2. There are two reasons for the improvements. One is that OS dependent GUI can be achieved by using wxWidgets, and the other is that its operation becomes simpler.
3. From MOS and SUS scores, easy to use evaluation was obtained. The present version is Japanese based with a few English explanations. Onomatopoeia is totally in Japanese.

5. CONCLUSION
We carried out restructuring and integration of subsystems of electronic timbre dictionary in C++. Due to the development of a unified language, processing speed of the proposed system has been improved. Moreover, adoption of wxWidgets enabled multi-platform by a single source code.

The usability evaluation experiment was done to find the sound by using the search function of the three types, which resulted in a 96.3% rate of task completion, and a MOS (easiness) of 2.9, which is better than the previous system. For overall evaluation, a MOS of 3.0 and 79.8 SUS score were acquired. These scores are considered to be satisfactory.

In the future, we will further improve the usability GUI, to build on Mac OS and Linux, and to proceed towards the release of the full-scale system.

6. REFERENCES

Figure 4. Comparison of the average time it takes each condition of all search functions

Figure 5. Comparison of the MOS (easiness) for each condition of all search functions

4.4. Other evaluation aspects
Correlation between search time as a physical evaluation and the MOS (easiness) of 18 conditions for the previous system was as low as -0.3, while the new system has a correlation of -0.79, which is a sufficiently high negative correlation between the two evaluation values. This is interpreted as follows: a large amount of time was spent in superficial GUI operation that was not search-related in the previous system, while in the new system, most of the time was spent on search due to the efficiency of the improved GUI, with little time needed for GUI operation.

The average value of each MOS (easiness) of all 18 conditions for the previous system is 4.4. and the MOS (easiness) of 18 conditions for the previous system was improved to 2.9. and the MOS (easiness) of 18 conditions for the previous system was improved to 2.9.

5. CONCLUSION
We carried out restructuring and integration of subsystems of electronic timbre dictionary in C++. Due to the development of a unified language, processing speed of the proposed system has been improved. Moreover, adoption of wxWidgets enabled multi-platform by a single source code.

The usability evaluation experiment was done to find the sound by using the search function of the three types, which resulted in a 96.3% rate of task completion, and a MOS (easiness) of 2.9, which is better than the previous system. For overall evaluation, a MOS of 3.0 and 79.8 SUS score were acquired. These scores are considered to be satisfactory.

In the future, we will further improve the usability GUI, to build on Mac OS and Linux, and to proceed towards the release of the full-scale system.

6. REFERENCES
2.2 Media files handling and video input

‘DIPSrPixTable’ object imports still image files as well as movie files from the storage devices to the main memory to use them in DIPS patches. On the other hand, ‘DIPSrQTPlayer’ plays streaming movies directly from the storage. ‘D3DcObjTable’ imports 3D model files in .obj format to DIPS patches. (see Figure 3)

‘DIPSrVideoIn’ object captures live video input from cameras attached to the computer. It can handle multiple video camera inputs at the same time. Those objects also have wrapper objects: ‘DlibImageTable’ and ‘DlibMovieTable’ in the case of ‘DIPSrPixTable’, ‘DlibQTPlayer’ in the case of ‘DIPSrQTPlayer’, and ‘DlibVideo’ and ‘DlibVideoIn’ for ‘DIPSrVideoIn’. Besides these media porting objects, managing texture memory is another critical issue to handle media files in 3D programming. The DIPS offers ‘DlibTexGenerator’ object to create a texture memory, and ‘DlibTexBind’, ‘DlibTexImage’, and ‘DlibTexMovie’ objects bind image files to the texture. Sphere texture mapping is available as well as ordinary texture mapping using ‘DlibTexRect’ and ‘DlibTexQuads’ objects.

