Computer Understanding of Venn and Euler Diagrams

Diunuge B. Wijesinghe, Surangika Ranathunga, Gihan Dias

Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka

{diunuge.10, surangika, gihan}@cse.mrt.ac.lk

Abstract—Venn & Euler diagrams are well-defined mathematical diagram types, which are the major representation methods of Set Theory. Although understanding of different diagram types such as, no research has been done for Venn and Euler diagram interpretation.This paper describes the research methodology and for computer understanding of Venn and Euler diagrams.

Keywords— Diagram Understanding, Venn Diagram, Euler Diagram, Set Theory, Vector Images

1. Introduction

Diagrams is a very important communication medium. This is especially the case with Mathematical diagrams. Although humans can easily interpret these diagrams, mathematical diagram Understanding is a complex challenge in the computer field. Diagram understanding is an important part of various fields such as image database systems, educational diagram grading systems. Significant research has been done to understand mathematical diagrams in few domains such as coordinate graphs as well as charts (bar chats, pie charts). But there is no significant research done to interpret Venn diagrams.

computer of andthat are available ass

Venn Diagrams have been developed by John Venn (1843 - 1923) and is a common representation tool in proportional logic and related branches of mathematics such as Boolean Algebra [7]. Venn diagrams are a specialized instance of a more general notation for representing relationship among a set of classes of concepts referred to as Euler diagrams. Venn diagrams can be described as diagrams that represent pictorial relations among sets. Venn diagrams have been used in a variety of educational contexts, usually in education.

Definition of Venn and Euler diagrams varies throughout the literature [20]. Generally, Venn and Euler diagrams can be defined as a finite set of labelled, closed curves. The closed curves in the diagram partition the plane into minimal regions, where each minimal region is a connected component of the plane inside a set of curves [20]. While a Venn diagram contains all the minimal regions, Euler diagrams omit the empty minimal regions. Normally 2 and 3 sets of Venn and Euler diagrams are being used in most of educational context. Representation up to 6 sets is less complex in the graphical representation. Fig. 1 Shows a Venn diagram and relevant Euler diagram of 2 sets that has an empty set of A∩Bc.



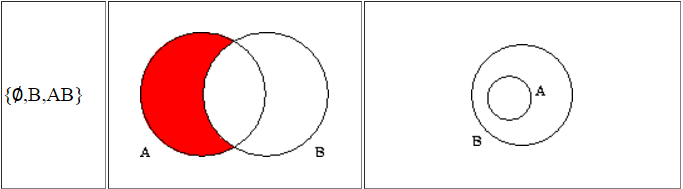


Fig. 1 Example of a Venn and an Euler Diagram

In vector representation of an image, polygons are used to represent an image [21]. Vector graphics based on vectors are defined on x-y axes on a plane. Vector graphics are the main media of information graphics. Since the W3C standard of vector graphics is SVG, we accepted the input as SVG images. SVG has a XML structure, capable of supporting any Venn diagram representation and supports all major browsers, supports mobile devices and scalable.

This paper presents a methodology to extract set details from a vector image and produce the output as an xml structure. First part of this vector image interpretation is the sets identification. In vector images, labelled Jordan Curves are identified as sets. Normally, in SVG sets can be represented with circles, ellipses, rectangles and closed curves. In this research we only consider circles, ellipses and rectangles. However, same concepts can be extended for the arbitrary closed curves as well, though it is not currently handled. In the label identification, only nominal text labels are accepted as labels. Since the label identification is subjected to various ambiguity problems, various heuristics had to be used to identify correct labels.

After identifying the sets, minimal regions are identified. Then remaining text is classified as zone elements. After extracting all the Venn information, extracted data is output in a structured XML format that can describe any Venn diagram. It contains set details and the zone details. Arrangement difference in the vector image of the Venn diagram can produce several XML outputs that are equivalent.

