Musical Note Recognition Using Minimum Spanning Tree Algorithm

Yoppy Sazaki, MT
Faculty of Computer Science
Sriwijaya University
Palembang, Indonesia
yoppysazaki@gmail.com

Rosda Ayuni, S.Kom
Faculty of Computer Science
Sriwijaya University
Palembang, Indonesia
rosda.ayuni92@gmail.com

Abstract**—*****Musical Notes are notes which is placed in staff. This research was developed a musical note recognition software using Minimum Spanning Tree Algorithm. This software was developed to help beginner in learning music especially in recognizing musical notes. The input for this software was musical notes image and the output were information of musical note which is name of musical note and beat's length sound of recognized musical note. There were four pre-processing involved in this research namely Sobel edge detection, binarization, segmentation and scaling then the result from pre-processing was used in training process. Accuracy of musical note recognition using this algorithm reached 97.9 per cent out of 97 trained data and 97.4 per cent out of 40 tested data.***

Keywords—Musical Note, Musical Note Recognition, Minimum Spanning Tree Algorithm, Sound.

I. INTRODUCTION

Human could know the value of musical note's beat relative ease by seeing the shape of it. On the other hand, computer needed more process to recognize musical note. Related research about musical note had been developed. Methods that being used such as Artificial Neural Network [1] and Mask Matching [2]. Accuracy by using Artificial Neural Network method is 84.1 per cent until 96.8 per cent whereas the accuracy by using Mask Matching method is 99.68 per cent.
Musical note recognition using Minimum Spanning Tree Algorithm is rarely found so that the research to recognize musical note with Minimum Spanning Tree Algorithm is done.

II. RESEARCH METHODOLOGY

This research is using Minimum Spanning Tree Algorithm and Euclidian distance to count the pixel length. Beforehand, musical note will be put through four pre-process steps, which are, edge detection, binarization, segmentation, and scaling. After the scaling process, the pixel length will be counted using the Euclidian distance. When the minimum length of each pixel has been found, the calculation using Minimum Spanning Tree Algorithm could be done.

The purpose of this research are:

1. Implementing the Minimum Spanning Tree method to recognize musical note onto the software.
2. To find out the level of accuracy of the software.

The problem formulation of this research are:

1. To help the musical art lesson for beginner who wants to learn about music, especially to recognize musical note.
2. To give information about musical note from the software.

In order to be focused on the research, the limitations of this research are:

1. Input in form of musical note without staff from one to seven musical note.
2. The musical note used as input is on 6 form, which are, quarter note, half note, eighth note, sixteenth note, and thirty-second note.
3. The programming language used in this software is C++.

III. RESULT AND ANALYSIS

The system being built on this research is algorithm MST application to recognize musical note. This application will recognize musical note based on the form and produce a long sound from the recognized musical note. 2. The musical note used in this research is on 6 form, which are, quarter note, half note, eighth note, sixteenth note, and thirty-second note.

978-1-4799-7447-4/14/$31.00 ©2014 IEEE

The system was built using Unified Modelling Language (UML). This system has the ability to shows image, training and recognizing.

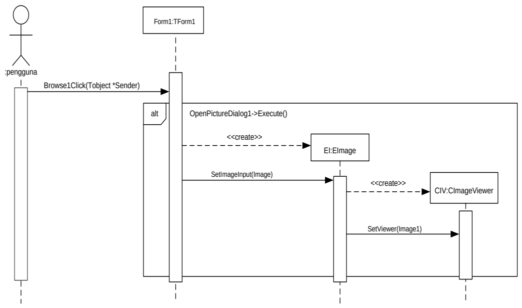

Fig. 1. The Sequence Diagram Showing Image

The Sequence Diagram above is portraying the process of showing image on system. The image will then be processed to be recognized.

First, user choose the Training menu after the image shows. The process in this training contain pre-process and weight calculation between pixels. The pre-process steps are, edge detection, binarization, segmentation, and scaling. After the pre-process on image is done, the weight between pixel will be calculated. The purpose of this total weight calculation using MST is to find the total weight which will be used on the recognizing process as the determined variable of whether the musical note is recognized or not by the software. Before calculating the total weight of a certain musical note, the search of the minimum length of each pixel dot on the musical note image. The length of each pixel is counted with Euclidian Distance.

