CS5343 Advanced Computer Animation

Semester 2, 2010-2011

03-May-2011

School of Computing

Kinect Motion Capture with Maya

Professor: Kang Kang YIN

Student: Ying-Chieh HUANG

Master of Computing, A0066253H

Introduction

Since the first official launch of Microsoft Kinect in November 2010, it soon becomes the fast-selling gadget in the world. People love it because Kinect sensor provides a nature user interface to connect reality with virtual space. For the game programmer/designer in the gaming industry, this huge impact introduces brand-new interactivity with game content by providing the real-time skeleton extraction. In the film industry, the expensive motion capture system had used for post-editing skeleton animation in a very accurate demand, however, low cost Kinect motion capture system is gradually built up by hackers and other third parties. It provides a more mature, accessible, and stable motion capture for filmmakers.

Moreover, all kind of mash-up applications using Kinect sensor is published and developed everyday (*1). And it merges the boundary of game and film industry. For example, the real-time motion capture capability allow user to do special effect at home in an interactive way, like indie games (*2). And some high quality short films (*3) are produced recently by small studios using kinect motion capture system. This project is implemented in a way to help leverage Autodesk Maya as a powerful environment for Kinect Motion Capture pipeline.

Indie game: MikuMikuDance

Indie game: Kamehameha

Short film: All Nighter

Short film: Woodman Animation

Motivation

To make Autodesk Maya a more interactive modeling, animation, film developing environment, Kinect sensor may change the way people interact with virtual content by providing the first person perspective view. Especially, in the pipeline of pure CG movie production, actors usually don't know how to interact with virtual scene (ex: room), object (ex: cup), or character (ex: monster). Director need to provide actors enough imagination for them to interact with virtual content, so that the VFX compositing / match moving have to adjust accordingly by post-processing. Ex: While producing the movie, Avatar, Director James Cameron actually uses this Augmented Reality technology to guide actor how to interact with virtual scene.

Therefore, by controlling virtual camera view using Kinect, the real-time visualization of the 3D scene becomes interactive and intuitive. While Director's point of view is accessible, then actors can roughly know how they should interact with the virtual scene. With the possibility and capability to make ad-hoc film shooting scenario, director and actor don't need to pretend something over there in the empty space. The director can immediately show actors the environment built in Maya. For example, if the actor didn't precisely fellow the original setting of the interaction. Director could notice that in the first place without re-taking the whole shoot in another time. This idea is now prototyped into Maya, which allows user to interact with virtual scene by holding the screen (*6). Hopefully, It could speed up the film making process.

Related Work

Recently, Kinect motion capture pipeline are well-established in Autodesk Motion Builder by using commercial tools, iPiSoft (*4), and free tools, Brekel (*5). Brekel provides real-time streaming to Motion Builder and lots of useful detailed setting. However, both of them are not available for real-time motion capture data streaming to 3D Max and Maya. The real-time streaming capability allows Maya user to directly link up Kinect data with virtual characters and scene and greatly simplify their work without presetting scenario in Motion Builder.

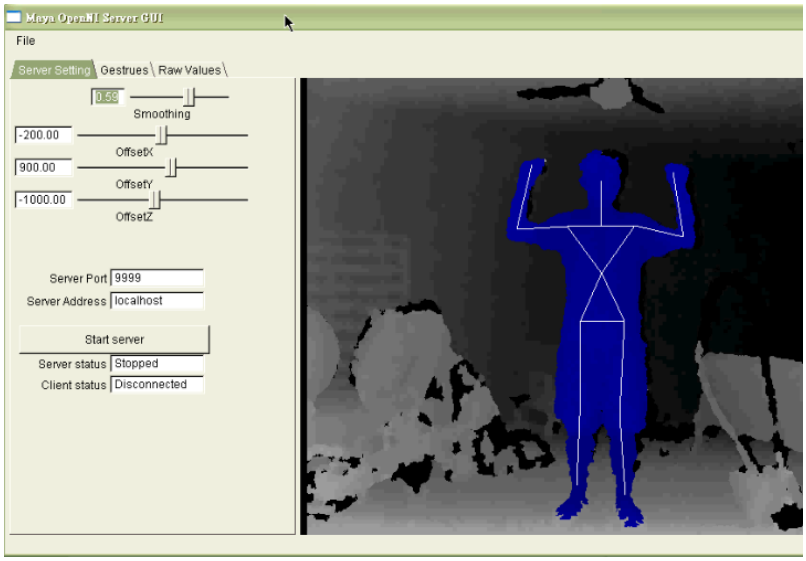
Another great work is just done recently (end of April) by EON Reality (*10), Singapore. They integrate Kinect motion capture data into virtual reality with very immersive interaction and visualization.