This method has been tested against the Venn and Euler diagrams produced from university students and secondary school students. We have given Venn and Euler diagram question paper with 4 questions to university undergraduate and postgraduate students and collected answers in hand written format. Also we have collected answer from secondary school examination which has Venn and Euler diagram questions. Collected answer sheets are converted to electronic format (SVG) before the parsing. Those Venn and Euler diagrams are parsed using the system and manually checked each parsed diagram for the validation of the system.

This paper is arranged into following sections: Section II describes the related work. Abstract solution details of the complete research, the proposed solution and how the information extraction methods applied are included in the methodology section (Section III). Section IV provides an evaluation of the system. Finally, Section V concludes the paper with possible future extensions to the research.

1. Related Work

In this section, we explore existing research related to mathematical diagram recognition, interpretation, diagram similarity measurements and diagram data representation methods.

Mathematical diagram recognition, understanding and evaluation is a relative new research field. As an early attempt, Futrelle et al [1] presented a diagram understanding system to interpret diagrams based on constraint grammars. The system is capable of handling x-y graphs and gene diagrams on Biological domain [2]. Tsintsifas et al [8] developed a java based framework called DATsys that can be used to understand and evaluate diagrams. They were able to develop the diagram input system that can be scalable to various diagram types. This system was developed as an extension to existing Ceilidh Computer Assessment System [17]. Thomas et al [9, 10, and 12] developed a computer aided assessment system that can handle graph based diagrams such as Entity-Relationship diagrams, flow charts where information can be represented as data nodes and relations between. Diagrams are interpreted using basic set of units called “Minimal Meaningful Units” and introduced a marking criteria based on “Minimal Meaningful Units” which can be applied for diagrams which can be represented using nodes and their relationships. Tsintsifas et al [8] developed a method to assess diagrams in formative assessment in which primary aim to assist the process of learning. They followed the work of Brett et al [15] and developed a feedback system which can work with graph based diagrams. They also discussed a simple evaluation method and develop a grammar for E-R diagrams. Batmaz et al [18] developed a diagram drawing tool which can use for semi-automated database diagram assessment. Huang et al [4, 5] developed a system which can understand chart images. They were able to recognize and identify various types of chart images (both 2D and 3D) and interpret those images and produce an xml output which can be used for further processing.

In the area of diagrammatic reasoning Anderson et al [16] discussed the fundamental components of a diagrammatic processing system, (1) Means to input diagrams which can be a vision component or direct link to a diagram source, (2) Diagram representation to internally represent diagrams, (3) Storage management component and (4) Processing component that can synthesizes and abstracts new knowledge from combinations of diagrammatic and other forms of knowledge representations. In early research, they dealt with many types of input system types based on their research focus. Research from Futrelle et al [1] and Huang et al [4] worked with raw pixel images which extracted from hard documents. They converted those pixel images to Vector Graphics format for further processing. All of CBA (Computer Based Assessment) Systems that we are aware of use custom input interface with various degree of input freedom. Embedded texts in an image is major key to recognize and interpret image information. When it comes to diagram based assessment text plays major role in similarity measurements. When using basic image input formats, pixel based or vector based, as input text association problem arises which leads to ambiguity problems in image understanding (Futrelle et al) [3]. When CBA systems have their own custom input systems which use associated text in images, can resolve ambiguity problems. But those custom input systems provide less degree freedom to the user. Since our primary focus is to give a student a real exam based experience where they use hard papers to answer and has more degree of freedom in drawing a diagram, we selected vector based images as input system.

When it comes to vector based input systems, text label processing is a tricky problem. Some diagram based assessment systems provide restriction to the label choice in order to minimize the association problem. Diagram drawing tool by Batmaz et al [18] provide label which can select as inputs. In DATsys [8] and OpenMark [9] they provide label insertion areas but give freedom of the label which can use to insert. When simulating exam environment we can’t give filtering to labelling (location wise or label selection). Since the labelling has more freedom, text association is a hard problem. Futrelle et al [2] discussed the object association problem and heuristic approach development using special techniques in general. Huang et al [5] developed a machine learning technique based on the decision graph technique to pre-process texts to help text association. Also in the diagram marking text similarity measure is also important. Jayal et al [13] discussed the label similarity problem from Natural Language Processing aspect.

