International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 3, ISSN No. 2455-2143, Pages 230-235
Published Online July 2019 in IJEAST (http://www.ijeast.com)

MODELLING AND ANALYSIS OF 2-D STRUCTURES USING MATLAB IMAGE PROCESSING AND SAP 2000 API

Ronak kalal
Student, M. Tech (Structural)
Department of Applied Mechanics,
GCE Karad, Maharashtra, India.

J. P. Patankar
Professor,
Department of Applied Mechanics,
GCE Karad, Maharashtra, India.

Abstract—**With the development and advancement in the structural software packages, analysis and the designing of the structures became easy and convenient. But still one of the most labour-intensive and time-consuming steps associated with structural modelling is defining the geometry of the structure for analysis. Digital image processing has minimized efforts of day to day tasks in many areas. The aim of this paper is to present a method to automate the analysis of 2-D trusses. To get the analysis done just by capturing an image of a truss drawn on the piece of paper, we will use morphological image processing techniques in MATLAB to get the required information from the input image and integrate SAP2000 API in MATLAB to model and analyse the truss by the extracted information from the image.**

Keywords- **Automated structural analysis, truss, morphological image processing, SAP2000 API**

I. INTRODUCTION

Smartphones with cameras are becoming so cheap and are in most people's reach. But still, the structural analysis software packages are limited to computers because it's not easy to provide the same software interface as of the computes on the smartphones because of screen size indifference. Although with an alternate approach, these software packages can be taken to smartphones at least for analysis of 2-D structures. Potential of image processing is limitless and can revolutionize the way we interact with the software in the civil engineering field if implemented. The software will become more interactive, easy to use and less time-consuming. In this paper, we are passing an input image of hand-sketched or computer-generated truss through an algorithm, in which image will be step by step segmented with the help of image processing tools of MATLAB and will be solved with the integrated codes of SAP 2000 API to provide the solution for the truss. Certain notations need to be assumed for drawing a truss on the piece of paper for analysis so that the algorithm could work properly. Supports will be in a triangular shape. Loads will be represented by arrows. Coordinates and magnitude of loads will be written near to corresponding nodes and arrows respectively. Only x and y coordinates will be written and with separated comma. Units will not be mentioned in the image and considered KN-m-C by default, and can be changed if required. The diameter of the nodes will be assumed large enough to survive erosion. Typical images are shown in figure 1 and 2 to represent the notations assumed.

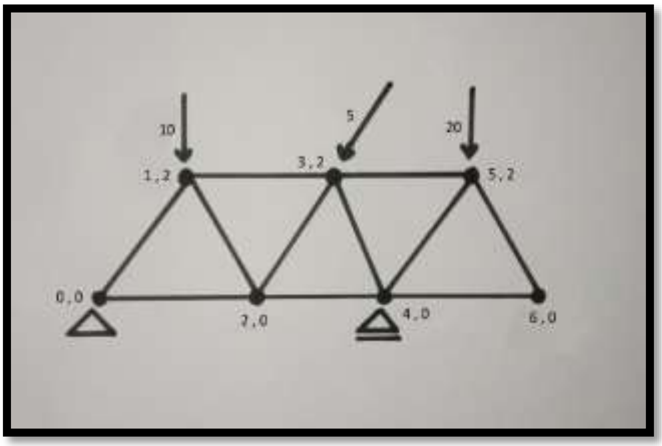

Figure 1 : Typical truss example 1

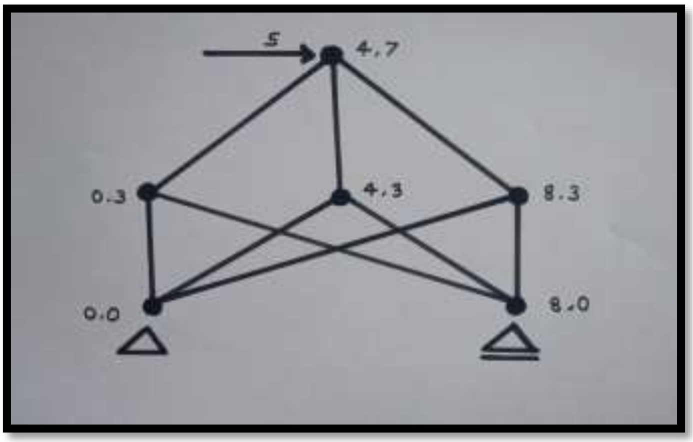

Figure 2 : Typical truss example 2

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 3, ISSN No. 2455-2143, Pages 230-235
Published Online July 2019 in IJEAST (http://www.ijeast.com)

