Colorscore - Visualization and Condensation of Structure of Classical Music

Aki Hayashi, Takayuki Itoh
Graduate School of Humanities and Sciences
Ochanomizu University
Tokyo, Japan
{*aki, itot*}*@itolab.is.ocha.ac.jp*

Masaki Matsubara
Graduate School of Science and Technology
Keio University
Kanagawa, Japan
masaki@nak.ics.keio.ac.jp

***Abstract*—It is not always easy to quickly understand musical structure of orchestral scores for classical music works, because these works contain many staves of instruments. This paper presents Colorscore, a technique for visualization and condensation of musical scores. Colorscore supports two requirements for composers, arrangers and players: overview and arrangement. Colorscore divides each track of the score into note-blocks, and determines their roles. Colorscore then displays all the note-blocks in one display space to provide the overview, so that novice people can quickly understand the musical structures. In addition, Colorscore supports vertical condensation which reduces the number of displayed tracks, and horizontal condensation which saves the display space. It is especially useful as hints to rearrange music for smaller bands.**

***Keywords*-Visualization; Condensation; Classical music;**

I. Introduction

Listening to the music necessarily takes time. Visualization of music information is very useful and powerful if we would like to understand the contents and story of the music in a short time. Many recent techniques on music visualization focus on representation of musical structures of musical works. We are focusing on visualization of structures of musical works composed for orchestras. Generally, it is not easy for beginner conductors, players, and composers, to read and/or write orchestral scores. We think one of the main reasons is that such scores line up many staves of instruments vertically in one page; therefore, readers must look at the many staves of instruments at one time, and turn over the pages frequently. In addition, it may be difficult for them to observe the overall structures and contexts of the musical works from the scores containing large number of pages. We think visualization is useful to solve the above problem; because it has the potential to provide overview of musical information, and interactive mechanisms to observe important parts of the works.

This paper presents "Colorscore", a technique for visualization and condensation of musical structures of MIDI-based classical music data. Colorscore consists of the following components: role determination, overview, and interactive condensation. Colorscore first divides notes of each track into blocks based on the rests of the tracks. It then determines the roles of blocks, by matching to predefined patterns of melodies and accompaniments. Colorscore draws all the blocks into single display space, for the overview of musical structures, while it assigns independent colors to the blocks based on their roles. It also provides two mechanisms of interactive condensation to save the display spaces. One of the mechanisms is "vertical condensation", which reduces the number of displayed tracks, by remaining only note-blocks well matches to the predefined patterns. The other mechanism is "horizontal condensation", which shrinks the bars if no roles are changed at that time. We suppose Colorscore is especially useful for beginner or amateur players, composers, arrangers, and conductors. Score reading may be a hard task for junior high school or high school students who aim to enter the music departments of universities. Also, there have been more than a thousand of amateur orchestras at least in authors' country, and again, it may be a hard task for amateur players and conductors. They may need a long time to understand the whole structure of music by reading a score. We expect the overview provided by Colorscore will help to quickly understand it, because we can observe it by looking at single display space.

Also, we think that condensation of Colorscore will help to rearrange music. There have been many musical works that are arranged by various styles of bands. For example, "Pictures at an Exhibition" was originally composed for the piano by Mussorgsky, then arranged for orchestra by Ravel, and recently played by various bands including brass quintet (Canadian Brass) and 3-piece rock band (Emerson, Lake and Palmer). We expect condensation of Colorscore may suggest hints to such arranges for smaller bands, because it displays the structures of music as smaller number of tracks.

II. RELATED WORK

There have been many techniques on visualization of musical structures or scores. Many of such works [1], [2], [3] aim visualization of acoustic data, not MIDI data. Since audio source separation is still a difficult problem, such works did not visualize musical structures track-by-track. Some of early note-based music visualization works [4] were also simple representations, not detailed track-by-track visualization. Colorscore is quite different from these works because it aims to visualize roles of note-blocks of each track of orchestra separately into one display.

