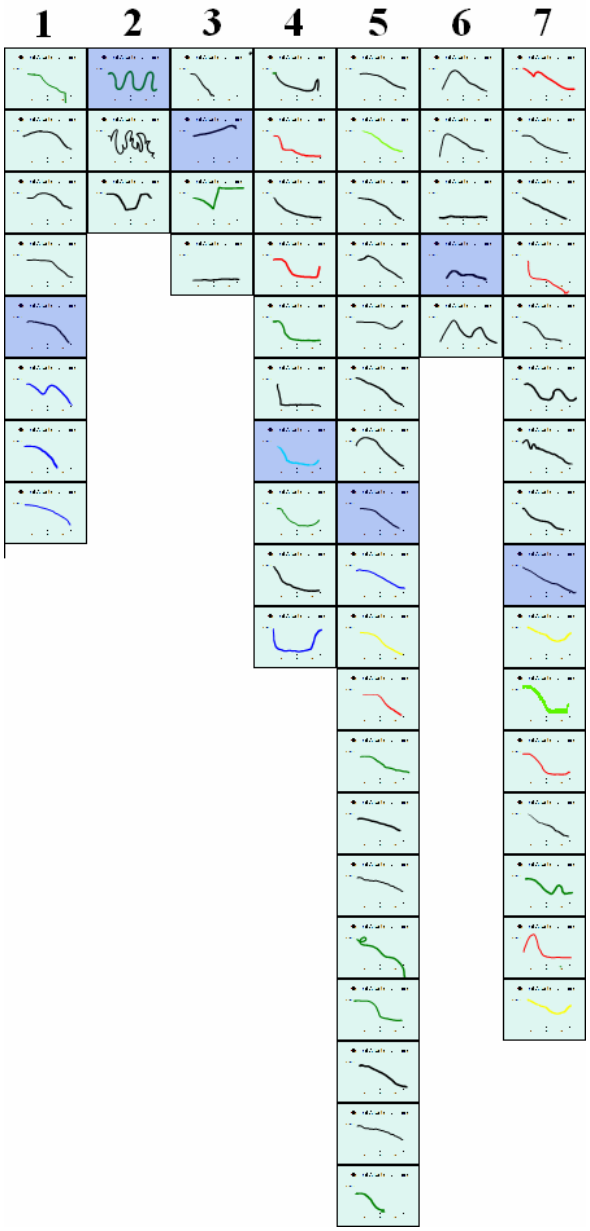

Figure 12 Clustering results for the Single Graph dataset. Each column represents a cluster and the shaded slides represent the solutions for each cluster that are closest to the mean.

We can see from the shaded slides in Figure 12 that the slide closest to the cluster mean is a good representative of the cluster as a whole. A quick glance at these specific slides will quickly give an instructor an overview of the variety of submissions received. Finally, by looking quickly at the size of the various clusters we get a good idea of the popularity of various answers.

The second data set (Dual Graphs) presented an interesting challenge for our algorithm because it contained two different graphs instead of just one. We can see in Figure 13 that, with some exceptions, the clustering worked pretty well. We see that the clusters 3, 5, and 6 are all unique. In cluster 2 we see that two different clusters were incorrectly combined. That is, the slides labeled B are different from the one labeled A. However, the most interesting failure is in the first cluster of the results. Notice, that all of the slides in this cluster had their axes labeled by the student. Our clustering algorithm discovered this similarity and grouped the slides accordingly – even though the graph lines differed in some circumstances. So while the clustering did pick out a feature in common to all the slides in this group, it was the wrong feature. We can expect this type of error to occur often when attempting to cluster arbitrary ink.