The input image has two colors, black (as background) and white (as musical note), if a found pixel value is 255 (white), then the length calculation from one white pixel to another will continue until all the white pixel detected in an image is counted.

One white pixel will have some length from the length calculation before. The minimum length from a white pixel could be find by the sequence process. The shortest length from the sequence process is the length from the first array, not from the zero array. It is because the length calculation from one white pixel, the coordinate point is included on the white pixel coordinates in a musical note image. If the length used for the sequence process is from the zero array, then the value of the array will always be zero on each minimum length counted from the white pixel. This will influenced to the total weight of the musical note.

The result of all the white pixel minimum length from a musical note will be sums up and used as total weight from the musical note. The total weight of each musical note will be used in the recognition process to classified the musical note.

Written below is the Euclidean Distance to count the length between pixels:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

where :

d = length between node

i = index (0,1,...,n)

x_1, x_2 = horizontal coordinate of two point which will be counted

y_1, y_2 = vertical coordinate of two point which will be counted

MST is a tree with the most minimum length. To find the total weight, the formula is:

$$L = \sum_{e_{i,j} \in G} \| e_{i,j} \|$$

where:

L = MST total length

$e_{i,j}$ = connector between nodes (*edges*)

G = MST

Musical note recognition was done after total weight of musical note obtained. Total weight used in classification process accordance with specified condition in software. The condition is obtained from the result of total weight count of 97 trained data.

Result of total weight count of 97 trained data is different accordance with the musical note. One musical note could have a different total weight, so range is given for each musical note. If a musical note total weight did not include in one of given condition then that musical note is not recognized by the software.

TABLE 1. Total Weight Range

No.	Musical Note	Total Weight Range
1.	Whole Note	132 – 179
2.	Half Note	288 – 323
3.	Quarter Note	274 - 282 and 324 – 413
4.	Eighth Note	416 – 510
5.	Sixteenth Note	513 – 552
6.	Thirty-second Note	571 – 644

Final result of recognizing process is the information regarding to musical note namely musical note name and beat's length sound of recognized musical note.

Software Testing

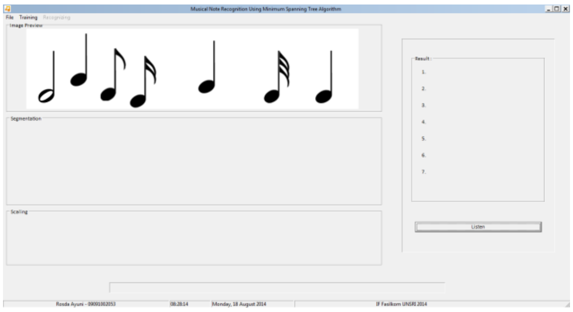

Fig. 2. Testing of Showing Image

In showing image process, image shown at the system after the user choose File menu – Open then OK.

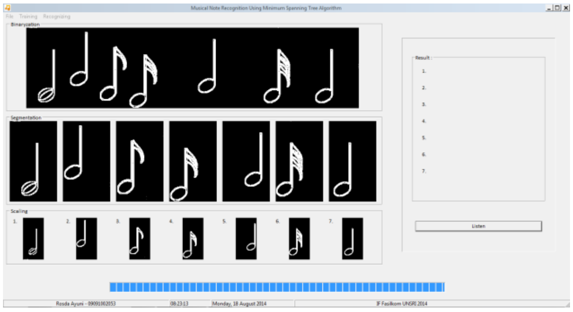

Fig. 3. Testing of Training

System response which is showed on training process is when user choose training menu then pre-process and training is run. Pre-process result, edge detection and binarization, is shown in first row. Segmentation result is in second row and scaling result is in third row.

