IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-3, Mar 2019 E-ISSN: 2347-2693

MAHI As a Sketching Language

Bhupesh Kumar Singh[1*], Meghna Luthra[2]

[1,2]Department of Computer Science and Technology, SDIET, Faridabad, India

Corresponding author: bhupeshkumarsingh@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i3.11021109 | Available online at: www.ijcseonline.org

Accepted: 20/Mar/2019, Published: 31/Mar/2019

Abstract— The people of various domains, mechanical engineering for machine drawing, electronics engineering for circuit drawing, computer engineering for architectural design etc. generally use sketches. Sketch recognition is defined as the process of identifying symbols that user draw using single or multiple stroke. We have developed MAHI a sketching language based on geometrical rules for multi-domain Sketch Recognition MAHI offers the user more liberty for free-style sketching as well as for grouping the strokes, reducing the complexity for recognizing the sketches. And at the same time it edits the sketches as well as recognizes the sketches efficiently natural environment to the user. The recognition engine interprets the sketches based on the information obtained from the description of domains associated with the input from the user. We have created and tested a Multi-domain recognition engine for Sketch Recognition based on the system use multi-layer architecture for recognition engine**.**

Keywords— Sketch recognition, Bayesian Network, Heuristic

I. INTRODUCTION

The sketching is an essential activity for creative design as it allows a designer to think aloud and to go in for quicker evaluation of new ideas. It also assists the designer's short-term memory and communication with other people. The field of sketch recognition has gained interest in the last few years. Freehand sketching is an efficient supplementary means when we communicate each other. Free Hand Drawing has been one of the creative techniques that an engineer or the scientist considers as the basic approach that can help them to visualize their concepts easily [31].Scientists generally and engineers specifically express thoughts and designs using sketches. Engineers use sketches to exchange designs as a natural method of communication rather than writing or speaking. This is mainly due to the fact that a sketch is a convenient tool to catch enough ideal, also is the conceptual diagram which can be easily understood than words.

Design engineers need a powerful computer based system with sketch recognition to help them design, manipulate and store sketches more effectively than using only papers**.** Increasing the interaction between computers and users in sketch and computer aided system (CAD) has been the reason for the emerging of a few advanced sketch recognition systems

II. RELATED WORK

Sketch recognition is the process of identifying user strokes into meaningful sketch that can be further used. As pen-based input devices have become more common, sketch recognition systems are being developed for many hand-drawn diagrammatic domains. Alvarado [1], Landay and Myers[16], and Stahovich[31] has discussed for mechanical engineering, Hammond and Davis [12], Damm et al.[10] and Lank et al.[17] for UML class diagrams, Lin et al.[19] for webpage design, Caetano et al. [7],and Lecolin et al[18] for GUI design, Do [11] for virtual reality, Mahoney and Fromherz[20] for stick figures , Pittman, et al.[21]for course of action diagrams, and many others. Free hand sketching involves abilities that involve perspective covering several dimensions [32].

These sketch interfaces (1) allow for more natural interaction than a traditional mouse and palette tool discussed by Hse et al.[13] by allowing users to hand sketch the diagram, (2) can automatically connect to a CAD system preventing the designer from having to enter the same information twice, (3) can offer real-time design advice from CAD systems, (4) allow more powerful editing since the shape is recognized as a whole, (5) provide diagram beautification to remove mess and clutter, and (6) use display as a trigger to inform the sketcher that the shapes have been correctly recognized. However, sketch recognition systems can be quite time consuming to build if they are to handle the intricacies of each domain. Also we would prefer that the builder of a sketch recognition system be an expert in the domain rather than an expert in sketch recognition at a signal level.

In the current research, sketch recognition is classified into four directions that is stroke-based, primitive-based, feature-based and composite-based recognitions. The stroke based research directly applies the user hand drawn track. Rubin [22] presents typical trainable gesture recognition. Based on the linear discriminate classifier, it sorts the Gesture into 11 geometry features and 2 dynamic features, and thus, only a few train samples are demanded. But it requires users to finish one symbol in only one stroke. The second one is primitive-based research, which considers the graph as space combination of line, arc, circle, curve and other segment units, eliminating the limitation for user input mode and having a much more efficient recognition result. Segzin and Calhoun [8] point out the basic template matching models. The Sketch was recognized by using curvature and speed feature of the input stroke. However, the extreme of curvature is oscillated due to a bad drawing sketch and the speed feature is unreliable feature which depends on user's drawing style .The third one is feature-based recognition, which extracting the sketch geometric feature to segmentation and recognition. Gross's Electronic Cocktail Napkin Project ECNP is a typical model of multi-stroke recognition based on this kind. The last direction is composite-based recognition. The task is to solve the recognition of graph with complex figure structure and avoid the intricate details of all domains.