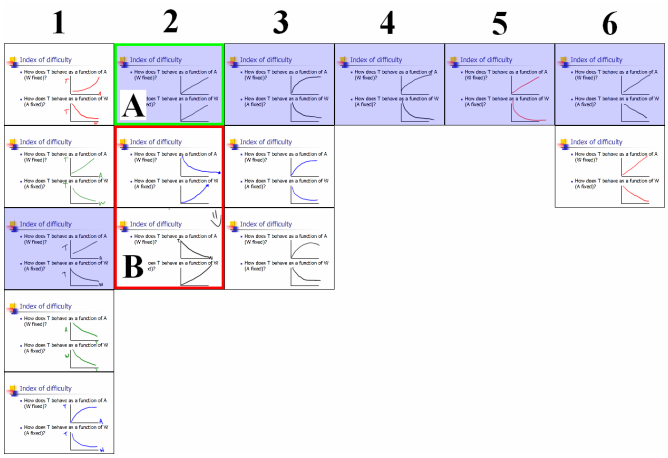

Figure 13 Clustering results for the Dual Graphs dataset. Each column represents a cluster and the shaded slides represent the solutions for each cluster that are closest to the mean.

In the third set we move away from graphs and onto a more complicated structure. Our clustering results for the UI Layout dataset are in Figure 14. Notice that, not surprisingly, the first cluster is well defined, including all the vertical, rectangular menus. Cluster 3 is also well differentiated. The slides labeled A in Figure 14 are both 4-way radial menus and belong in the same cluster. Our algorithm correctly groups them, but incorrectly added the third slide into this cluster. Nevertheless, the slide closest to the cluster mean is still representative of the group.

The most challenging datasets is the Tree dataset. Not only are trees highly structured, but also most of the slides in this dataset fall into the same cluster. We can see from Table 2 that for the Tree dataset 70% of the slides are in a single cluster. This makes searching for the unique solutions very difficult and means that random sampling will usually fail to give good results.

The automatic clustering results for the Tree dataset can be seen in Figure 15. There were five clusters in total and our algorithm was able to correctly identify three of them. First we have cluster 3 that contains all those submissions that start from the root and only have left branches. Then we have cluster 4 that is a more balanced tree. And finally there is the correct answer that is spread across multiple clusters. In Figure 15 we incorrectly grouped the slides labeled A and B into the wrong clusters. Our algorithm was not able to detect that these were different enough to be given a unique cluster. This is one shortcoming of our approach, since we are tolerant to minor differences, focusing instead on overall shape.

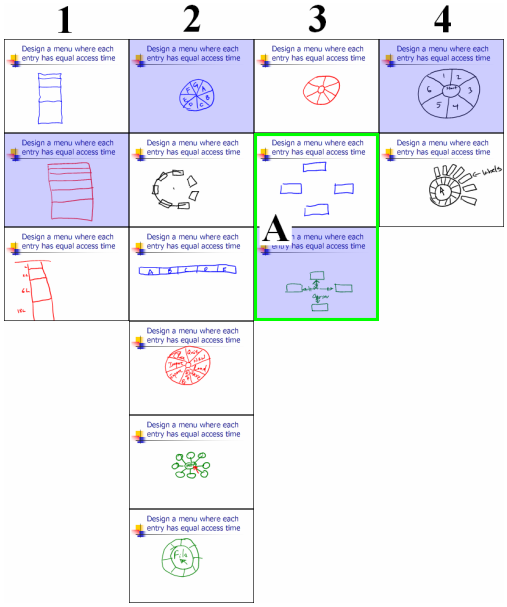

Figure 14 Clustering results for the UI Layout dataset. Each column represents a cluster and the shaded slides represent the solutions for each cluster that are closest to the mean.

For the results above we used versions of the slides where we cleaned up the slides data. We want to determine how aversely our results are affected by excess digital ink on slides. As a test, we took the non-cleaned version of the Tree dataset and ran our algorithm against it. Figure 16 below shows the results of this test. We can clearly see that the non-cleaned version performs much worse than the cleaned version. We do not pick up any interesting clusters. Also our algorithm gets confused by the excess ink, hallucinating features that do not exist. Clearly, removing non-pertinent ink is vital to successful clustering.