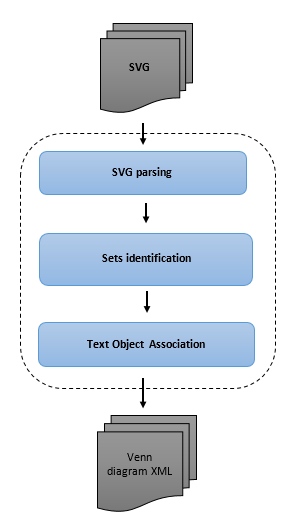
None of above researches addresses the issue of Venn or Euler diagram interpretation and we are not aware of any general Venn or Euler diagram interpretation research work.

Fig. 1 SVG XML structure of a Venn diagram

1. Methodology

In this section we explain the research methodology of the parser. High level architecture of the system and details of the main sections are explained here.

1. High Level Architecture

Input Diagram is given as the SVG image. It contains SVG primitive objects such as circles, ellipse and rectangles with geometric details as well as other details such as presentation details. Since it contains presentation and SVG specific data, SVG image has to be parsed and has to extract the primitive shapes details which are needed to build the Venn Data Information.

Parsed SVG details contains mostly geometrical details such as size, primitive object geometry details and text label details. (Fig. 2 shows high level architecture of our methodology). From the parsed SVG sets are identified by finding the Jordan curves. In this case only circles, rectangles and ellipses are considered. After finding the sets relevant set labels are identified using a heuristic algorithm. Only nominal labels are considered as valid set labels. Few heuristic parameters are considered to classify the set labels such as closeness to the set area boundary. If sets are labels using arrows, relevant arrows are identified and the associated text with the arrow is considered as the set label. Generated temporary label is given to the set in case of there is no associated set label.

After identifying the sets, all the minimal regions are identified and minimal region geometric details are calculated for text association building such as centroid of the minimal region area and size of the area.

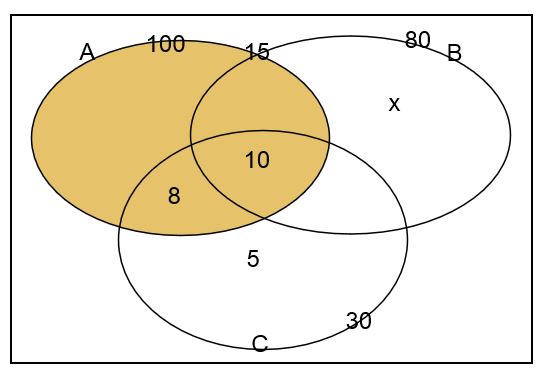
Then the text are associated for correct minimal regions or zones (Set of minimal regions) based on the Venn diagram domain knowledge. Several heuristic algorithm are used to deal with human errors. Heuristic parameter tuning is depend on the source of the Venn diagrams.

Fig. 3 Venn diagrams as a SVG image

After associating the text labels Venn diagram is build using the extracted knowledge and output is created as structured XML which is able to contains only the Venn details without the initial Venn diagram presentation details such as orientations, set curve shapes… etc.

Fig. 2 High Level Architecture of the tool

1. SVG Parsing

SVG format is the standard W3C standard of vector graphics. SVG format is capable of handling Venn diagrams. (Fig. 3 shows Venn diagram drawn using SVG format).

SVG of a Venn diagram contains (Fig. 4 shows XML structure of Venn diagram presented in Fig 3). Basic mandatory SVG features such as size and file type, Grouping details, image title, primitive object (rectangle, line, circle …etc.) shape details, primitive object presentation details (fill, stroke details …etc.), text label details ...etc. In the SVG parsing only usable details to construct the Venn information is filtered.