On the other hand, several recent works focus on visualization of track-by-track information of musical score. comp-i [5] visualizes musical structures of MIDI data in virtual 3D space. However, comp-i does not analyze roles of note-blocks. ScoreIlluminator [6] improves the readability of musical scores by assigning colors to phrases of tracks based on their similarities. However, ScoreIlluminator does not take the roles of the phrases into account for assigning colors. Moreover, comp-i and ScoreIlluminator do not support condensation of musical score. BRASS [7] realizes accordion-like interactive visualization, by partially expanding or shrinking the musical score, which looks similar to condensation of Colorscore. However, it does not take roles of phrases into account for expanding or shrinking. Against such related works, Colorscore realizes colored visualization and condensation of musical scores based on roles of note-blocks.

III. TECHNICAL COMPONENTS OF COLORSCORE

This section presents procedures and technical components of Colorscore. As a preprocessing, Colorscore analyzes musical structures to divide each track into multiple note-blocks, and determines their roles and transitions. Then, Colorscore displays the roles of note-blocks in multiple colors. Also, Colorscore can interactively condense scores by vertically or horizontally packing the displaying information.

A. Analysis of musical structure

Colorscore supports SMF (Standard MIDI file) as input data. It requires note information of MIDI, including pitch, strength, duration and timing. Colorscore firstly read the SMF file, and divides each track of the score into note-blocks. Then, Colorscore determines roles of note-blocks by matching their notes to patterns given by users.

1) Providing the patterns to decide the role: First of all, Colorscore requires the patterns used to determine the roles of the note-blocks. In this paper, "pattern" means a short set of notes which consists of just one track in MIDI format. Currently we assume that patterns are given by expert users. Colorscore determines whether each block plays melodies or accompaniments, as a "role." As regard to melodies, Colorscore supposes to input basic phrases of several main melodies. At the same time, Colorscore supposes to input only typical rhythms for the accompanying phrases such as harmonic or bass accompaniments: it does not analyze transition of intervals for accompaniments. We consider the accompaniments are often characterized by repeated rhythm rather than by transition of interval, and therefore we designed Colorscore to input patterns of accompaniments as rhythms.

2) Generating the initial note-blocks: After providing user-given patterns, Colorscore generates rough note-blocks, called "initial note-blocks." Colorscore generates the initial note-blocks by the following:

1. Treat a track as a single block.
2. Scan a block, and divide it into two blocks at a whole note rest.
3. Repeat 2. for all blocks until all whole note rests are eliminated from the blocks.
4. Repeat 2. and 3. for all tracks.

3) Pattern-matching of the blocks with patterns: Colorscore matches each initial note-block (see Section III-A2) to user-given patterns (see Section III-A1). In this step, Colorscore calculates distances between the patterns and each note-block, and chooses the pattern closest to the note-block. It determines that the note-block has the role which is the same as the chosen pattern, if the distance between the note-block and the pattern is smaller than the predefined threshold. To calculate the distance between the i-th pattern and the j-th note-block, our implementation applies the following distance $D(i,j)$ [6] :

$$D(i,j) = w_1 D_{RA}(i,j) + w_2 D_{MA}(i,j) \qquad (1)$$

Here, w_1 and w_2 denote constant weights, $D_{RA}(i,j)$ is the cosine of timing which features the rhythm, and $D_{MA}(i,j)$ is the cosine of transition of the notes which features the melody. $D_{RA}(i,j)$ corresponds to the cosine of RA vectors between the i-th pattern and the j-th note-block. Here, RA vector is an n-dimensional vector denoting the timing of note-on events of note-blocks or patterns. To generate RA vector, Colorscore divides note-blocks into n pieces by the constant note. It then assigns positive number if there is the note-on event, otherwise assigns "0". Consequently, Colorscore generates the RA vector as an n-dimensional binary vector. $D_{MA}(i,j)$ corresponds to the cosine of MA vectors between the i-th pattern and the j-th note-block. Here, MA vector is an $(n-1)$-dimensional vector denoting the pitch transition of note-on events. To generate MA vector, Colorscore calculates the difference of pitches between k-th and $(k+1)$-th pieces as the k-th element value of the vector. Here, it takes a positive value if the pitch gets higher, otherwise a negative value. Figure 1(Left) shows an example of RA and MA vectors. Here, the bars in Figure 1(Left) are divided into nine pieces by quarter notes. Values of RA vector take 3 at the first beat, and 2 at the third beat. The MA value is calculated based on chromatic distances between two tones.