We test our algorithm above by asking for a number of clusters equal to the actual number of clusters. However in this context and many others the number of clusters in a group is usually unknown. Therefore, it is important to know how changing the number of clusters affects our clustering. Do we only get interesting results when we choose the correct number of clusters? If we select fewer clusters than actual do we still get good separation? Do we continue to get interesting clusters even when the number of clusters we choose is greater than the actual number of clusters? To answer these questions we use the Single Graph dataset and show how our clustering performs as we increase the number of clusters from two through eight.

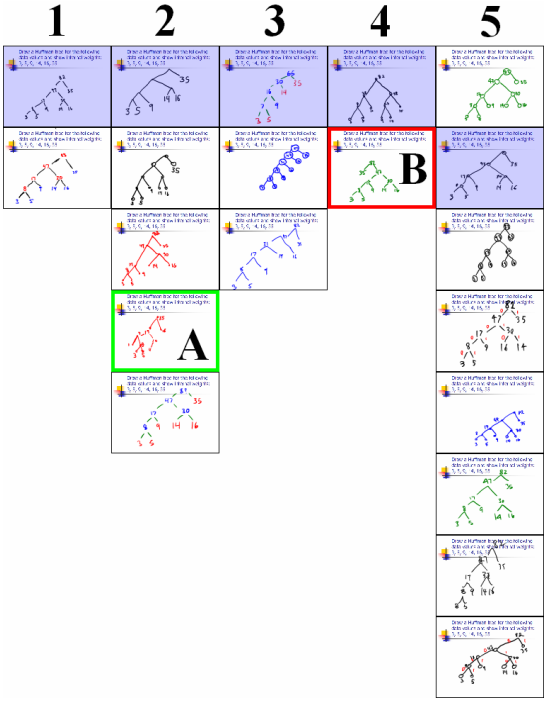

Figure 15 Clustering results for the Tree dataset. Each column represents a cluster and the shaded slides represent the solutions for each cluster that are closest to the mean.

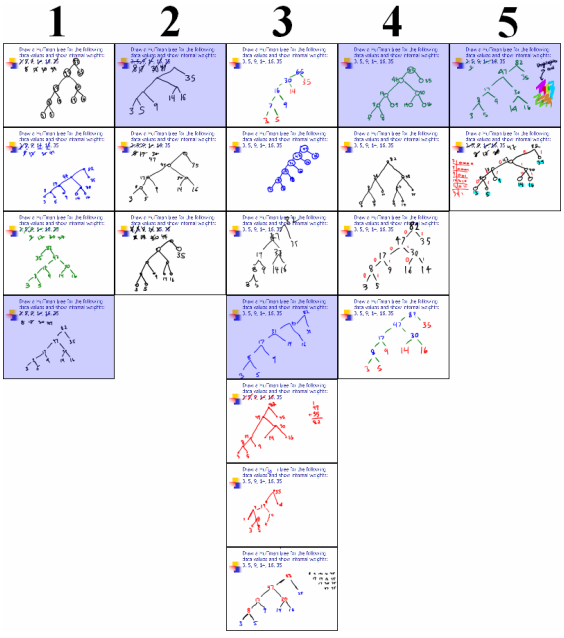

Figure 16 Results from clustering the non-cleaned versions of the Tree dataset slides.

Figure 17 below summarizes the results. Each row in the figure represents a different numbers of clusters and for each row the

columns show the slide closest to the mean of that cluster. By looking at the first row we can see that with only two clusters we get two very different graphs. This makes sense since there is a lot of variation and so we should cluster into two very distinct groups. As we increase the number of clusters, interesting properties of the graphs begin to be separated and clustered. For example we can see that for three clusters we not only keep the flat graph and the graph sloping down, but add a new distinct cluster – one that slopes down then remains flat. This is very distinct from the other clusters. We continue on like this adding a new distinct cluster each time. By the time we reach 5 clusters we start picking up the wavy graph, which remains as we continue. What is very surprising is how stable these clusters means actually are. We can see that most of the same means reoccur even as we increase the number of clusters.