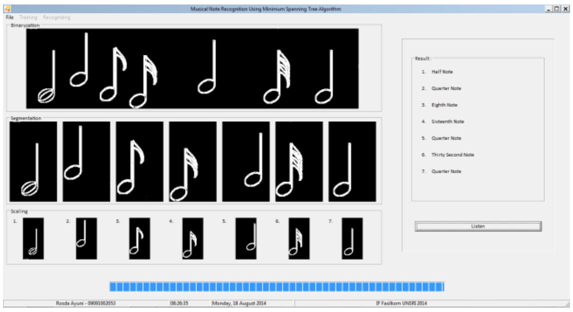

Fig. 4. Testing of Recognizing

System response which is showed on recognizing process is when user choose recognizing menu then recognition process is run. The result of recognition process is musical note name and beat's length sound of recognized musical note. Beat's length sound of recognized musical note can be heard by clicking listen button.

The calculation of accuracy recognition is done with this formula:

$$\frac{Reognized\ Data}{Whole\ Data} \quad X \quad 100\%$$

Testing is done by using digital image which contain musical note. Trained data that is used is 97 data and tested data that is used is 40 data. The level of accuracy from the testing using trained data is 97.9 per cent, meanwhile the level of accuracy from the testing using tested data is 97.4 per cent.

TABLE 2. Table of Trained Data Testing

No.	**File Name**	**Musical Note Amount**	**Recog-nized Amount**	**Unrecog-nized Musical Note Amount**
1.	Data-1.bmp	7	7	0
2.	Data-2.bmp	7	7	0
3.	Data-3.bmp	7	7	0
4.	Data-4.bmp	7	7	0
5.	Data-5.bmp	7	7	0
6.	Data-6.bmp	7	7	0
7.	Data-7.bmp	7	7	0
8.	Data-8.bmp	7	7	0
9.	Data-9.bmp	7	7	0
10.	Data-10.bmp	7	7	0
11.	Data-11.bmp	7	7	0
12.	Data-12.bmp	7	7	0
13.	Data-13.bmp	7	7	0
14.	Data-14.bmp	7	7	0
15.	Data-15.bmp	7	7	0
16.	Data-16.bmp	7	7	0
17.	Data-17.bmp	7	7	0
18.	Data-18.bmp	7	7	0
19.	Data-19.bmp	7	7	0
20.	Data-20.bmp	7	7	0
21.	Data-21.bmp	7	7	0
22.	Data-22.bmp	7	7	0
23.	Data-23.bmp	7	7	0
24.	Data-24.bmp	7	6	1
25.	Data-25.bmp	7	7	0
26.	Data-26.bmp	7	7	0
27.	Data-27.bmp	7	6	1
28.	Data-28.bmp	7	7	0
29.	Data-29.bmp	7	7	0
30.	Data-30.bmp	7	7	0
31.	Data-31.bmp	7	7	0
32.	Data-32.bmp	7	7	0
33.	Data-33.bmp	7	6	1
34.	Data-34.bmp	7	6	1
35.	Data-35.bmp	7	7	0
36.	Data-36.bmp	7	6	1
37.	Data-37.bmp	7	7	0
38.	Data-38.bmp	7	7	0
39.	Data-39.bmp	7	7	0
40.	Data-40.bmp	7	7	0
41.	Data-41.bmp	7	6	1
42.	Data-42.bmp	7	7	0
43.	Data-43.bmp	7	7	0
44.	Data-44.bmp	7	6	1
45.	Data-45.bmp	7	7	0
46.	Data-46.bmp	7	7	0
47.	Data-47.bmp	7	7	0
48.	Data-48.bmp	7	7	0
49.	Data-49.bmp	7	7	0
50.	Data-50.bmp	7	7	0
51.	Data-51.bmp	7	7	0
52.	Data-52.bmp	7	7	0