Using our framework, in order to build a sketch recognition system for a new domain, a developer need only write a domain description which describes what the domain shapes look like, and how they should be displayed and edited after they are recognized. Thus, the writer of the domain description does not need to know how to program a system to perform sketch recognition. This domain description is then automatically translated into shape recognizers, editing recognizers, and shape exhibitors for use with the customizable base domain independent recognition system creating a domain specific sketch interface that recognizes the shapes in the domain, displaying them and allowing them to be edited as specified in the description.

The inspiration for such a framework stems from work in speech recognition given by Lecolin et al [18] and Do [11], which has used this approach with some success. Hammond and Davis [13] have discussed LADDER as a first sketching language for UML Class Diagram System. The author's claim that the language was designed and developed successfully while describe its domain classes. Unfortunately the application of language is subject to a few limitations of describing shapes such as a fixed graphical grammar, shapes that have a lot of regularity and not too much of the details

The language can describe domains for a few curves and there is difficulty in specifying curves control points. These shape definition languages are not intended for use with an online system and do not provide ways for specifying how to display or edit a shape. Bimber [] describes a simple sketch language using a BNF-grammar. The language describes three dimensional shapes hierarchically. This language allows a programmer to specify only shape information and lacks the ability to specify other helpful domain information such as stroke order or direction and editing behaviour, display, or shape interaction information. Shilman [30] has developed a statistical language model for ink parsing with a similar intent of facilitating development of sketch recognizers. The language consists of seven constraints: distance, delta X, delta Y, angle, width ratio, height ratio, and overlap, and allows you to specify concrete values using either a range or Gaussian. It becomes difficult to describe some shapes using this technique as the language requires providing quantitative discrete values about a shape's probable location. Calhoun et al [8] also uses a semantic network representing the shape in recognition, but the language is limited as it specifies only relative angles and the location of intersections.

III. Methodology

To reduce the limitations of language we proposed a new approach to sketch recognition. Recently we have introduced its interface MAHII (Machine and Human Interactive Interface). In this paper, we describe a language that provides the description of basic shapes of sketch in terms of its geometry, defining the domains that implicitly inherit the geometrical based shapes. Since the dependence of domain sketch recognition system is the main disadvantage for all the systems, therefore to provide a natural and more attractive interface MAHII has been developed, MAHI sketching language places minimum constraints on the users in view of the system being independent of the domain.

MAHI is the language describing the shapes drawn, displayed, and edited subject to the domain. It has been successfully tested to multi-domain recognition system along with a code generator that parses MAHII interface based on domain description and generates the codes pertaining to jess code to give complete recognition to the system. MAHI identifies a few basic shapes through which recognition engine can identify its corresponding domains and then it connects the particular domain to itself through an interfacing. This way, the system provides the developer to define the domains independently which describe what the domains shapes look like, and how they should be edited and displayed after they are recognized. The recognition engine is followed by heuristic engine that finally recognizes the input sketch completely. We discuss components of MAHI (language), contents of the sketch grammar in MAHI, Testing, and System- Implementation along with conclusion in the subsequent sections.

A. SYSTEM FRAMEWORK

It basically contains three blocks: Domain Class block, input refining block and recognition engine block. Domain classes contain the description of a particular domain. So, a programmer needs only to interface his code with the desired domain. Thus, it simplifies the developer's job. Interface shows the translation scheme used between domain classes and recognition engine. It bridges the gap between the two. Input Refining block takes raw input or a roughly designed figure and finally gives a noise-free, smooth and properly drawn figure. The main thing about this system is that it also displays the name of the recognized object.'