If the length of note-block is longer than that of pattern, it may often happens that the note-block partially matches to the pattern. Considering such situation, Colorscore first extracts parts of the note-block, where the lengths of the parts are equal to the pattern, and calculate $D(i,j)$ applying each part. If one of the parts matches to the pattern, Colorscore divides the note-blocks into two or three note-blocks, where one of them corresponds to the part matches

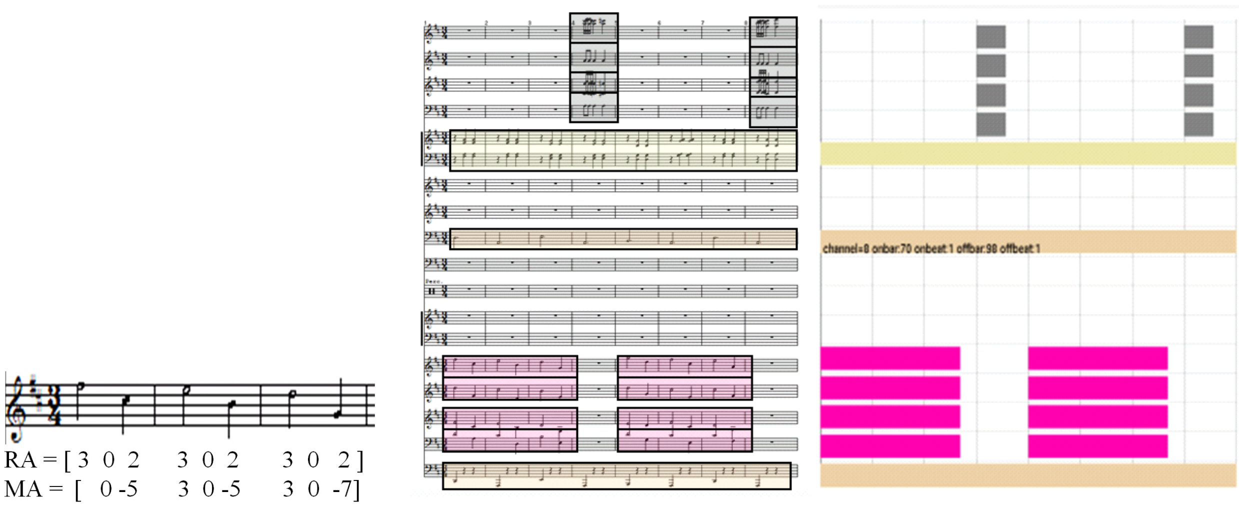

Figure 1. (Left) RA and MA vectors. (Center) Result of note-block generation and role determination. Pink blocks denote the main melody, yellow and flesh blocks denote accompaniments, and gray blocks denote decorations. (Right) Visualization by Colorscore.

to the pattern. Then, Colorscore applies the same process to the remaining note-blocks.

Colorscore applies the above-mentioned process to every note-block of every track. Here, Colorscore determines that a part of the j-th note-block matches to the i-th pattern, if the $D(i,j)$ is smaller than the predefined threshold D_0, where D_0 is a function of n.

The procedure to assign roles is as follows:

1. Calculate RA and MA vectors of the i-th pattern.
2. Calculate RA and MA vectors of the j-th note-block.
3. Extract a part of the note-block, where the length of the part is equal to the i-th pattern, and then calculate $D(i,j)$ between the part and the i-th pattern.
4. If $D(i,j)$ is smaller than D_0:
   a) Divide the j-th note-block if necessary.
   b) Assign the role of the i-th pattern to the matched note-block.
5. Repeat 3. and 4. for all possible parts of the j-th note-block.
6. Repeat 2. to 5. for all note-blocks.
7. Repeat 1. to 6. for all patterns.