53.	Data-53.bmp	7	7	0
54.	Data-54.bmp	7	7	0
55.	Data-55.bmp	7	7	0
56.	Data-56.bmp	7	6	1
57.	Data-57.bmp	7	6	1
58.	Data-58.bmp	7	6	1
59.	Data-59.bmp	7	7	0
60.	Data-60.bmp	7	7	0
61.	Data-61.bmp	7	7	0
62.	Data-62.bmp	7	6	1
63.	Data-63.bmp	7	7	0
64.	Data-64.bmp	7	5	2
65.	Data-65.bmp	7	7	0
66.	Data-66.bmp	7	7	0
67.	Data-67.bmp	7	7	0
68.	Data-68.bmp	7	7	0
69.	Data-69.bmp	7	7	0
70.	Data-70.bmp	7	7	0
71.	Data-71.bmp	7	7	0
72.	Data-72.bmp	7	7	0
73.	Data-73.bmp	7	7	0
74.	Data-74.bmp	7	7	0
75.	Data-75.bmp	7	7	0
76.	Data-76.bmp	7	7	0
77.	Data-77.bmp	7	7	0
78.	Data-78.bmp	7	7	0
79.	Data-79.bmp	7	7	0
80.	Data-80.bmp	7	7	0
81.	Data-81.bmp	7	7	0
82.	Data-82.bmp	7	7	0
83.	Data-83.bmp	7	7	0
84.	Data-84.bmp	7	7	0
85.	Data-85.bmp	7	7	0
86.	Data-86.bmp	7	7	0
87.	Data-87.bmp	7	7	0
88.	Data-88.bmp	7	7	0
89.	Data-89.bmp	7	7	0
90.	Data-90.bmp	7	7	0
91.	Data-91.bmp	7	7	0
92.	Data-92.bmp	7	6	1
93.	Data-93.bmp	7	7	0
94.	Data-94.bmp	7	7	0
95.	Data-95.bmp	7	7	0
96.	Data-96.bmp	7	7	0
97.	Data-97.bmp	7	7	0
Total		679	665	
Accuracy Percentage		**97.9 per cent**		

TABLE 3. Table of Tested Data Testing

No.	File Name	Musical Note Amount	Recog-nized Amount	Unrecog-nized Musical Note Amount
1.	Test Data-1.bmp	2	2	0
2.	Test Data-2.bmp	2	2	0
3.	Test Data-3.bmp	6	6	0
4.	Test Data-4.bmp	3	3	0
5.	Test Data-5.bmp	4	4	0
6.	Test Data-6.bmp	5	5	0
7.	Test Data-7.bmp	1	1	0
8.	Test Data-8.bmp	2	2	0
9.	Test Data-9.bmp	1	1	0
10.	Test Data-10.bmp	3	3	0
11.	Test Data-11.bmp	4	4	0
12.	Test Data-12.bmp	4	3	1
13.	Test Data-13.bmp	1	1	0
14.	Test Data-14.bmp	6	6	0
15.	Test Data-15.bmp	6	6	0
16.	Test Data -16.bmp	3	3	0
17.	Test Data -17.bmp	5	4	1
18.	Test Data -18.bmp	5	5	0
19.	Test Data -19.bmp	7	7	0
20.	Test Data -20.bmp	7	7	0
21.	Test Data -21.bmp	1	1	0
22.	Test Data -22.bmp	2	2	0
23.	Test Data -23.bmp	3	3	0
24.	Test Data -24.bmp	4	3	1
25.	Test Data -25.bmp	5	5	0
26.	Test Data -26.bmp	6	6	0
27.	Test Data -27.bmp	7	7	0
28.	Test Data -28.bmp	5	5	0
29.	Test Data -29.bmp	4	4	0
30.	Test Data -30.bmp	2	2	0
31.	Test Data -31.bmp	3	3	0
32.	Test Data -32.bmp	7	7	0
33.	Test Data -33.bmp	2	2	0
34.	Test Data -34.bmp	6	6	0
35.	Test Data -35.bmp	3	3	0
36.	Test Data -36.bmp	2	2	0
37.	Test Data -37.bmp	6	5	1
38.	Test Data -38.bmp	1	1	0
39.	Test Data -39.bmp	4	4	0
40.	Test Data -40.bmp	5	5	0
Total		155	151	
Accuracy Percentage		**97.4 per cent**		

On research, there are two kind of mistake in recognizing musical note. The first mistake is the musical note is really unrecognizeable by the software and shows the result “NOT RECOGNIZED” on the form. This is caused by the total weight of the musical note is not in the range of any total weight recognizeable by the software. The total weight of each musical note can be seen on Table 1.