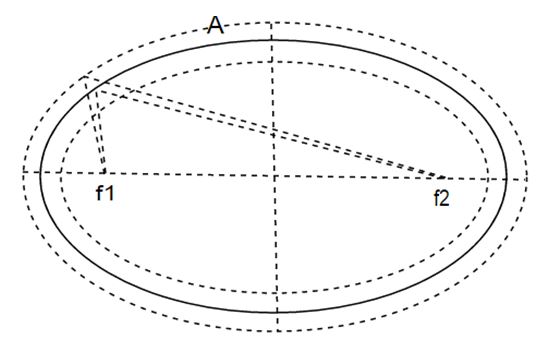
1. Sets identification

In the Venn and Euler diagrams, sets are represented with Jordan curves (non-self-intersecting closed curves). We identified those closed curves in the SVG diagram. In SVG diagrams there are several possible primitive objects which are Jordan curves such as circles, rectangles, ellipses and closed paths. In this research only the only the circles, rectangles and the ellipses are considered. But the same methodology can be extended to the other primitive object which acts as Jordan curves.

1. Set Label identification

After identifying the sets, associated set labels has to be identified. Sets can be labelled in majorly two ways; with arrows and without arrows by putting the text label near the boundary of a set area. If labelled with arrows, arrow ending has to exist near the set area boundary. If associated arrow is found for any set, then the associated text with the arrow tail is considered as the set label.

Fig. 5 Boundary condition check for an ellipse.



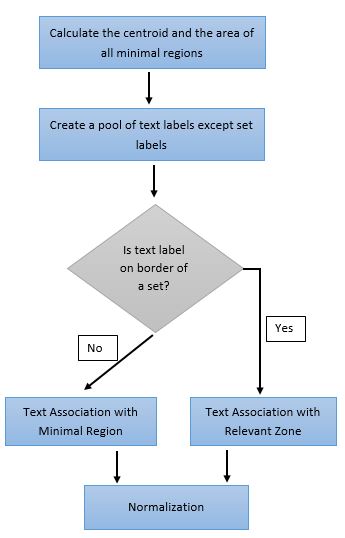
If there is no associated arrow for a set, then closest nominal label near the boundary of the set area is considered as the set label. (Fig 5. Shows the boundary condition check for an ellipse). In the ellipse boundary condition check, only the center coordinates and horizontal and vertical radius are given in the SVG. Using those details focal points and the sum of distant to any point on the ellipse from focal pints are calculated which used to check the boundary conditions. The text label exists in narrow area near the boundary is considered as for possible set labels. Nominal label that is closest and only exist in the considered set area boundary is selected as the set label from those labels.

For sets which doesn’t have a text label is given a computer generated label for further processing.

Fig. 7 Venn/ Euler data output XML structure

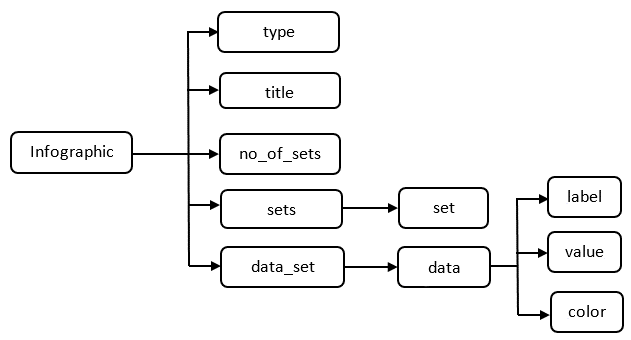
1. Text Association Mapping

Fig. 6 Text association mapping



In the Venn or Euler diagram associated elements with minimal regions are marked on the minimal regions area. Since the minimal regions constructed from combinations of set boundary parts, minimal regions normally have complex boundaries. Since, identification of minimal region boundaries are complex task, minimal region centroid and the area of the minimal region is approximated. (Fig 6. Shows the division of the SVG area for minimal region centroid & area approximation).

Then, the possible text which can be element of a minimal region is filtered using the centroid & minimal region area. From the filtered text, correct texts which are in the region are identified.