B. *Visualization of note-blocks*

Colorscore visualizes the result of note-block generation and role determination. Figure 1(Center) shows the result of the analysis drawn on a traditional musical score, and Figure 1(Right) shows the result of visualization by Colorscore. It vertically draws the tracks, and horizontally draws the blocks in each track. It assigns colors to the note-blocks based on their roles. Our implementation assigns high-saturation colors to melodies, low-saturation colors to accompaniments, and gray to unmatched note-blocks.

C. *Vertical condensation*

Vertical condensation consists of the following two steps. The first step removes decoration note-blocks, then removes tracks which have no note-blocks to be drawn, and finally vertically packs the remaining note-blocks. The second step removes more note-blocks so that only the note-blocks especially similar to the given patterns remain in the visualization results. Our implementation remains only note-blocks whose $D(i,j)$ values are smaller than a predefined threshold D_1 ($D_1 < D_0$). This step eliminates harmonic melodies, and remains main themes. Above process is repeated until the tracks are reduced to the user-specified number. We think this functionality is especially useful while arranging orchestra music into smaller organization such as piano solo or chamber ensemble.

D. *Horizontal condensation*

Colorscore horizontally saves the display space based on transition of roles. It shrinks bars if no note-blocks end or change their roles, while it keeps other bars longer. For example, for Figure 1(Right) Colorscore shrinks the second and third bars because no note-blocks change their roles at that time. Also, it keeps the fourth bars longer, because new note-blocks start at that time.

IV. RESULTS

This section introduces examples of our visualization results applying the MIDI data of ”Valse des fleurs” composed by Tchaikovsky.

A. *Visualization*

Figure 2(Upper) shows an example of visualization result of whole MIDI data which contains 16 tracks. Colorscore represents the musical structure in a single display space,