This mistake has been overcome by widening the total weight value range for the unrecognizeable musical note. Hence the probability of a musical note to be unrecognizeable is not happen.

The second mistake is the mistake in recognizing musical note. This mistake enable a musical note could be recognized by the software but not classified. This thing caused by the total weight of the musical note is misrecognized into the range of other musical note total weight range. This mistake could happen because the data was created in different way. The image of musical note looks the same by the naked eyes, but on the pre-process edge detection and binarization, the result is different. This is because the shape precision of the musical note produced is different because the creating way is also different. The other thing caused mistake on recognizing is the image of the musical note was cut at some parts, thus the pixel in the musical note image is decreased, influencing the total weight of the musical note.

For example, the sixteenth note on data 64.bmp is misrecognized to eighth note. The sixteenth note on the data has a total weight of 510. Meanwhile the range of total weight for sixteenth note is from 513 to 552 and the total weight of 510 is included on the value range of eighth note, which is 416 to 510. This is causing the sixteenth note on data 64.bmp is misrecognized. Meanwhile the sixteenth note on data 65.bmp has the total weight of 529 which means the musical note is recognized by the software.

The misrecognized can not be solved by changing the musical note into a similar musical note which has the right total weight and could be classified as the right musical note. Therefore, the misrecognized could be solved by adding pre-process step to get the more specific characteristic from the image. The image of musical note could be put into another segmentation to check the head part followed by the flag part. So the head and flag part has different total weight.

IV. CONCLUSION

The conclusion of this research are :

1. Minimum Spanning Tree Alghorithm could be used in recognizing musical note.
2. From 97 trained data, the accuration of the recognition is 97.9 per cent. Meanwhile, the tested data accuration of recognition is 97.4 per cent.

REFERENCES

[1] Youssef, K. and P. Y. Woo. (2008). *Music Note Recognition Based On Neural Network. Fourth International Conference on Natural Computation*. IEEE.

[2] Yoo, J. M., G. H. Kim, G. S. Lee. (2008). *Mask Matching for Low Resolution Musical Note Recognition.* IEEE.

[3] Crawfis, R. (2011). *Introduction to Algorithms Spanning Tree*. Retrieved from Ohio State Buckeyes. http://www.cse.ohio-state.edu/~crawfis/cse680/Slides/CSE680-15SpanningTrees.pdf.

[4] Franco, P., J. M.Ogier, P. Loonis, R. Mullot. (2010). *A New Minimum Trees-Based Approach for Shape Matching with Improved Time Computing: Application to Graphical Symbols Recognition*. Springer.

[5] Franco, P., J. M.Ogier, P. Loonis, R. Mullot. (2004). *A Topological Measure for Image Object Recognition*. Springer.

[6] Kruchten, P. (2003). *The Rational Unified Process An Introduction Third Edition*. Addison Wesley.

[7] Kleinberg, J. and É. Tardos. (2005). *Algorithm Design : Chapter 4 Greedy Algorithm*. Pearson – Addison Wesley http://www.cs.princeton.edu/~wayne/kleinberg-tardos/04MinimumSpanningTrees.pdf.

[8] Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development, Third Edition*. Addison Wesley Professional.

[9] Ma, B., A. Hero, J. Gorman, O. Michel. (2000). Image Registration with Minimum Spanning Tree Algorithm. IEEE.

[10] Munir, R. (2009). *Matematika Diskrit Edisi Ketiga.* Bandung. Informatika.

[11] Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta. ANDI.

[12] Rahardiantoro, D. (2008). *Minimal Spanning Tree*. http://dickyrahardi.blogspot.com/2008/05/minimal-spanning-tree.html.

[13] Solich, Warsono, F.X. Sadono, Dedeh, Sunardi. (2007). *Seni Budaya dan Keterampilan*. Erlangga. Jakarta.

[14] Sutoyo, T., E. Mulyanto, V. Suhartono, O.D. Nurhayati, Wijanarto. (2009). *Teori Pengolahan Citra Digital*. ANDI. Yogyakarta.