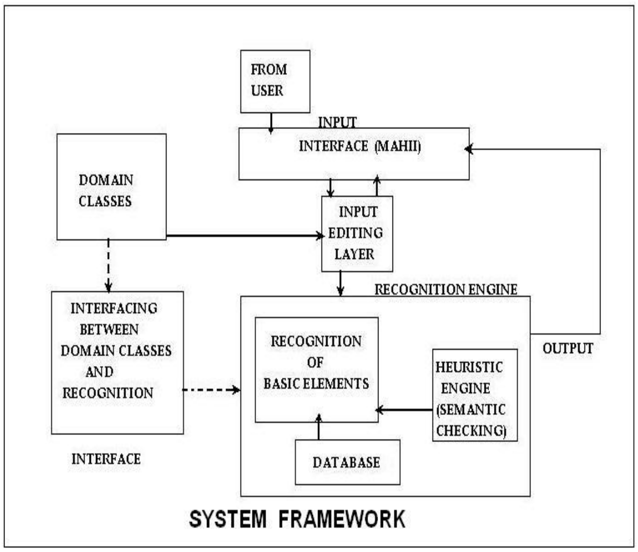

Fig. 1

To explain the working of the recognition system, an example of a hut is taken. The user draws a rough sketch of a hut and then system refines it. Then user input is segmented using control flow technique. The system recognizes all segmented shapes using the domain description and databases of shapes. Database contains the sample images with recognized output. The system edits all the recognized shapes to display a smoothened image. The next most important task which is, the system performs to make a structure out of the given basic shapes. The Heuristics inference engine uses the Database set and takes the edited image as input and makes the complete structure and also performs semantic checking to clarify whether the structure shows some meaningful object.

For sketching and recognizing the user drawn sketches, we designed interface known as Machine and Human Interactive Interface (MAHII). This system MAHII is built so that it proves to be useful to a computer-expert and even to a novice person. It contains some of the known commands of computer as well as normal behavioural commands that a layman can use in sketching the diagrams. The System MAHII also gives the user more liberty to draw sketches freely and naturally. Moreover, the user need not to learn every capability of the system but still he can draw the sketch and get accurate results of recognition. Various design issues have been predicted from previously sketch-based systems. We arrive at the following design issues that can help in overcoming these issues successfully. The primary goal of the interfacing mechanism is to connect the domain class description to the recognition engine only when its few corresponding elements are recognized by the recognition engine through Heuristic engine using a database containing shapes

B. ABSTRACT VIEW OF A SYSTEM FRAMEWORK

The system framework has been discussed which gives the information about the working of the system. Now, this framework is implemented as the architecture of the system and the corresponding interface is named as MAHII. The architecture is explained below.

User provides input to the system by drawing sketches on the MAHII. This interface gets the sketch input from the user and sends it to the input refining block where input is refined by removing noise from the sketch input and then it is sent to the recognition engine as well as to the MAHII window. The edited or refined sketch is sent back to MAHII window because it displays the user the edited version of the sketch drawn at previous time instant. If it is not according to the user's requirement, user can change it by using simple commands that are of common behaviour. We explain such commands in a later section. When the edited input is sent to MAHII, it is also sent to the recognition engine also. After recognition, the output of the recognition is sent to the MAHII window. The flexibility and ease use of this system reflects here that user needs not to know when he should change the sketch mode, edit mode or recognition mode. The system takes care of these things by itself.

If a user draws a stroke say in 2 seconds and next stroke after 4 seconds then he draws the third stroke after 5 seconds and so on, the system takes the average of the first few time gaps between the consecutive strokes. Thus, it calculates the average of time difference then sets the systems' recognition time greater than the calculated average waiting time. The system does this for every user. So, if a user is faster in sketching, the system provides him results in less time and for a novice user, it displays the result according to the time taken by the user to draw next stroke. Thus, the system changes itself according to the user. The interface, which we represent in the framework that is between domain class description and recognition engine, helps the system to work in a multi-domain environment while it works actually over a single domain. In comparison to the earlier multi-domain sketch systems discussed by Alvarado[3] where the user had to select explicitly the domain that he wants to use, MAHII does not impose any such requirement. The recognition engine actually requires the domain descriptions.

Moreover, the database contains knowledge about recognized shapes and their particular domain. So, when the system needs to select a particular domain, it identifies few basic shapes or elements that are parts of a particular domain followed by the class description with the recognition engine.

C. MAHI Layered Architecture

The MAHII interface uses modular approach. Basically the architecture of MAHII consists of three layers namely.