Implementation

During 2 month of intensive implementation, I had been familiar with Kinect Setup, PyMEL in Maya 2010, GUI, Automatic Kinect skeleton setup, Interactivity with Gesture, Visualization in Autodesk Maya 2010.

Kinect Setup

Except installing basic Kinect sensor driver and middleware (installing guide, *8), I mainly use "MayaOpenNIServer" socket server application to process Kinect data and stream it to Maya.



Python and PyMEL (setup) in Maya 2010

Therefore, the main scripting language is Python using PyMEL, Maya official Python API at Autodesk Maya 2010 Environment, which PyMEL is still not native build-in. We also need to prepare Microsoft Kinect Sensor hardware, and install all the necessary driver and middleware including PrimeSense driver, OpenNI/NITE(*7) middleware, PyMEL, and MayaOpenNIServer. Please follow the Readme (*8) file for setup details.

PyMEL (*9) was authorized as Audodesk Maya's official Python API since 2011 version, and it had been used in several movies like "Kung Fu Panda", "Shrek 4", and "How to Train Your Dragon" by DreamWorks, " Pirates of the Carribean" and "Harry Potter 6" by Luma Pictures. Although it's still new to the film industry and lack of resource on the Internet, I still would like to try it because of the merit of Python scripting style.

GUI

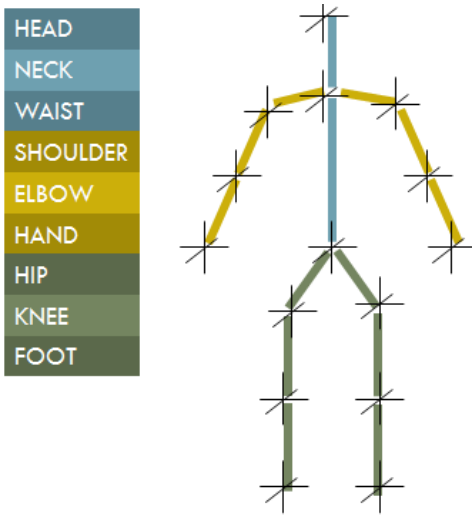
The GUI design is based on Mr. Yasutoshi Mori's MayaOpenNIServer_02Alpha MEL script. Porting from MEL to PyMEL is still considered a bothering work for beginner. Most of the functions are reformatted and enhanced. Ex: The Record function won't work in the Alpha version. And I also create automation skeleton setup specifically for Kinect… etc. Artists won't need to set up those skeletons joints and locator's naming or assigning attributes.

Automatic Kinect Skeleton Setup

However, the Kinect skeleton data is subjected to few technical constraints. For example, firstly, the OpenNI (NITE) library (*7) extracts the limbs and body skeleton data from Kinect sensor's depth map data in real-time. But the two arms or legs are sometimes twisting, jittering, or flickering, especially if they are too close to each other. Or if arms are occluded or stuck with body, the articulated skeleton will try to fit the possible pose, but it's still up to precision.

Secondly, the Kinect Motion Capture data are lack of the orientation information of hand, and feet. If all those details information above are inadequate, the live character retargeting process will be more complex and tedious. These two issues are still not well solved or developed in Maya environment yet. It can be assigned automatically though this GUI setting easily in near future.

To solve above issue, automatic Kinect skeleton setting, the three buttons of "Create Skeleton", "Create IK" and "Recreate Skeleton", had included all the repetitive and tedious work of naming locator, mapping joint, creating correct skeleton, creating Inverse Kinematics.



Default Skeleton only have 15 joints without the finger, torso, collar, angle, …etc. The main propose for the scarification is on behalf of real-time

articulate skeleton detection and tracking.

