DIPS5 FOR MAX

- Digital Image Processing with Sound, version 5 -

Takayuki Rai
Lancaster University
Lancaster Institute for the Contemporary Arts
t.rai@lancaster.ac.uk

Shu Matsuda
Kunitachi College of Music
Sonology Department
shu@kcm-sd.ac.jp

Yota Morimoto
University of Birmingham
yotamorimoto@gmail.com

Takyuto Fukuda
Universität für Musik und darstellende Kunst Graz
Institut für Elektronische Musik und Akustik - IEM
takujin3@yahoo.co.jp

Keitaro Takahashi
Musik Akademie Basel
Elektronik Studio Basel
neoterize@mac.com

ABSTRACT

The DIPS was first presented in the international computer music world at the ICMC 2000 in Berlin as a plug-in software for the legendary Max family application 'jMax' in order to perform the real-time image-processing in Max patches, thus to support the creation of interactive multimedia art. In 2005, the DIPS was ported to Max running on Macintosh computer, and in 2007, Dlib (DIPS utility library) and Dfx (DIPS visual effect) objects were introduced to the DIPS to make image-processing programming much simpler and easier. This version of DIPS 'DIPS3' was presented at the ICMC 2007 in Copenhagen, and it kept evolving for the last five years. Now, the DIPS consists of more than three hundreds Max external objects and abstractions. At this occasion of ICMC 2013, we would like to introduce the new features of DIPS5 including DIPS OpenCV objects and demonstrate how easily the real-time image-processing can be programmed in Max environment.

1. INTRODUCTION

The DIPS was developed for Max/FTS first in 1997 by Shu Matsuda at Sonology Department, Kunitachi College of Music in Tokyo. It was ported to jMax programming environment in 2000, and later to Max running on Macintosh computer. The early versions of DIPS consisted of only DIPS core objects and DGL objects those are wrapper objects of various OpenGL functions. Therefore, to realize 3D image-processing using DIPS the knowledge of OpenGL programming technique was still essential. To resolve this difficulty for ordinary composers and artists, we introduced Dlib (DIPS utility library) and Dfx (DIPS visual effect) objects in the third version of DIPS in 2007. Since then further Dlib and Dfx objects have been developed as well as new functions such as OpenCV and Kinect sensor technology have been implemented. Here, we would like to introduce those enhanced features and demonstrate its ease of programming, power, and efficiency in Max programming environment.

2. Principal DIPS objects

2.1 DIPS core objects

The DIPS programming in Max begins with creating 'DIPSWindow' object (see Figure 1), where the result of the image-processing calculations will be rendered. The image-processing and the constructing and handling procedure of 2 and 3 dimensional objects must be programmed between 'DIPSSetCurrentWindow' and 'DIPSSwapBuffer' objects with specifying target 'DIPSWindow'. Multiple DIPSWindows can be created with any window sizes. 'DlibWindow' object is the wrapper object of 'DIPSWindow' to simplify the creation of 'DIPSWindow'. All of DIPS objects are bang-oriented; therefore, they don't consume any CPU power, unless they get bang at the leftmost inlet as every Max control object does. (see Figure 2)

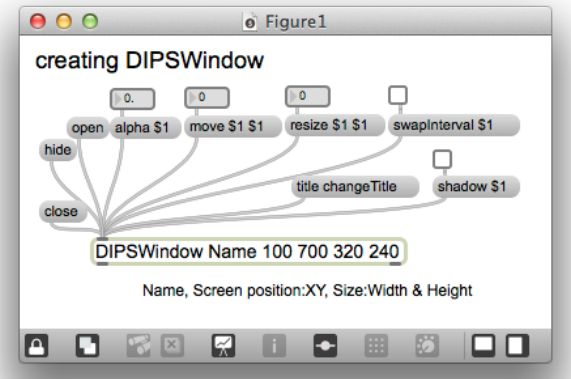

Figure 1. DIPSWindow object

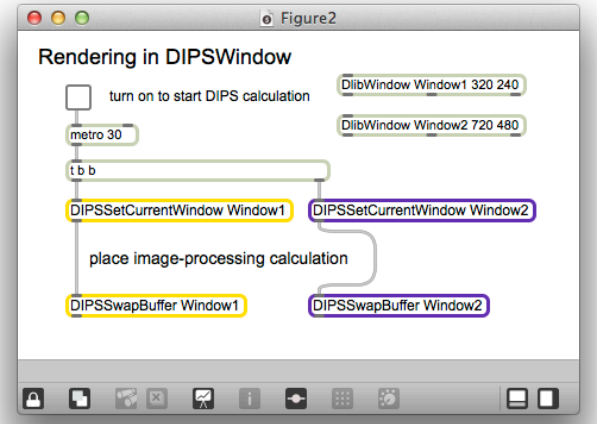

Figure 2. basic DIPS programming structure

2.2 Media files handling and video input

'DIPSPixTable' 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. 'D3DOBJTable' 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 'DIPSPixTable', '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.