- *Input Layer*

The input layer is the window that is visible to the user for drawing the sketches. The user can draw frequently the sketches and these are used as input to the system. The user even may not have the knowledge of the functionality of the computer. As the user draws the symbols or sketches, he gets the edited image as well. The user need not wait for the system as and when to draw a sketch on a pen-based system. This window after taking the input from the user not only displays the smooth sketch but also displays the recognition layer's result. We consider many issues and tackle various problems encountered in the earlier sketch interfaces.

In previous recognition interface user draws in one window and gets the result in the other window. So, user switches explicitly between the two. MAHII makes this task easier. It uses a single window for all purposes and it does not have any synchronization problem. Further, many interfaces provide menu based tools like edit, undo, redo, cut, copy, paste. We make these tools available in our system in addition to some extra natural commands that can easily be used by novice.

SkRUI discussed by Alvarado [3] in which user cannot draw new items when it is in edit mode whereas MAHII does editing by it and allows the user to draw the new sketches as well as to modify the previously drawn items. For this purpose, it uses buffers for temporary storage of previously drawn sketches and whenever the user wants to copy a sketch, he just needs to redraw the sketch over the existing one.

- *Editing Layer*

It gets the input from the input layer and refines it by removing the noise from the sketch and increases the contrast of faded image/sketch and providing input to the recognition layer. This layer performs the editing of the sketch input. This layer includes the functioning of image filtering, image segmentation and image editing.

- *Recognition Layer*

This is the only layer that is given the power of recognition named as recognition layer. It gets the input from the editing layer as mentioned above and uses heuristic engine to perform searches for the relevant details from the system database.

In this Layer it recognizes the basic elements belonging to a particular domain using the Heuristic engine and database of samples of shapes and then identifies to which domain those elements belong to and then selects a particular domain class description out of many and connects it to the recognition engine. This layer recognizes the output and displays the result to the input-output layer of the interface

IV. MAHI AS A SKETCH LANGUAGE

MAHI domain descriptions include defined shapes and other information related to the recognition process, such as stroke order and stroke gradient. It contains the defined geometrical shapes with minimum constraints and takes into account its editing behaviour, display methods and syntax for specifying a domain description. We create a domain description reusing defined shapes to build the desired shape hierarchically has been discussed.

A. Basic Elements of MAHI

In MAHI, We define the geometric elements in terms of the entities given: The basic core entity (BCE) unit defines a point, a collection of BCE forms a core entity (CE) defined by a line or an arc and the collection of BCE and CE forms a derived entities (DE) as open or closed shaped.

B. Geometrical Analysis

In MAHI, we are able to draw any of the shapes based on entity BCE, CE, and DE as defined by point, arc, line, rectangle, square, circle, ellipse, polygon, curves, surfaces, volume etc. Here a shape is a geometrical structure with basic input as geometrical rules associated to recognition process such as the stroke. Using the shape properties, an arbitrary shape is extended or reduced to the desired shape. The language enables more accurate and fast sketch recognition by using bottom-up as well as top-down recognition in view of mathematical definition of line being defined as locus of point and point being defined as limiting case of a line. Thus, a point is recognized and is used to identify and be identified as a line or arc since the line or arc is extended to form any other extended shape(s) or extended shape(s) merged to a line or curve.

C. Shape Geometry

Generally, we define a point as a precise location or place on a plane, usually represented by a dot. We remark here that geometrically it is appropriate to define a point in terms of line segment. We define a line segment as the shortest distance between any two given points, whereas a point is a line segment of join of two points whose distance in one dimensions or gradient in two dimensions tend to zero. Further, if the distance between two consecutive points is not Minimum then it will define an arc and consequently piece-wise continuous arcs will give rise to a curve.

D. Shapes with Constraints

A number of geometrical shapes can be defined by the algebraic equations subject to the given constraints. For example ax+by+c=0 subject to the condition $d \le x \le e$ represents a number of lines including family of parallel, perpendicular lines etc. in a plane of two dimensions for different values of a, b, c, d and e. In other words if the sketch grammar consisting of algebraic equation subject to the constraints as given above is represented geometrically, then we find that it is far superior to express constraints in terms of mathematical language than otherwise given by traditional constraints such as rotatable, angle, horizontal, vertical, etc. has been discussed by Caetano et al. [7] and Shilman et. al [30] This language discussed by Hammond and Davis [14], Shum et al.[28], Sezgin et al.[27] and Alvarado and Oltmans [5] containing a library of pre-defined shapes with pre-defined constraints such as horizontal, posSlope, above left etc including IsRotable, angle L, vertical constraints to define the orientation subject to the relative co-ordinate system, which seems cumbersome to the user. However, MAHI ignores all these complicated expression and having inbuilt mathematical system, the shape takes the form with ease than in Ladder.