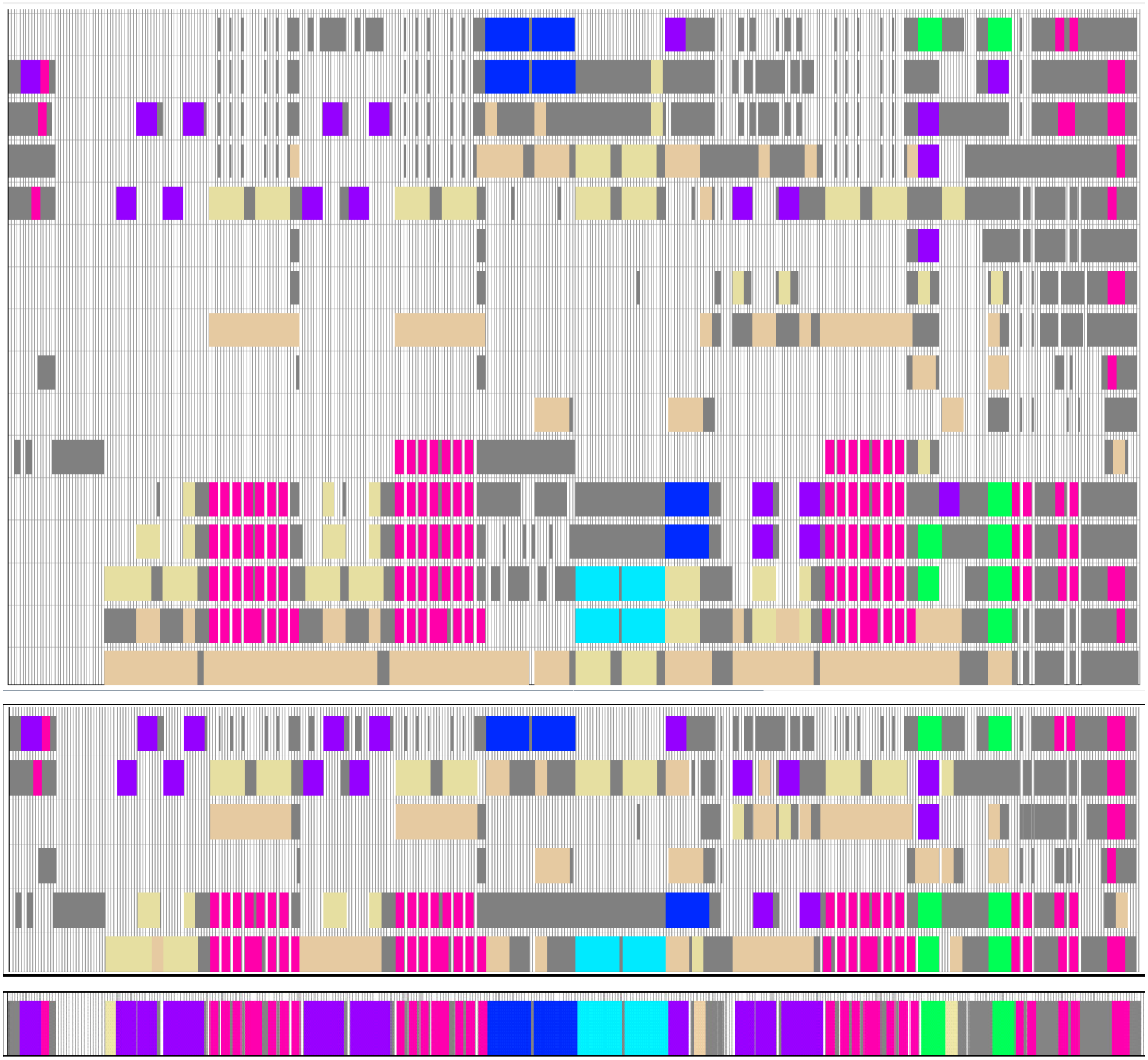

Figure 2. (Upper) Visualization result of ”Valse des fleurs” by Tchaikovsky. (Center) Vertical condensation into 6 tracks. (Lower) Vertical condensation into 1 track.

even though a miniature score of ”Valse des fleurs” we bought occupies 33 pages. Many traditional classical musical works have two themes, and forms musical structures while repeating and varying these two themes. Also, they may contain several additional melodies delivered from the themes. Considering such composition techniques, we prepared five melody patterns to visualize the music. We also prepared typical Waltz patterns for harmonic and bass accompaniments.

Also, we can observe various orchestration techniques from the visualization result. We can find that roles are switched across the tracks, or shared by several number of tracks, along the variation of the music. We can also find combinations of multiple melodies at the same time.

B. *Vertical condensation*

Figure 2(Center)(Lower) shows the condensation of the visualization result shown in Figure 2(Upper). Figure 3(Left) shows a part of the visualization result shown in Figure 2, corresponding to 314 to 328 bars. It also shows a result of the second step of vertical condensation. It reduces the number of tracks from 16 to 1, 2, 4, or 6, and note-blocks from 29 to 4, 6, 11, or 13. This result shows that a melody drawn in pink are played two bars after a melody drawn

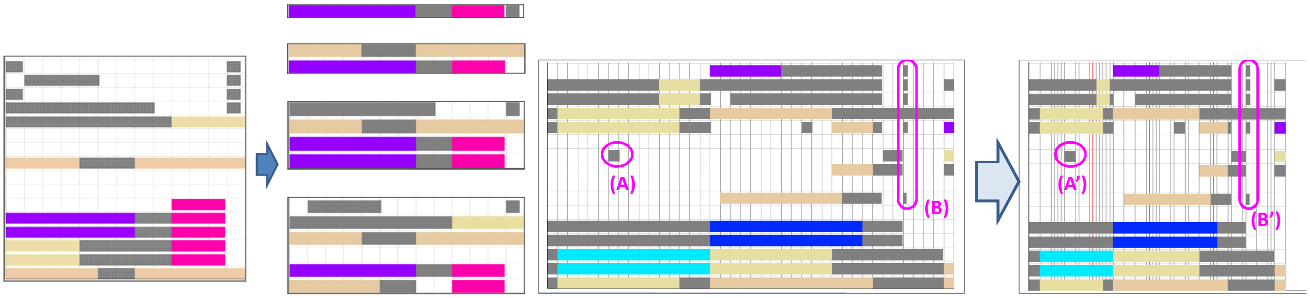

Figure 3. (Left) Vertical condensation into 1, 2, 4, or 6 tracks. Melodies drawn in purple and pink are preferentially remained when the number of tracks are fewer. (Right) Horizontal condensation. The bars (A) and (B) remained wider while other bars get narrower.