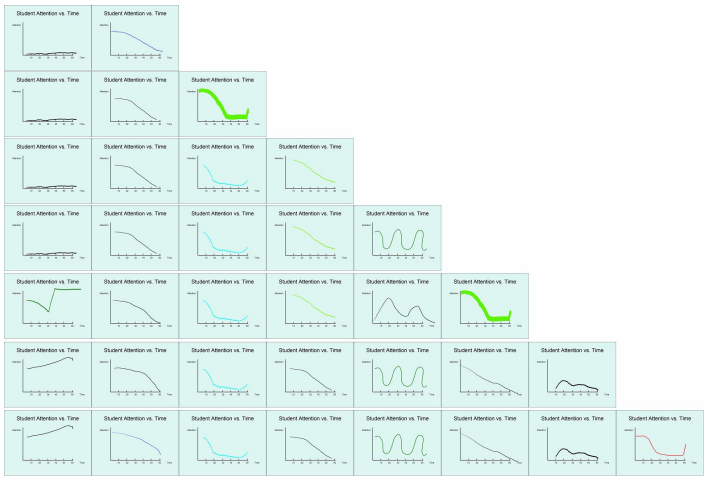

Figure 17 Results showing how increasing the number of clusters affects the cluster mean. Each row represents an increasing number of clusters from 2 to 8. Each entry in a row represents a different cluster mean, displayed is the slide closest to the mean.

From this result we can see that choosing an incorrect number of clusters is not detrimental to using clustering in a classroom environment. In fact the clustering is fast enough that it would be possible to try several different numbers of clusters to look for the best one or until adding more clusters stops giving interesting results. Reducing the number of clusters can also be a way to reduce the cognitive load on an instructor. By intentionally choosing a number of clusters below the actual amount, you reduce the number of slides that the instructor must examine, while still ensuring that differing solutions will be presented.

5. Conclusions and Future Work

In this paper we have looked at the problem of automatically clustering student responses to activities. We have found that even using simple image-based metrics and basic clustering algorithms we were able to generate meaningful clustering. We consider this a very positive result. Considering that our data was from real usage, we have shown that there does exist natural groups within student solutions to exercises and that it is possible to identify these groups.

We were surprised that the direct approach of using ink features was not more successful. In future work we will continue to pursue both techniques. There are many other future directions to take in this work as well including the following tasks:

- Integrate our clustering into the Classroom Presenter application,
- Evaluate our clustering algorithms to quantify if and how much they reduce the cognitive load on the instructor,
- Try different and more advanced clustering algorithms to see if our results can be improved,
- Apply simple forms of training to attempt to improve our results, and

Finally, in future work we would like to tackle textual and numerical responses that we did not address in this work. We believe that the technique for these activities will be much different than those for successful in diagrammatic responses.

6. Acknowledgments

The author would like to thank Professor Richard Anderson and Doctor Beth Simon for their support, direction, patience, and assistance in this work.

7. References

[1] Anderson, R.J., Anderson, R.E., Simon, B., Wolfman, S. VanDeGrift, T., Yasuhara, K., Experiences with a Tablet PC Based Lecture Presentation System in Computer Science Courses, SIGCSE 2004.

[2] Abowd, G., Atkeson, C., Feinstein, A., Hmelo, C., Kooper, R., Long, S., Sawhney, N., and Tani, M., Teaching and Learning as Multimedia Authoring: The Classroom 2000 Project, ACM Multimedia '96, Boston, MA, pp. 187–198, 1996.

[3] Alvarado, C and Davis, R., SketchREAD: A Multi-domain sketch recognition engine, UIST '04, pp. 23-32 , November 2004.

[4] Angelo, Thomas A., and Cross, K. Patricia. Classroom Assessment Techniques: A Handbook for College Teachers. Jossey-Bass, 1993.