In MAHI gradient for any line in two dimensions is given by mi= -(coefficient(x))/(Coefficient(y)), i=1,2 as such a user can draw any line following the language which naturally is one of these Perpendicular, parallel, collinear, coincident, same side, opposite side meet, intersect, tangent, center below, center above, positive slope, negative slope, etc. The components and properties of these lines can be used hierarchically top-down in the shape descriptions. The other related new shapes are drawn by extending the properties of a line, rectangle, parallelogram, circle, to any of the shapes.

E. Defined-Editing Behaviour

MAHI knows when and how to recognize, edit and display the shapes. Editing and displaying are the important components of sketch interface that vary as per different domains. The editing of the object helps the sketcher to make the recognition not only possible-shape but makes it beautiful by removing the clutters followed by editing gestures consisting of event for each shape.

```
Loop (for i = 1 to n)
(
trigger DoubleClickHoldDrag side 'i'
action
translate this
setCursor DRAG
showHandle MOVE side 'i' // where 'i' is the side number
trigger holdDrag side 'i'
action
translate this
setCursor DRAG
showHandle MOVE side'i' … side'n'
 )
```

F. Curser Behaviour

Use of Editing (curser) in MAHI is dependent on its mode that, one can use pen based editing that is when the curser is in pen mode (sketching) in which case the user can draw sketches and when it is in curser mode it means the editing mode. Thus, sketching and editing use distinct pen motions. One of the editing behaviour in MAHI is if one clicks and holds the pen on the surface of the rectangle and drags the pen, the entire rectangle will translate along with the movement of the rectangle. In this way the rectangle along with the vector is translated, scaled, and rotated as one whole shape. MAHI editing behaviour of triggers include click, double click, hold, hold drag, encircle, etc. along with their subsequent application.

G. Display Methods

Controlling is a vital part of sketching interface when the user recognizes the shape after he has drawn that sketch. MAHI defines recognition layer that has the power of the recognition of the sketch. It gets the input from editing layer as defined above and uses heuristic engine to perform semantic checking for the relevant details from the system database. It also includes the concept of domain independence with reference to multi-domain. It recognizes the output and displays the results to the input/output layer of MAHI interface.

V. DOMAIN DESCRIPTION OF MAHI

We define domain description as an aggregate of elements. Here elements are referred as the list of domain shapes and group shapes

A. MAHI Shape-Definition

A shape is defined if it is associated to a domain belonging to domain description. In particular, geometrical shapes are defined as the blocks belonging to the domain shapes of multi-domain. We can form new desired shape using MAHI domain description. The MAHI (Sketching Language) based on the domain description is translated into shape recognizer such as (geometrical) components & constraints. MAHI has the exhibitors and editors representing the display and edit sections respectively to be used in conjunction with the recognition systems to obtain a drawn sketch.

```
define shape Rectangle
description "A quadrilateral with four sides and all angles 90 degree"
```


Components
Line side1
Line side2
Line side3
Line side4
Clauses // Constraints
coincident side1.p1 side2.p1
coincident side2.p2 side3.p2
coincident side3.p3 side4.p3
coincident side4.p4 side1.p4
parallel side1 side3
parallel side2 side4
perpendicular side1 side4
perpendicular side1 side2
perpendicular side2 side3
perpendicular side3 side4
equalLength side1 side3
equalLength side2 side4
Aliases
Point side1 side2.p1
Point side2 side3.p2
Point side3 side4.p3
Point side4 side1.p4
Editing
Edit (side1 side2 side3 side4)
display
cleaned-strokes side1, side2, side3
original-stroke side4
color blue