in purple. Colorscore remained this structure even in the 1-track condensation result. On the other hand, it remained bass patterns in the 2-tracks result, and other blocks in the 4- and 6-tracks results. The result shows that Colorscore adaptively realizes level-of-detail control of condensation.

C. Horizontal condensation

Figure 3(Right) shows a part of the visualization result in Figure 2, corresponding to 211 to 250 bars. It also shows a result of horizontal condensation. It reduces the width of the visualization space as approximately 60% of the original width. However, it does not shrink the timings when roles of note-blocks change: for example, it keeps the length of short note-blocks surrounded by circles (A) and (B) after the horizontal condensation, indicated as (A') and (B') in Figure 3(Right). With this functionality, Colorscore can draw longer music in limited display space, while preventing us from missing changes of roles. We think it is useful for various purposes of music understanding: for example, students majoring conduct can practice signs to players just before they have new roles.

V. EVALUATION

This section introduces our two kinds of evaluations to demonstrate effectiveness of Colorscore.

A. Understanding of novice examinees

We tested how novice people can understand the musical structure. First, we showed printed visualization result shown in Figure 2(Upper) to examinees. Then, we asked to mark bars to examinees, if they think that the bars can be separators. At the same time, three authors marked bars as examinees did, and treated a set of bars which two or more authors marked as correct answers. Consequently, we made 12 bars as correct answers.

We then calculated precision and recall of the answers of examinees. We tested with 27 university students, divided into two groups. "Group A" contained 18 of the examinees who had experiences of playing musical instruments and reading notes, but did not have experiences of reading scores. The other, "Group B" contained 9 of the examinees who did not have such experiences. The experiment obtained precision of 0.8056 from Group A, and 0.6991 from Group B. Also, it obtained recall of 0.7870 from Group A, and 0.7037 from Group B. We think the result is totally good; however, differences between two groups were not ignorable. As future works, we would like to observe user tests more, and improve Colorscore reflecting these tests. Especially, we would like to carefully discuss how the visualization results are mistakable for non-experienced people.

B. Feedback from music computing researchers

We showed the visualization result shown in Figure 2 to music computing researchers and asked to give us feedbacks. We asked three researchers majoring music computing who are players in amateur orchestras, including authors of musical visualization techniques [5] [7]. This section summarizes their comments and suggestions.

[Discovery from the visualization result]
Answerers pointed that they discovered features of the music from the visualization result as follows:
1) Coloring is so effective that it is easy to understand the musical structures. For example, the theme indicated in pink appears three times, and finally arranged in the Coda. Other repetitions are also easy to understand.
2) Coloring is also effective to understand the roles of the musical instruments. Examples are as follows:, The theme indicated in pink is mostly played by the strings. Double Bass and Tuba do not play any melodies. Trombone and Percussions have long rests.
3) It is easy to visually compare the roles of specific track with others. For example, when horns play the theme, Violoncello and Double Bass play accompaniments in many bars. When horns play harmonies, strings play melodies, and wood winds play decorations.
4) It is easy to understand that the music gets more sensational in the latter part.

These discoveries were exactly what we expected, and we felt confident about the effectiveness of Colorscore.

[Possible applications]
Answerers pointed that Colorscore can be used as the following applications:

1) Additional musical information display for music player softwares or devices, or digital score softwares.
2) Education of music structure understanding.
3) Reference for staging, camera work, and choreography.

Actually we expect Colorscore can be used as above applications, and needs to develop more functions so that Colorscore can be used as above.