2.3 Implementation of Core Image Filter and importing Quartz Composer files

As we described in our paper regarding the DIPS3 in the ICMC 2007, ‘DCIFilter’ object implements Apple’s Core Image Filter. In DIPS5 about two dozens of newly released Core Image Filter are added, and more than ninety of them are now available in DIPS. Those are introduced to the DIPS as Dfx objects along with other visual effect objects. The parameters of Dfx objects can be changed from the control panel of each object that can be opened by double-clicking the Dfx object. (see Figure 4) This control panel function is also implemented to most of Dlib objects. The number of Dfx objects exceeds one hundred ten in February 2013. On the other hand, another object introduced in the ICMC 2007, ‘DIPSrRenderer’ enables to import user-programmed Quartz Composer files to Max programming environment. By sending message ‘getInputKeys’ to ‘DIPSrRenderer’ object, labels of parameters of Quartz Composer file in Max patch can be obtained.

2.4 DIPS utility library ‘Dlib’ objects

Most of fundamental OpenGL objects have their wrapper objects in order to make OpenGL programming handier and to make it controllable from their control panels rather than sending numbers to the inlets of each object in Max patch. For instance, ‘DGLUTSolidTeapot’ and ‘GLUTWireTeapot’ objects are merged as a single object called ‘DlibTeapot’, and it can be switched between wire frame and polygon display mode as well as its size can be changed from its control panel. Now, more than seventy such kind of Dlib objects are implemented.

2.5 Integration of DIPS windows

Another superior point of the DIPS is the integration of DIPS windows at the final stage of DIPS programming. DIPS users may create and render more than one DIPS windows and want to integrate them into a single or a few DIPS windows to be projected to the screen. Rendered DIPS windows can be ported instantly to another DIPS window as a texture using DIPS objects such as ‘DIPSsSurfaceTexture’, ‘DlibTexSurface’, and ‘DlibTexCopy’.

Furthermore, ‘DIPSrWindowMixer’ object offers the flexible video mixture just like a hardware video mixer. It can realize scaling, shifting center position as well as fade-in and fade-out of each DIPS widow. (see Figure 5)

3. New features of DIPS version 5

We introduced a few motion detect functions in the previous version of the DIPS. At the last release of DIPS4 in 2009, we added DIPS OpenCV objects. In this release of DIPS5, OpenCV function is enhanced. Now, ‘DlibTrack’ object, one of DIPS OpenCV objects, can detect not only the position of face but also more details such as mouth, nose and eyes.

DIPSKinect object captures DepthImage from Kinect sensor and outputs the distances between the Kinect sensor camera and objects in certain areas and specific pixel points of Kinect DepthImage input. More of Kinect sensor functions are planned to be added to the DIPS along with new experimental objects derived from such as ARTskit.

In addition, the image in ‘jit.matrix’ is able to be ported to DIPS window using ‘DIPSjixMat2DIPS’ object as well.

4. Example of a DIPS video effect programming

After creating DIPS window, just place ‘DIPSVideo’ object between ‘DIPSCurrentWindow’ and ‘DIPS SwapBuffer’. (see Figure 6) Then, the incoming video camera image is rendered in DIPSWindow by turning on metro that keeps sending bang to them or stopping it. Parameters of each Dfx object can be changed from its control panel. One of DIPS motion detect objects is added to this example patch. (see Figure 8)
2.2 Media files handling and video input

‘DIPSpxTable’ object imports still image files as well as movie files from the storage devices to the main memory to use them in DIPS patches. On the other hand, ‘DIPSQTPlayer’ plays streaming movies directly from the storage. ‘D3DOBITable’ imports 3D model files in .obj format to DIPS patches. (see Figure 3)

‘DIPSVideoIn’ object captures live video input from cameras attached to the computer. It can handle multiple video camera inputs at the same time. Those objects also have wrapper objects: ‘DlibImageTable’ and ‘DlibMovieTable’ in the case of ‘DIPSpxTable’; ‘DlibQTPlayer’ in the case of ‘DIPSQTPlayer’; and ‘DlibVideo’ and ‘DlibVideoIn’ for ‘DIPSVideoIn’.

Besides these media porting objects, managing texture memory is another critical issue to handle media files in 3D programming. The DIPS offers ‘DlibTexGenerator’ object to create a texture memory, and ‘DlibTexBind’, ‘DlibTexImage’, and ‘DlibTexMovie’ objects bind image files to the texture. Sphere texture mapping is available as well as ordinary texture mapping using ‘DlibTexRect’ and ‘DlibTexQuads’ objects.