Domain description of rectangle using MAHI as a sketching Language

B. *Recognition*

In MAHI the recognition system contains domain independent module that can recognize, exhibit, and edit all of the basic core, core, and derived shapes. Low–level recognition is performed on the drawn sketch even without having been identified as an editing gesture. The modules not depending on the domains determine if the stroke can be classified as a point, line, arc, circle, ellipse, or other technique given by Sezgin [28]. In LADDER Domain shape recognition system only chooses one interpretation of a single stroke. In order to ensure only one interpretation is chosen, each shape has an ID, and each shape keeps a list of sub-shapes, including its stroke. At any particular time each sub-shape is allowed to belong to only one final recognized domain shape. Here the figure specifies each shape's ID followed by the IDs of all the sub-shapes corresponding to a stroke having two interpretations of being a line or arc. If shape has polyline interpretation the stroke is divided into segments so that the sub-strokes added as sub-part give the original full stroke as the continuity of the line segment. In MAHI line and arc also share the same stroke as the sub-part. Shape is a line giving rise to piece-wise lines and since, piece-wise functions are continuous or discontinuous, therefore the set of final shapes can share any sub-shape line or arc. However, in case of Ladder discussed by Hammond and Davis [14] the set of final shape cannot share any sub-shape which prevents the curve and polyline for both being chosen in final interpretation. This limitation with this bottom up recognition method is that if the primitive shape does not provide correct interpretation of a stroke, the domain shape recognizer will never be able to correctly recognize a higher-level shape using this stroke. Whereas we use top-down recognition method to overcome even if primitive shape does not provide correct interpretation of a stroke, domain shape recognizer will be able to correctly recognize a higher-level shape.

C. *Sketch Recognition*

Sketch recognition is the process of combining lower-level shapes on the screen to create higher-level shapes. These higher-level shapes are defined by the sub-shapes which they are composed and constraints specifying how the sub-shapes fit together. To recognize all higher-level shapes, the recognition system must examine every possible combination of shapes, as well as every permutation of these sub-shapes. This implies that any straight forward algorithm would take exponential time, (as in case of the geometric features defining the geometric attributes of the shapes as which is clearly impractical for any non-trivial sketch.

We transform a grammar into a domain recognizer of hand-drawn shapes. The translation process is on parallel lines to the work done on compilers, as done by with regard to visual language worked by Costagliola et al [9]. This compilers allows a user to specify grammar for visual language then compile it into recognizer subject to its being valid syntactically. We translate domain description into the user interface for that domain. We have built a system REFMD (Recognition Engine for MMAHI Description) that automatically generates a jess code for sketch recognition system from MAHI domain description. The domain description of MAHI is translated into shape recognizer with the help of component and constant solver, exhibitors and editors that recognize, edit and display each domain shape within the generated domain user interface.

A. *Recognition*

Our approach is to recognize shapes by how they look rather than how they were drawn which is based on using mathematical description of each shape. We describe the shapes using the MAHI domain, and automatically generate shape recognizers from these descriptions as discussed by Hammond & Davis [14], Shum et al.[28], Sezgin [27], Alvarado and Oltmans [3]. Our approach differ from all these, we used heuristic search as a recognition algorithm to use of parallel search in multiple domain.

VI. Results and Discussion

We have developed MAHI LADDER domain descriptions for a variety of domains as shown in Figure3, the descriptions have been automatically translated into a sketch interface which recognizes, displays, and allows editing in real time as specified by the domain description. These descriptions include over one hundred shapes, some containing text. Figure 3 shows the unrecognized and recognized strokes from a drawing made in an automatically generated finite state machine, and mathematics sketch interfaces.