1. Venn Data Structure

In theVenn data XML structure, there are five top level tags; (1) type: Type of the diagram (Ex: - “Venn Diagram”), (2) Title: Diagram Title Label (If any), (3) no\_of\_sets: Number of sets in the diagram, (4) sets: set of sets, (5) data\_set: minimal region & zone data.

Data set order is ignored in the Venn information XML Also depending on the set order, different Venn or Euler diagrams may produce set of equivalent solutions. When reading the output XML ignoring the order of sets, it’ll produce the same Venn or Euler data.

1. Implementation and Analysis
2. *Implementation*

The parser is developed using java. Input Venn or diagram images are in a XML format as SVG. In the parser SVG model is present with only relevant details of SVG attributes such as geometric details, most of the styling details are omitted. Reading these input diagram, conversion to the relevant object are done in the initial phase using Java XPath.

Circles, rectangles, ellipses are identified as sets in the sets identification phase from the SVG model. Minimum font size of a SVG image is extracted using all the text labels. In the set label mapping, closeness to the boundary of a set has upper limit of (1.5 \* MINIMAL\_FONT\_SIZE). Nominal label closest only to a set boundary is considered as the set label when arrow labelling is not present. Arrows are presented as lines in the SVG. If an arrow ending are exist near to a boundary of a set, closest nominal label up to the maximum distance of (5 \* MINIMAL\_FONT\_SIZE) to the other end of the arrow is considered as the set label. After set identification sets without any label are given a generated label name.

After the text label association to the minimal regions and zones. Object oriented model of the Venn data is built and the output is generated as an xml format.

1. *Analysis*

For a given set, output xml can have limited number (n!) of equivalent formats depending on the arrangement of sets. Variation of Venn or Euler diagram for 2 sets produce 2 equivalent xml formats.

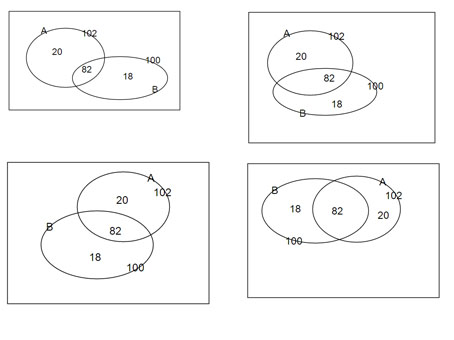


Fig. 8 Variation of Venn diagrams representing same set information

Since in the set theory set order details are not significant, by omitting the sets order in the xml Venn data, equivalency of two diagrams can be measured. Fig 8 shows few variations to a set information. Fig 9. Shows the one of two equivalent xml output.

Venn parser is tested using the Venn diagrams gathered from university undergraduates. Also validated using set of 19 answers from a school grade 10 examination which are drawn in a paper. Those diagrams converted to Vector format as SVG before given to the parser. Parser successfully parsed all the drawn diagrams.



Fig. 9 Venn diagram equivalent variation to 2 sets

As our next step we are conducting an extensive evaluation process to improve this method.

1. Conclusions
2. Future Work

There is a number of possible further extensions can be identified with respect to the research. A few of those possible future research ideas can be listed as follows;

In this research sets can be drawn from few shapes such as rectangles, circles and ellipses. This method can be extended to apply for sets drawn with any Jordan curves. In some Venn and Euler representations one set label can have more than one Jordan curve. This method can be extended to address those representations.

Machine learning approaches can be used to improve the label classification and clustering. Currently machine learning method are not used since we don’t have significantly large image database.

This method can be easily extended into other vector format of images. Since, conversion methods of pixel images format into vector format is already developed. Using those researches this method can be extended to parse printed images.

1. Concluding Remarks

Diagram understanding is a complex problem in the computer research field. Structured diagrams can be dealt with using the domain ontology, but the unstructured drawing understanding is very complex problem. Specially, dealing with human made diagram is a hard due to the ambiguity problems and human errors exist with those diagrams. Also, for some diagrams, diagram definitions are not standardized. Having various structured and unstructured notation for same diagrams types make it more complex to address diagram understanding.