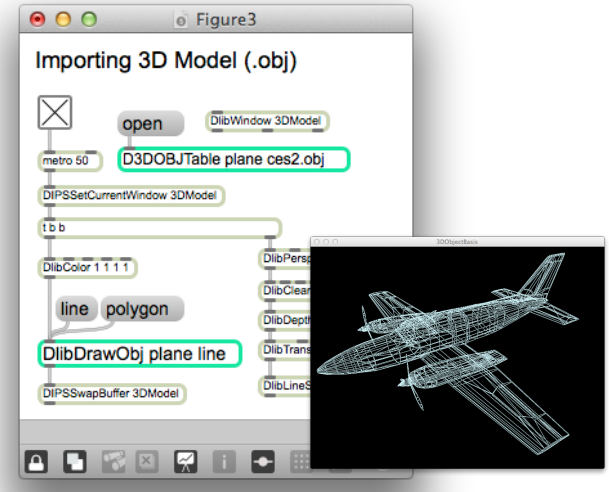

Figure 3. D3DOBJTable

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.

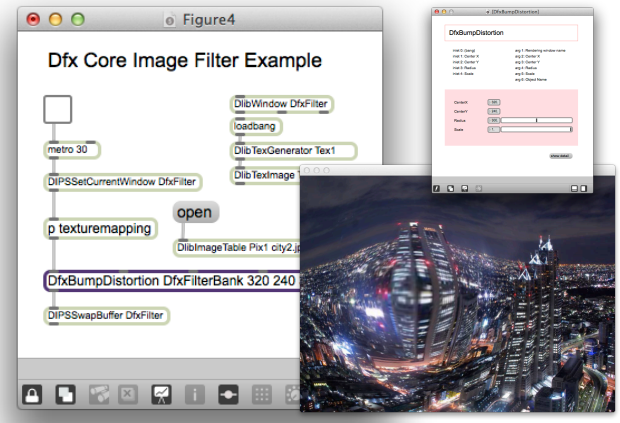

Figure 4. DfxBumpDistortion

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 'DlibTeaport', 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)

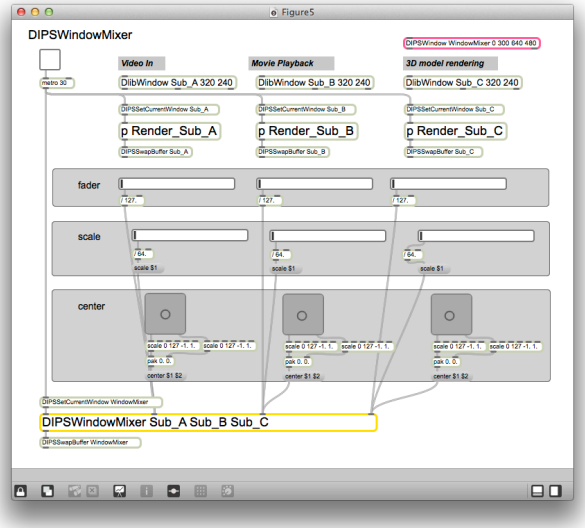

Figure 5. Integration of three DIPS windows

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 'DIPSSwapBuffer'. (see Figure 6) Then, the incoming video camera image is rendered in DIPSWindow by turning on metro that keeps sending bang to DIPS objects. The diverse Dfx objects can be inserted after DIPSVideo object. (see Figure 7) They can be turned on and off by 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)