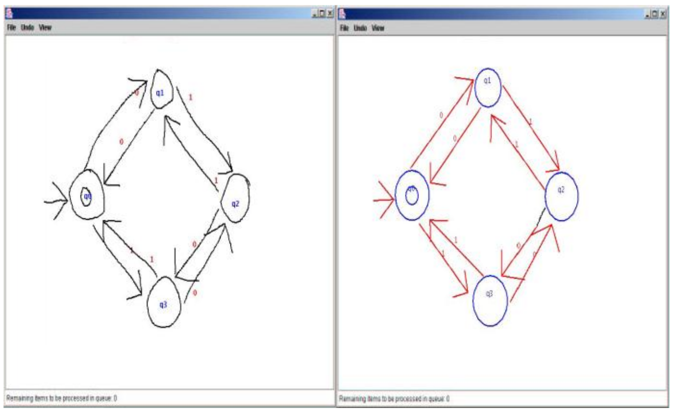

Figure 2: Auto-generated finite state machine interface

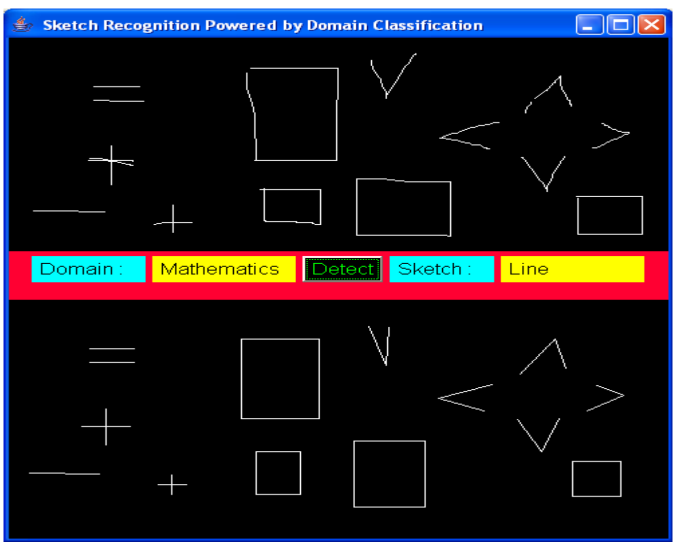

Figure 3: Mathematical Notations

VII. Conclusion and Future Scope

This paper describes Heuristic with indexing algorithm for multi domain sketch recognition. During stroke processing, each stroke is broken down into a collection of primitive shapes (Derived Entity) during shape recognition; the properties of the strokes and shape are searched heuristically in each domain. Its shape group describes how domain shapes interact and provide information in top-down as well as bottom–up recognition. Finally, for improving the recognition performances and the usability of the system it would be useful to incorporate information from external knowledge sources in the recognition process. As an example, the user could explicitly indicate the interpretation of a set of strokes interacting with the user interface or giving voice commands. So, external information can be incorporated in the system independently from the nature of the supplied information.

REFERENCES

[1] Alvarado, C. (2000) 'A natural sketching environment: bringing the computer into early stages of mechanical design' Master's thesis, MIT.

[2] Alvarado, C.(2004) 'Multi-domain sketch understanding' PhD thesis, Department of Electrical Engineering and computer science, Massachusetts Institute of Technology.

[3] Alvarado, C.(2004) 'Sketch Recognition And User Interfaces: Guidelines for Design and Development', AAAI Fall Symposium on Pen-Based Interaction.

[4] Alvarado, C. Davis, R. 'Sketchread a multi-domain sketch recognition engine' In Proceedings of UIST '04, pp.23–32.

[5] Alvarado. C. and Oltmans. M. (2002) 'A Framework for multi-domain sketch recognition' Proceedings of AAAI Spring Symposium on sketch Understandings, Stanford University, pp.1-8.

[6] Alvarado, C. Oltmans, M. Davis, R (2002) 'A Framework for multi-domain sketch recognition' Proceedings of AAAI Spring Symposium on sketch Understandings, Stanford University, pp.1-8.

[7] Caetano, A. Goulart, N. Fonseca, M. Jorge, J. (2002) ' JavaSketchIt:issues in sketching the look of user interfaces' Sketch Understanding. Papers from the AAAI Spring Symposium.

[8] Calhoun, T. F, Stahovich, T(2002)' Recognizing multi-stroke symbols' Proc. of AAAI 2002 Spring Symposium: Sketch Understanding Workshop

[9] Costagliola, G.; Tortora, G.; Orefice, S.; and Lucia, D. 1995. Automatic generation of visual programming environments. In IEEE Computer, 56–65

[10] Damm, C.H. Hansen, K.M. and Thomsen, M. (2000) 'Tool support for cooperative object-oriented design: gesture based modeling on an electronic whiteboard' In: CHI, pp.518–25.

[11] Do, EY-L.(2001) 'Vr sketchpad—create instant 3d worlds by sketching on a transparent window' In: de Vries B, van Leeuwen JP, Achten HH, editors. CAAD Futures 2001. pp.161–172.

[12] Hammond, T. and Davis, R.(2002) 'Tahuti: a geometrical sketch recognition system for uml class diagrams' AAAI Spring Symposium on Sketch Understanding, pp.59–68.

[13] Hammond, T. and Davis, R. (2003) 'LADDER: A language to describe drawing, display, and editing in sketch recognition' In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico.

[14] Hammond, T. and Davis, R.(2005) 'LADDER: A sketching language for user interface developers. Elsevier, Computers and Graphics, 28. pp.518–532.

[15] Hse, H. Shilman, M. Newton, A.R. and Landay, J.(1999) 'Sketch-based user interfaces for collaborative object-oriented modeling' Berkley CS260 Class Project.