And, the IKs method is a good attempt to solve stability issue. But I found that the IKs alone are still not enough, joint rotations are noisy (only pelvis uses joint orientation) and limb distances are not the same in model and in NITE skeleton, so some funny things occur (like horse knees). Therefore, I managed the new IKs skeleton with default skeleton by recording the joints location. The IKs will only be created by point's constraint of joints location. That is not only stabilize the skeleton, and also eliminate sudden horse leg.

Interaction with Gesture

Wave Gesture and Click Gesture can be recognized via OpenNI. It provides more controls for motion capture scenario. Basically, in this project, I intentionally set Wave gesture as "Create IK" and Click gesture as "Recreate Skeleton". That works well when I am debugging my motion capture data in real time. The IKs skeleton can be generated more correctly and precisely according to my current default skeleton. If no one help me to click the button, I may have to run out of screen which will lead to weird IKs skeleton.

Issue:
PyMEL commandPort's **sourceType** can't change to Python command**.**
I think that is the instability of current PyMEL. Some query function or key word argument are somehow disable. Luckily, I still find the way out to solve this issue by using python predefined name, so that I can pass python command under MEL command.
For python: python "(mayaKinectGesture(\"Wave\"))"
For mel: mayaKinectGesture("Wave")

Visualization

Director and actors can interact with virtual scene
Virtual Reality / Augmented Reality

Build interactive Maya by linking skeleton data to virtual camera
- Mapping head position and rotation to the camera attribute
- Control camera perspective with hand position and orientation
- Control virtual camera parameter (AspectRatio) using PyMEL API
- Build Python GUI to automate this process

Issues and difficulty

1. Camera - Kinect raw data (set point constraint), Gimbal Lock (attribute), Camera Setting (lock of attribute) (issue: camera can't flip to front view)
2. Mel, Python in Maya, PyMEL – Global variable can only be accessed, but can't be assigned. I found an issue while I never face this kind of problem at normal Python environment.
http://www.mail-archive.com/python_inside_maya@googlegroups.com/msg01165.html

Results

As described above, 3 main tasks of this project are accomplished and demonstrated here.

First one is to solve the skeleton jittering problem as much as possible, and use suitable method like inverse kinematic and smoothing. Second, gesture signal can be recognized in OpenNI middleware and provides immediate interaction to Maya. I experiment wave gesture and click gesture to activate automatic normal skeleton setup and inverse kinematic skeleton setup in an interactive way. Third, I create a virtual camera in first person perspective using Kinect data to track user's head position and orientation.

Please check the video for better illustration.

Future Work

Retargeting, Rigid body hand and foot, Ground constraint by setting ground plane for food constraint, Human body clothes and skin 3D points reconstruction in Maya, Python for Maya with Kinect rgb-depth data (*11), add more Kinect sensor, and add more than 1 people are interesting future works.

Acknowledgements

Yasutoshi Mori, MirageWrks@gmail.com
MayaOpenNIServer_02Alpha
http://d.hatena.ne.jp/Mirage/20110104/p1

References

1. Kinect Hacks: http://kinecthacks.net/
2. DragonBall (Kamehameha):http://www.youtube.com/watch?v=lajP7HO7ZyE
   Ultra Seven: http://www.youtube.com/watch?v=RUG-Uvq-J-w
   MikuMikuDance: http://www.youtube.com/watch?v=bQREhd9iT38
3. All Nighter: http://www.youtube.com/watch?v=xv9yFkEP6qI
   Film Studio: http://woodman.kusos.com/2011/01/bbs-kinect-for-maya.htm
4. iPiSoft
   http://ipisoft.com/en/wiki/index.php?title=Quick_Start_Guide_with_MS_Kinect_Sensor
5. Brekel Kinect for Motion Builder http://www.brekel.com/
6. Interactive 3D Camera: http://vimeo.com/18149872
7. OpenNI: http://www.openni.org/
8. Readme
9. PyMEL http://code.google.com/p/pymel/
10. EON Reality Interact http://www.youtube.com/watch?v=a9uPx05yYx4
11. Python for Maya with Kinect rgb-depth data
    http://folio.heylight.com/#958712/Code