[Concerns]

Answerers pointed that they felt concerns about the visualization result and functionality of Colorscore as follows:
1) Since Colorscore horizontally shortens the musical score, beginners may feel that the first and the last bars are so close, and the top and the bottom tracks (usually piccolo and double bass) are so distant.
2) When we apply Colorscore to music much longer than "Valse des fleurs," it may be difficult to show the whole music in one display space. Authors need to test with such long music and discuss how to effectively show it.
3) Effectiveness of visualization results may depend on the number of patterns. It may be misunderstanding if the number is too small and therefore same colors are assigned to moderately similar note-blocks. On the other hand, it may be difficult to distinguish the roles of note-blocks if too many colors are used.

The first and second comments suggest that we need to have tests with more various users and music, and improve based on the results of the tests. The third comment suggests that we need to discuss what kinds of melody and accompaniment pattern definitions are effective for Colorscore. Also, it may be interesting to integrate Colorscore with automatic theme finding techniques.

[Potential future issues]

Answerers pointed that the following would be potential future issues:
1) Play of the music by clicking arbitrary points, or arbitrary tracks of the visualization result. .
2) Display of the specific tracks or note-blocks as staff notation.
3) Sound of harmonies or chords of arbitrary points.
4) Functions that when users click arbitrary note-blocks or interactively input melodies, it can jump to display the bars corresponding to the specified note-blocks or melodies.
5) Navigation for novice users to the most important parts of the score.
6) User interfaces and animations to smoothly control and display while condensation.

We have not developed the above interactive functions yet, but actually we think they are very important.

VI. CONCLUSION AND FUTURE WORK

This paper presented Colorscore, a visualization and condensation technique for MIDI-based classical music data. This paper introduced three components: role determination, overview visualization, and condensation. This paper also introduced user evaluations, and discussed its effectiveness and problems.

As short-term future works, we would like to develop additional functions discussed as potential future issues in Section V-B. Also, we would like to test Colorscore with more variety of users and music, to find more issues to improve Colorscore. Especially, we have not tested condensation functions with examinees yet, and therefore not discussed their issues well. Such additional tests will be helpful for our future enhancement.

Following are our long-term future issues. We are interested in more detailed information display of note-blocks. Especially, we would like to represent differences of note-blocks painted in the same colors, by assigning independent textures or glyphs. We are also interested in development of horizontal condensation so that it can detect repetition of note-blocks and shrink the repeated note-blocks.

As discussed in Section V-B, it may be an essential problem that we need to manually input melodies and accompaniment patterns, and therefore visualization result may strongly depends on the input. We would like to discuss how Colorscore can integrate with automatic theme or pattern discovery techniques, so that Colorscore can provide visualization results not depending on users' input.

REFERENCES

[1] J. Foote, Visualizing Music and Audio using Self-Similarity, 7th ACM International Conference on Multimedia, Part 1, pp. 77-80, 1999.

[2] M. Goto, SmartMusicKIOSK: Music Listening Station with Chorus-Search Function, 16th Annual ACM Symposium on User Interface Software and Technology, pp. 31-40, 2003.

[3] C. S. Sapp, Harmonic visualizations of tonal music, Proc. International Computer Music Conference (ICMC, Havana, Cuba, 2001), pp. 423-430, 2001.

[4] M. Wattenberg, Arc diagrams: Visualizing structure in strings, Proc. IEEE Symposium on Information Visualization 2002, pp. 110-116, 2002.

[5] R. Miyazaki, I. Fujishiro, and R. Hiraga, comp-i: A System for Visual Exploration of MIDI Datasets, Transactions of Information Processing Society of Japan, vol. 45, no. 3, pp. 739-742, 2004.

[6] M. Matsubara, H. Okamoto, T. Sano, and H. Susuki, Scoreilluminator: Automatic illumination of orchestra scores for readability improvement, International Computer Music Conference (ICMC 2009), 2009.

[7] F. Watanabe, I. Fujishiro, and R. Hiraga, Music Learning through Visualization, Second International Conference on WEB Delivering of Music (WEDELMUSIC'02), 2002.