[16] Landay, J.A and Myers, B.A.(1995) 'Interactive sketching for the early stages of user interface design' In: Proceedings of CHI '95: Human Factors in Computing Systems, pp.43–50.

[17] Lank,E. Thorley JS, and Chen SJ-S, (2000) 'An interactive system for recognizing hand drawn UML diagrams' In Proceedings for CASCON, pp. 7.

[18] Lecolin et, E.(1998) 'Designing guis by sketch drawing and visual programming' In: Proceedings of the International Conference on Advanced Visual Interfaces (AVI 1998), AVI, New York:,ACM Press, pp.274–6.

[19] Lin, J. Newman, MW. Hong, JI. Landay, J. (2000) 'DENIM: finding a tighter fit between tools and practice for web site design' In: CHI Letters: Human Factors in Computing Systems, CHI, pp.510–7.

[20] Mahoney, J.V. and Fromherz, M.P.J.(2002) 'Handling ambiguity in constraint-based recognition of stick figure sketches'SPIE Document Recognition and Retrieval IX Conference, San Jose, CA.

[21] Pittman, J. Smith, I. Cohen, P. Oviatt, S. and Yang, T.(1996) 'Quickset: a multimodal interface for military simulations' Proceedings of the Sixth Conference on Computer-Generated Forces and Behavioral Representation, pp.217–24.

[22] Rubine, D. 1991. Specifying gestures by example. In Computer Graphics, volume 25(4), 329–33

[23] Sahoo. G. et al. (2009) 'MAHI: Machine and Human Interface' International Journal of Image Processing (IJIP), Vol. 3, Issue 2, pp.80-91.

[24] Sahoo, G. and Singh, B.K.(2008) 'MAHII: Machine And Human Interactive Interface' International Journal of Image Processing(IJIP), Vol. 2, Issue 3, pp.1-11.

[25] Sahoo, G. and Singh, B.K. (2008) 'A New Approach to Sketch Recognition using Heuristic' International Journal of Computer Science and Network Security, Vol. 8, Issue 2, pp.102-108.

[26] Sahoo, G. and Singh, B.K. (2008) 'A Human Detector and Identification System' International Journal of Computer Science, Systems Engineering and Information Technology, Vol. 1, Issue 1, pp.39-44.

[27] Sezgin, T. M. Stahovich, Thomas. and Davis, R (2001) 'Sketch based interfaces: Early processing for sketch understanding' In The Proceedings of 2001 Perceptive User Interfaces Workshop (PUI'01),Orlando, FL.

[28] Sezgin, T.M.(2001) 'Feature point detection and curve approximation for early processing in sketch recognition' Master's thesis, Massachusetts Institute of Technology

[29] Shum, S.J.B. MacLean, A. Bellotti, V.M.E. and Hammond, N.V.(1996) 'Graphical argumentation and design recognition' Human-Computer Interaction, 12(3), pp.267-300.

[30] Shilman, M. Pasula,H. Russell, S. and Newton, R.(2002) 'Statistical visual language models for ink parsing. In Sketch

[31] BhupeshkrSingh, Luthra M, (2019) 'Hand Sketching a Necessary Tool for Learning Computation & Automata Theory',Inpress

[32] BhupeshkrSingh, Luthra M,(2018)' Free Hand Sketching an Essential Tool For Unified Modelling Diagrams in object Oriented Approaches'.

Authors Profile

Prof. (Dr.) Bhupesh Kumar Singh received B.E. in (Computer Science & Engineering) from Government College of Engineering. Tirunelveli, Anna University, India in the year of 1999, and Ph. D in Sketch Recognition from BIT Mesra, Ranchi in the year 2013. Currently working as Professor & Director at Satyug Darshan Institute of Engineering & Technology, Faridabad, India. His research interest includes theoretical computer science, image processing and pattern recognition. He is Having 19+ years of Teaching and Research Experience.

Ms Meghna Luthra M-Tech in Computer Engineering and B-Tech in Information Technology; with 5+ years of comprehensive professional teaching and Research and hands-on experience in Technology to B-Tech and M-Tech students. Strong technology background complemented industry needs mind-set. Exposure to all major CS subjects with work experience across other departments using Agile methodology. Primary faculty for Data Analytics and Business Intelligence.. Her research interest includes theoretical computer science, image processing and pattern recognition.