II. METHODOLOGY

An algorithm is created with the help of MATLAB image processing and SAP 2000 API integration, such that as an input image of truss is passed through the algorithm and it will perform one by one operation on the image and segment components step by step and extract information. After collecting and assembling information algorithm will call SAP 2000 through API form MATLAB, send the information and make sap 2000 model and analysis the truss. The algorithm will retrieve back the results to MATLAB and close the SAP through API.

Steps involved in the method are

1) Binarizing image and removing text
2) Segmenting truss components
3) Detection and association of text
4) Validating Information.
5) Calling SAP 2000 through API.
6) Getting results back.

2.1. Binarizing image and removing text

The input image (Fig. 3) is converted to a binary image in (Fig. 4). To increase the accuracy of the detection of components of the truss, the objects with the lower areas (texts) are removed from the main image (Fig. 5).

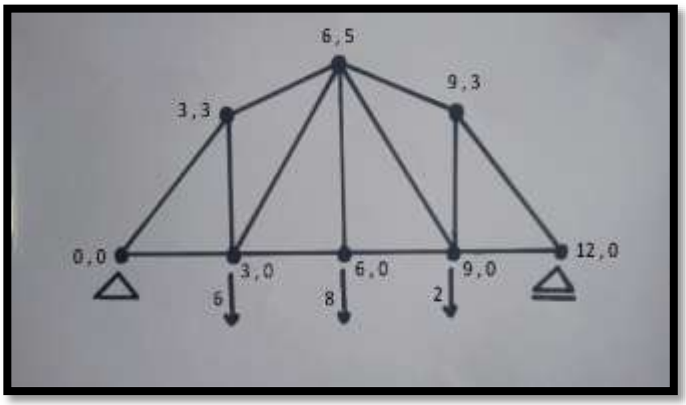

Figure 3 : Image of truss drawn on the piece of paper

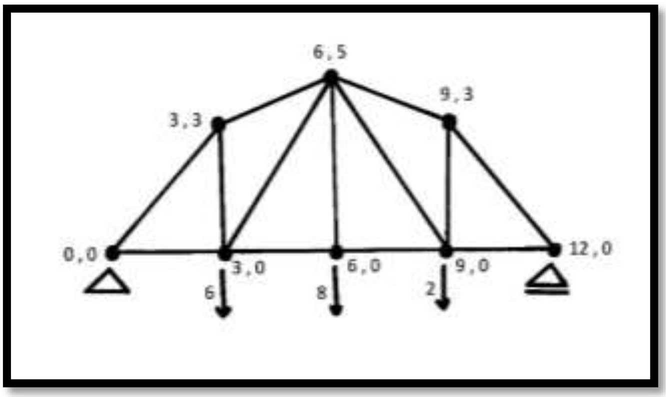

Figure 4 : After binarizing image shown in Fig. 3

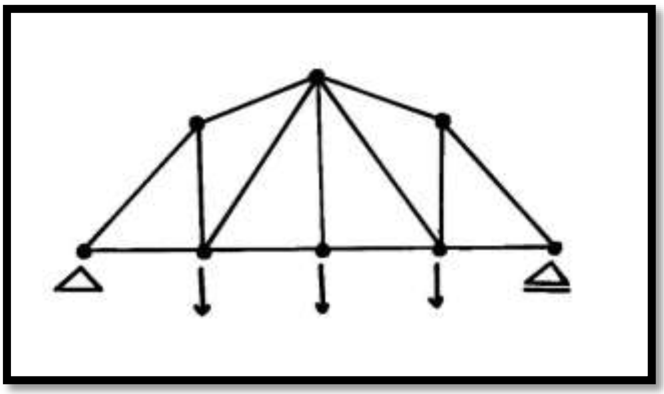

Figure 5 : After removing Text from image shown in Fig. 4

2.2. Segmenting truss components

The algorithm will segment nodes, member. load then supports from an input image in fixed and particular steps. And study all the segmented images to extract information accordingly. Segmentation is divined in the following steps:

2.2.1. Segmentation of the joint elements

The algorithm will try to determine the average diameter of nodes to define the size of the structuring element to perform erosion such that only nodes will survive. But the diameters of nodes are assumed to be uniform in size and large enough to survive erosion. The binary erosion of A by B denoted A ϴ B is defined as the set operation in equation 1.

$$A \ominus B = \{ z \mid B_z \subseteq A \} \qquad \text{... (1)}$$

In other words, it is the set of pixel locations z, where the structuring element translated to location z overlaps only with foreground pixels in A. It is assumed that the joint elements are large in diameter to survive the erosion operation. The binary gray-scale input image is eroded (Figure 6) to detect nodes. After eroding the image all the remaining objects, with lower boundary area, are considered to be the joint elements. The locations and the coordinates of the all detected joint elements are stored to check the availability of line between nodes of the truss.

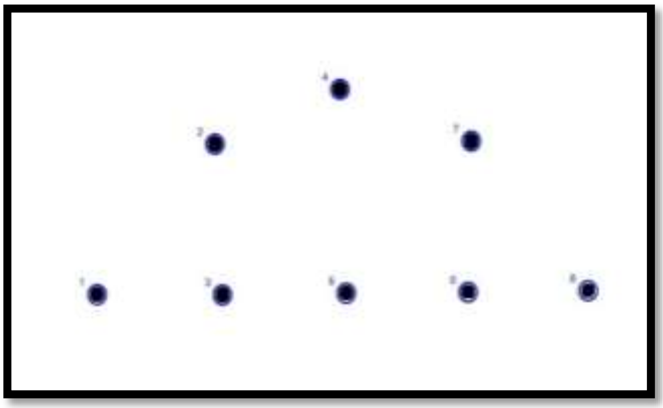

Figure 6 : Joint Segmentation

2.2.2. Segmentation of the line elements

After the node segmentation, the algorithm will store the centroids of the nodes. And between each pair of two nodes pixel values will be found to check the existence of line between them. The algorithm will only detect straight lines between nodes, curved lines will not be detected. Suppose 2 nodes are coming in one straight line, it will detect 3 lines 1st line 1-2, 2nd line 2-3 and 3rd line 1-3 by pixel values criteria between them but it is capable of detecting and detecting lines like 1-3 which is happened just because of misinterpretation of the algorithm. After the final detection of lines, a mask image of detected lines and nodes will be subtracted from the binary image to do the further segmentations. Line detection is shown below in Fig. 7.

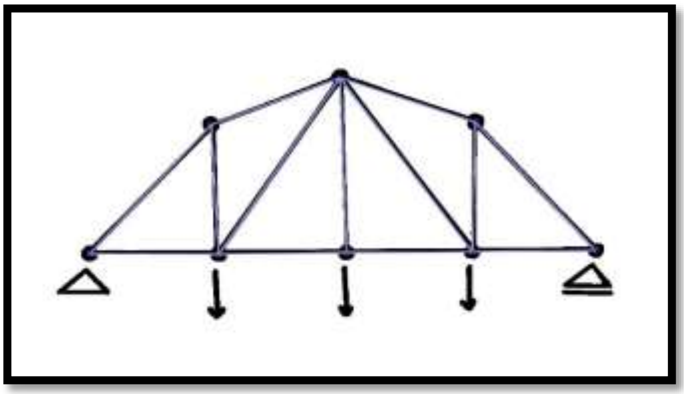

Figure 7 : Line detection

2.2.3. Segmentation of the arrows

After detection of the joint and the line elements and their elimination from the image in Fig. 7, arrows are detected in the binary image (Fig. 8). Morphological properties of the arrows can be explained as the line-shaped objects with their centroid shifted to their either ends. The algorithm will study the remaining boundary's centroids and also, it will assess the boundaries' similarity to a straight line. Thus, the following two criteria will have to be met to consider a boundary as an arrow:

- The line similarity value of the object will have to be greater than 0.95. This criterion will select all the objects similar to a line object.
- The centroid shift value calculated from (2) will have to be greater than 0.01. This criterion will select all the boundaries with their centroid shifted from their bounding box center.

$$centorid\ shift = \frac{\sqrt{(Bx-Cx)^2+(By-Cy)^2}}{\sqrt{(LBx^2-LBy^2)}} \dots (2)$$

Where, $(Bx,\ By)$ are the coordinates of bounding box Centre, $(Cx,\ Cy)$ are the centroid coordinates of the boundary and LBx, LBy are the length and width of bounding box.