In this research we have successfully established the required methods to understand Venn or Euler diagram which is in a SGV vector format. We believe that solution we have introduced will help to develop systems related to image understanding such as automatic grading systems and image database systems.

Acknowledgment

<To be included>

References

1. R. P. Futrelle, I. Kakadiaris, J. Alexander, C. M. Carriero, N. Nikolakis, and J. M. Futrelle, “Understanding diagrams in technical documents,” in IEEE Computer, 1992, Vol. 25(7), pp. 75-78.
2. R. P. Futrelle and N. Nikolakis, “Efficient analysis of complex diagrams using constraint-based parsing. In Document Analysis and Recognition,” in Proceedings of the Third International Conference on IEEE, 1995, Vol. 2, pp. 782-790.
3. R. Futrelle, “Ambiguity in visual language theory and its role in diagram parsing”, in Proceedings 1999 IEEE Symposium on Visual Languages, 1999.
4. W. Huang, C. Tan, and W. K. Leow, (2005, August). “Associating text and graphics for scientific chart understanding. In Document Analysis and Recognition,” in Proceedings of the Eighth International Conference on IEEE, 2005, pp. 580-584.
5. W. Huang, C. Tan, “A System for Understanding Imaged Infographics and Its Applications,” in Proceedings of the 2007 ACM symposium on Document engineering - DocEng '07, 2007.
6. C.Tselonis, J. Sargeant, and M. M. Wood, "Diagram matching for human-computer collaborative assessment," Proceedings of the 9th CAA Conference, 2005.
7. R. F. Gunstone and R. T. White, (1986). “Assessing understanding by means of Venn diagrams,” Science Education, Vol 70(2), pp. 151-158, 1986.
8. A. Tsintsifas, "A framework for the computer based assessment of diagram based coursework," in Computer Science and Information Technology, 2002.
9. N. Smith, P. Thomas, and K. Waugh, "Interpreting imprecise diagrams," Diagrammatic Representation and Inference. Springer Berlin Heidelberg, 2004. 239-241.
10. P.Thomas, K. Waugh, and N. Smith, "Experiments in the automatic marking of ER-diagrams," ACM SIGCSE Bulletin, Vol 37(3), pp. 158-162, 2005.
11. P. Thomas, K. Waugh, and N. Smith, "Generalised diagramming tools with automatic marking," Innovation in Teaching and Learning in Information and Computer Sciences Vol 10(1), pp. 22-34, 2011.
12. P. Thomas, K. Waugh, and N. Smith, "Using patterns in the automatic marking of ER-diagrams," ACM SIGCSE Bulletin. Vol. 38(3), 2006.
13. J. Ambikesh and M. Shepperd, "The problem of labels in E-assessment of diagrams," Journal on Educational Resources in Computing (JERIC), Vol 8(4), pp. 12, 2009.
14. K. Waugh, P. Thomas, and N. Smith, "Toward the automated assessment of entity-relationship diagrams," 2004.
15. B. Bligh and C. A. Higgins, "Formative computer based assessment in diagram based domains." ACM SIGCSE Bulletin, Vol 38(3), pp. 98-102, 2006
16. M. Anderson and R. McCartney, "Diagram processing: Computing with diagrams," Artificial Intelligence, Vol 145(1), pp. 181-226, 2003.
17. E. Foxley, C. Higgins, T. Hegazy, P. Symeonidis, and A. Tsintsifas. "The coursemaster cba system: Improvements over ceilidh," 2001.
18. F. Batmaz and C. J. Hinde, "A diagram drawing tool for semi–automatic assessment of conceptual database diagrams," 2006.
19. G. Hoggarth and M. Lockyer, "An automated student diagram assessment system," ACM SIGCSE Bulletin, Vol. 30(3), 1998.
20. P. A. Rodgers, “A survey of Euler diagrams,” Journal of Visual Languages & Computing, vol 25(3), pp.134-155, 2014.
21. (2014) The Vector graphics [Online]. Available: https://en.wikipedia.org/wiki/Vector\_graphics