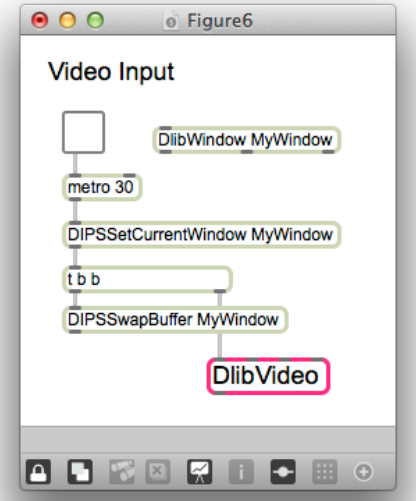

Figure 6. Video image capture

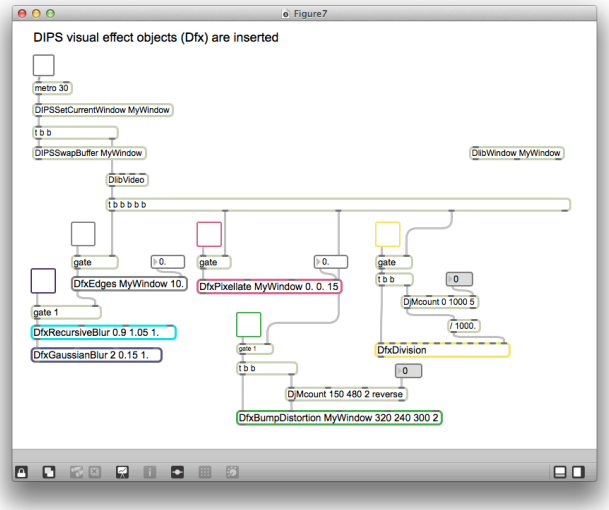

Figure 7. multiple video effect

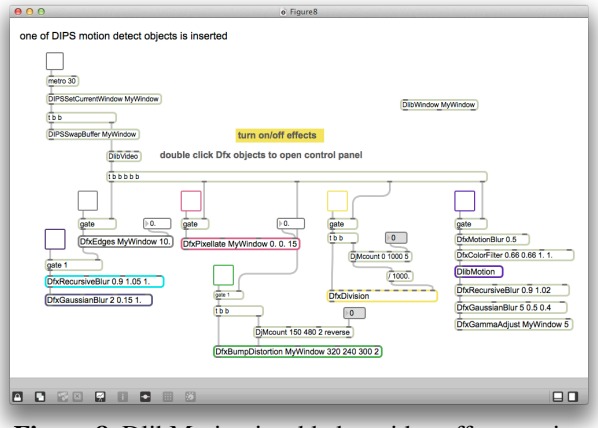

Figure 8. DlibMotion is added to video effect routine

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

[1] Matsuda, S.,Rai, T., DIPS : the real-time digital image processing objects for Max environment, in Proceedings of the International Computer Music Conference 2000.

[2] Matsuda, S., Miyama, C., Ando, D., Rai, T., DIPS for Linux and Mac OS X, in Proceedings of the International Computer Music Conference 2002.

[3] Miyama, C., Rai, T., Matsuda, S., Ando, D., Introduction of DIPS Programming Technique, in Proceedings of the International Computer Music Conference 2003.

[4] Rai, T., Miyama, C., Matsuda, S., Morimoto, Y., Hamano, T., Introduction of DIPS3 (version 2) for Max MSP, in Proceedings of the International Computer Music Conference 2007.

[5] OpenGL Architecture Review Board, Shreiner, D., Woo, M., Neider, J., Davis, T., (2009), OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL, Version 3.0 and 3.1, Addison-Wesley.

[6] OpenGL Architecture Review Board, Editor : Shreiner, D.,(2004). OpenGL(R) Reference Manual: The Official Reference Document to OpenGL, Version 1.4(4th ed.). Addison-Wesley.

[7] Apple Inc., Apple Developer Connection : Core Image Programming Guide.,(2004, 2013), http://developer.apple.com/documentation/GraphicsImaging/Conceptual/CoreImaging/index.html

[8] J.Rost, R., M.Kessenich, J., Lichtenbelt, B., Malan, H., Weiblen, M. Bailey, M.,(2006), OpenGL(R) Shading Language(2nd ed.), Addison-Wesley.

[9] opencv dev team, OpenCV 2.4.2 documentation : Reference Manual, (2012) http://docs.opencv.org/2.4.2/modules/refman.html

[10] openkinect.org, OpenKinect, (2013), http://openkinect.org/wiki/Main_Page

[11] Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tachibana, K. (2000), Virtual Object Manipulation on a Table-Top AR Environment, In proceedings of the International Symposium on Augmented Reality, pp.111-119, (ISAR 2000), Munich, Germany.

[12] The Human Interface Technology Laboratory (HIT_Lab) at the University of Washington, HIT_Lab_NZ at the University of Canterbury, New Zealand, and ARToolworks, Seattle, (2013), http://www.hitl.washington.edu/artoolkit/