Figure 8 : Arrow detection

2.2.4. Segmentation of the supports

All the supports are assumed to have a triangular shape. The algorithm is capable of detecting the pinned and the roller supports. To specify the direction of the degree of freedom in the roller supports, a line parallel to the lateral side of the triangle is assumed to be drawn. First, to segment the supports, the algorithm detects all the triangles. Then it separates the pinned supports from the roller ones and associates the direction of the degree of freedom to the roller supports. The bounding box, centroid and the area of each boundary are studied to detect the support objects.

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 3, ISSN No. 2455-2143, Pages 230-235
Published Online July 2019 in IJEAST (http://www.ijeast.com)

Figure 9 : Support detection

2.2.5. The segmentation results

Each segmented image is stored, and studied to collect information. Then the information is assembled and send to sap 2000 through API.

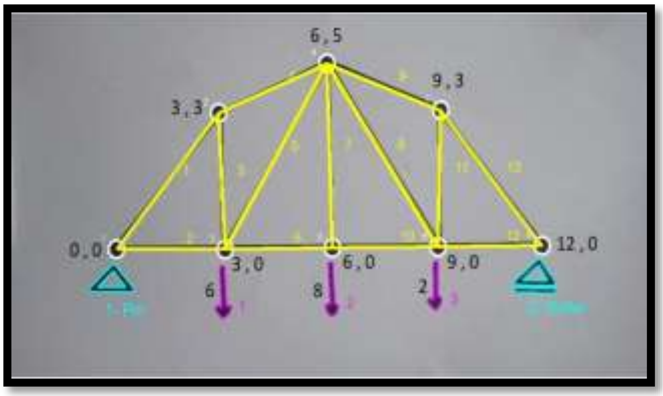

Figure 10 : Final overlay

2.3. Detection and association of text

After the segmentation process, the lower boundary area objects (text) which were removed from the binarized image is stored in a different image and sent for the OCR process. After the OCR recognition algorithm will associate coordinates to their appropriate nodes and magnitude of loads to their appropriate arrow.

2.4. Validating Information

Any information that is missing or has been assigned wrongly to any component is spotted before sending them to Sap2000 for analysis. The algorithm will ask the user to provide the correct and proper information. The algorithm proceeds to the next steps only if all the errors are fixed by the user.

2.5. Calling SAP 2000 through API and getting results

An automated interaction between MATLAB and Sap2000 though API ‘Application Programming Interface’ has been used to utilize Sap2000’s convenient analysis capability from MATLAB. Similar studies have been conducted to establish an integration between MATLAB and Sap2000. The geometry and loading information of the truss are sent from MATLAB to Sap2000 to provide the structural analysis, and then, the analysis results are retrieved back in MATLAB. The geometry information contains the coordinates, support type, and coordinates of the nodes and members location. The loading information includes the load magnitude and its angle of orientation.

III. RESULTS AND DISCUSSION

Output results will be in two tables one will be of Support reactions and others will be of member forces. (Fig. 9).

- Support reaction table –
  1st Colum - Contains a serial number
  2nd Colum - Contains node location on with the support is attached.
  3rd Colum - Contains Type of supports
  If Support type = 1, means Support is Pin Support.
  If Support type = 2, means Support is Roller Support.
  4th, 5th, and 6th columns are the reaction in the x-direction, y-direction, and z-direction respect. in KN.