[5] Beavers, J., Chou, T., Hinrichs, R., Moffatt, C., Pahud, M., Powers, L., and Van Eaton, J., The Learning Experience Project: Enabling Collaborative Learning with ConferenceXP, Microsoft Research Technical Report MSR-TR-2004-42, April 2004.

[6] Berque, D., Johnson, D., and Jovanovic, L, Teaching theory of computation using pen-based computers and an electronic whiteboard. Proceedings of ITiCSE'01, pp 169-172, 2001.

[7] Borgefors, G., Distance Transformations in Arbitrary Dimensions, Computer Vision, Graphics, and Image Processing, Vol. 27, (1984), pp. 321-345

[8] Bransford, John D., Brown, Ann L., and Cockingv Rodney R., editors. How People Learn: Brain, Mind, Experience, and School. National Academy Press, Washington, D.C., 2000. Expanded Edition

[9] Chen, Y., Wang, J. Z., and Krovetz, R., Content-based image retrieval by clustering. 5th ACM Workshop on Multimedia Information Retrieval, pp. 193-200, 2003.

[10] Crouch, C. H., and Mazur, E., Peer Instruction: Ten years of experience and results. The Physics Teacher, 69 (9) pp. 970-977, 2001.

[11] Dufresne, Robert, Gerace, William, Leonard, William, Mestre, Jose, and Wenk, Laura. Classtalk: A classroom communication system for active learning. Journal of Computing in Higher Eductaion, 7:3--47, 1996.

[12] Judson, E, and Sawada, D., Learning from past and present: Electronic response systems in college lecture halls. Journal of Computers in Mathematics and Science Teaching, 21 (2), 167-181, 2002.

[13] Kam, M., Wang, J., Illes, A., Tse, E., Chiu, J., Glaser, D., Tarshish, O., and Canny, J., Livenotes: a system for cooperative and augmented note-taking in lectures, CHI 2005, pp. 531-540, April, 2005.

[14] Kara, L. B., and Stahovich, T., Hierarchical Parsing and Recognition of Hand-Sketched Diagrams, UIST '04, pp. 13-22, November 2004.

[15] Kaufman, L., Rousseeuw, P, J., Finding Groups in Data: An Introduction to Cluster Analysis, Wiley-Interscience, 1990.

[16] McConnell, Jeffrey J. Active Learning and its use in Computer Science. Proceedings of ITiCSE 1996, 52-54.

[17] Muller, R., and Ottman, T., The "Authoring on the Fly" system for automated recording and replay of (tele)presentations. Multimedia Systems Journal, 8:3, 2000, pp. 158-176.

[18] Penuel, W. R., Abrahamson, A. L., and Roschelle, J. (forthcoming). Theorizing the transformed classroom: A sociocultural interpretation of the effects of audience response systems in higher education. In D. Banks (Ed.), Audience response systems in higher education: Applications and cases, 2005.

[19] Roschelle, J., and Pea, R. A walk on the WILD side: How wireless handhelds may change computer-supported collaborative learning. International Journal of Cognition and Technology, 1(1), 145-168, 2002.

[20] Roschelle, J., Penuel, W. R., & Abrahamson, L. A. The networked classroom. Educational Leadership, 61(5), 50-54. 2004.

[21] Simon, B., Anderson, R. E., Hoyer, C., and Su, J., Preliminary Experiences with a Tablet PC Based System to Support Active Learning in Computer Science Courses, Proceedings of ITiCSE 2004, pp. 213-217, 2004.

[22] Tarel, J-P, Boughortbel, S., On the Choice of Similarity Measures for Image Retrieval by Example, ACM Multimedia'02, pp. 446-455, 2002.

[23] Wolfman, S. A., Understanding and Promoting Interaction in the Classroom through Computer-Mediated Communication in the Classroom Presenter System. PhD Thesis, Department of Computer Science and Engineering, University of Washington, July, 2004.

[24] Zheng, X., Cai, D., He, X., Ma, W-Y., and Lin, X., Locality Preserving Clustering for Image Database, ACM Multimedia'04. pp. 885-891, 2004.