2.3 Implementation of Core Image Filter and importing Quartz Composer files

As we described in our paper regarding the DIPS3 in the ICMC 2007, ‘DCIFilter’ object implements Apple’s Core Image Filter. In DIPS5 about two dozens of newly released Core Image Filter are added, and more than ninety of them are now available in DIPS. Those are introduced to the DIPS as Dfx objects along with other visual effect objects. The parameters of Dfx objects can be changed from the control panel of each object that can be opened by double-clicking the Dfx object. (see Figure 4) This control panel function is also implemented to most of Dlib objects. The number of Dfx objects exceeds one hundred ten in February 2013. On the other hand, another object introduced in the ICMC 2007, ‘DIPSQCRenderer’, enables to import user-programmed Quartz Composer files to Max programming environment. By sending message ‘getInputKeys’ to ‘DIPSQCRenderer’ object, labels of parameters of Quartz Composer file in Max patch can be obtained.

2.4 DIPS utility library ‘Dlib’ objects

Most of fundamental OpenGL objects have their wrapper objects in order to make OpenGL programming handier and to make it controllable from their control panels rather than sending numbers to the inlets of each object in Max patch. For instance, ‘DGLUTSolidTeapot’ and ‘GLUTWireTeapot’ objects are merged as a single object called ‘DlibTeapot’, and it can be switched between wire frame and polygon display mode as well as its size can be changed from its control panel. Now, more than seventy such kind of Dlib objects are implemented.

2.5 Integration of DIPS windows

Another superior point of the DIPS is the integration of DIPS windows at the final stage of DIPS programming. DIPS users may create and render more than one DIPS windows and want to integrate them into a single or a few DIPS windows to be projected to the screen. Rendered DIPS windows can be ported instantly to another DIPS window as a texture using DIPS objects such as ‘DIPSSurfaceTexture’, ‘DlibTexSurface’, and ‘DlibTexCopy’.

Furthermore, ‘DIPSWindowMixer’ object offers the flexible video mixture just like a hardware video mixer. It can realize scaling, shifting center position as well as fade-in and fade-out of each DIPS widow. (see Figure 5)

3. New features of DIPS version 5

We introduced a few motion detect functions in the previous version of the DIPS. At the last release of DIPS4 in 2009, we added DIPS OpenCV objects. In this release of DIPS5, OpenCV function is enhanced. Now, ‘DlibTrack’ object, one of DIPS OpenCV objects, can detect not only the position of face but also more details such as mouth, nose and eyes. DIPSKinect object captures DepthImage from Kinect sensor and outputs the distances between the Kinect sensor camera and objects in certain areas and specific pixel points of Kinect DepthImage input. More of Kinect sensor functions are planned to be added to the DIPS along with new experimental objects derived from such as ARToolKit.

In addition, the image in ‘jit.matrix’ is able to be ported to DIPS window using ‘DIPSjitMat2DIPS’ object as well.

4. Example of a DIPS video effect programming

After creating DIPS window, just place ‘DIPSVideo’ object between ‘DIPSCurrentWindow’ and ‘DIPS_swapBuffer’. (see Figure 6) Then, the incoming video camera image is rendered in DIPSWindow by turning on metro that keeps sending bang to them or stopping it. Parameters of each Dfx object can be changed from its control panel. One of DIPS motion detect objects is added to this example patch. (see Figure 8)
5. CONCLUSION

The DIPS is a powerful and user-friendly programming tool for the creation of interactive multimedia art, supporting interaction between sound events and visual events in Max programming environment using Apple’s Core Image technology as well as OpenGL and OpenCV technologies. DIPS consists of a library of more than three hundreds Max external objects and abstractions, a comprehensive set of sample patches, and a detailed tutorial. The DIPS5 is free plug-in software for Max/MSP running on Macintosh computer. It is downloadable from http://dips.dacreation.com.

6. REFERENCES