- Member force table –
  1st Colum – Contains label of the truss member.
  2nd Colum – Start node of the member.
  3rd Colum – End node of the member.
  4th Colum – Force in the member in KN

International Journal of Engineering Applied Sciences and Technology, 2019
Vol. 4, Issue 3, ISSN No. 2455-2143, Pages 230-235
Published Online July 2019 in IJEAST (http://www.ijeast.com)

```
Support_type = 1 is Pin Support
Support_type = 2 is Roller Support
Support reactions are as below

Support_reaction =

  2x6 table

    S_No    Support_location    Support_type    Rx_KN    Ry_KN    Rz_KN
    ____    ________________    ____________    _____    _____    _____

     1             1                 1            0        0        9
     2             8                 2            0        0        7

Truss member forces are as below

member_force =

  13x4 table

    Member_No    StartNode    EndNode    Force_KN
    _________    _________    _______    ________

        1            1           2        -12.728
        2            1           3              9
        3            2           3              3
        4            2           4        -10.817
        5            3           4          3.499
        6            3           5            7.2
        7            4           5              8
        8            4           6         -0.389
        9            4           7         -8.413
       10            5           6            7.2
       11            6           7          2.333
       12            6           8              7
       13            7           8         -9.899
```

Figure 11 : Results

IV. CONCLUSION

The main focus of this paper is to find a method to automate analysis of 2-D trusses. With the help of morphological image processing of MATLAB an input image of drawn truss will be segmented step by step to extract geometry of the truss from the image, and SAP 2000 API codes are integrated to the algorithm to get the solution of the truss drawn. To get the exact geometry of the truss from the image, we have used nodal coordinates to describe the geometry of the truss on the paper. The current work segments and analyses the truss using an image as an input. This paper not only shows the automated truss analysis but also the potential of the image processing. The future scope is to extend the algorithm to analysis of 2-D frames and continuous beams.

V. ACKNOWLWDGEMENT

I take this opportunity to express my deep sense of gratitude towards my guide Prof. J. P. Patankar, Professor, Applied Mechanics Department, Government College of Engineering, Karad. I extent my sincere thanks to Dr. A. T. Pise, Principal, Government College of Engineering, Karad and Dr. Y. M. Ghugal, Head, Applied Mechanics Department, Government College of Engineering, Karad for providing institutional facilities and extending all kinds of co-operation. I am thankful to all the faculty members of Applied Mechanics, Civil Engineering Department and Library whose guidance and help have been immensely useful in my work.

VI. REFERENCE

[1] McAndrew, A., et al. (2004), "An introduction to digital image processing with Matlab notes for scm2511 image processing. School of Computer Science and Mathematics", Victoria University of Technology, p. 1-264.

[2] Adam, S., et al. (2000), "Symbol and character recognition: application to engineering drawings" International Journal on Document Analysis and Recognition, p. 89-101.

[3] Deseilligny, M.P., et al. (1995), "Character string recognition on maps, a rotation-invariant recognition method", Pattern Recogn. Lett., p. 1297-1310.

[4] Chiang, Y.-Y. and C. Knoblock., et al (2011), "Recognition of multi-oriented, multi-sized, and curved text", in Document Analysis and Recognition (ICDAR), 2011 International Conference on. 2011. IEEE.

[5] Jones, G.P., et. al. (2013), "Interoperable software for parametric structural analysis and optimization", Massachusetts Institute of Technology.

[6] Goo, Z.J., et. al. (2013), "Topology optimization of building bracing schemes", Massachusetts Institute of Technology.

[7] Gomez, D., et al. (2015), "Interactive Platform to Include Human-Structure Interaction Effects in the Analysis of Footbridges" in Dynamics of Civil Structures, Volume 2, J. Caicedo and S. Pakzad, Editors., Springer International Publishing. p. 59-65.

[8] Kamari Mirsalar, Oğuz Güneş, et al. (July 2016), "Segmentation and Analysis of a Sketched Truss Frame Using Morphological Image Processing Techniques", International Conference on Civil Engineering, Architecture and Cityscape.

[9] U.Scot.E, et. al. (1998), "Computer Vision and Image Processing", Prentice Hall, NJ, ISBN 0-13-264599-8.

[10] J. Gil and R. Kimmel, et. al. (December 2002), "Efficient dilation, erosion, opening, and closing algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 12, pp. 1606.1617

[11] G.V.Tcheslavski, et. al. (2010), "Morphological Image Processing: Grayscale morphology", ELEN 4304/5365 DIP, Spring.

[12] Nixon Mark, et. al. (January 2008), "Feature Extraction & Image Processing", Academic Press, ISBN 9780080556727.