

 kim hanwoong 내용 작성함.185a4b1 8 days ago

1 contributor

215 lines (172 sloc) | 7.42 KB

Alembic

Alembic은 3D 데이터를 교환하기 위해서 만들어진 Framework입니다. 소니이미지웍스와 ILM이 같이 개발하고 있습니다.

아래 상황에서 자주 사용합니다. 발췌 : <http://www.alembic.io>

- 애니메이션 데이터를 베이크하여 라이팅, 렌더링 할 때
- 애니메이션된 크리처 데이터에서 옷이나 피부를 시뮬레이션 할 때
- 옷, 피부 데이터를 저장하여 라이팅, 렌더링 할 때
- 애니메이션된 물체를 물리엔진으로 넘길 때
- 물리엔진에서 연산된 데이터를 라이팅, 렌더링으로 넘길 때

마야같은 툴에서 이미 abc 파일을 export 할 수 있습니다. 또한 우리가 배울 nuke에서는 abc파일을 import 할 수 있습니다.

abc를 지원하는 프로그램

- Maya
- Katana
- Houdini
- 3dsmax
- Blender
- Unreal
- Nuke
- Gaffer

Alembic 컴파일하기

소스코드 : <https://github.com/alembic/alembic>

abc는 Ogawa, Hdf5 두가지 포맷을 지원합니다. 교육의 편의성을 위해서 소스코드를 이용하여 Ogawa(기본값) 형식만 지원하도록 컴파일 하 겠습니다. Ogawa 파일은 HDF5 보다 용량이 작고 더 빠르게 로드됩니다. HDF5는 예전에 abc에서 사용하던 방식입니다. IlmBase 라이브러 리는 컴파일 중간에 사용되며 OpenEXR 컴파일 할 때 이미 생성한 데이터를 사용할 예정입니다.

참고 : 블렌더는 HDF5로 저장됩니다.

컴파일중간에 libatomic 에러가 발생합니다. 미리 필요한 라이브러리를 설치해둡니다.

```
# yum install devtoolset-6-libatomic-devel
```

```
$ cd ~/app
$ git clone https://github.com/alembic/alembic alembic_src
$ alembic_src
$ git tag
$ git checkout 1.7.10
$ cd ..
$ mkdir alembic_build
```

```
$ mkdir alembic
$ cd alembic_build
$ scl enable devtoolset-6 bash
$ ~/app/cmake-3.13.3/bin/cmake ../alembic_src -DILMBASE_ROOT=$HOME/app/IlmBase -DALEMBIC_SHARED_LIBS=OFF
$ make -j2
$ make install
```

컴파일이 되면 아래 명령어를 사용할 수 있게 됩니다.

- abcdiff

2개의 abc 파일을 입력해서 다른 부분만 abc파일로 출력됩니다. 다른 부분이 없다면 `No differences detected` 메시지가 출력됩니다.

```
$ abcdiff input1.abc input2.abc output.abc
```

- abcecho

.abc파일의 디테일한 정보를 볼 수 있습니다.

```
$ abcecho box.abc
```

```
AbcEcho for Alembic 1.7.10 (built Dec  5 2018 17:30:01)
file written by: Blender
using Alembic : Alembic 1.7.8 (built Sep 10 2018 16:59:10)
written on : Wed Dec  5 19:56:39 2018
user description : untitled

ScalarProperty name=.childBnds;interpretation=box;datatype=float64_t[6];arraysize=6;numsamps=250
Object name=/Cube
CompoundProperty name=.xform;schema=AbcGeom_Xform_v3
ScalarProperty name=.inherits;interpretation=;datatype=bool_t;arraysize=1;numsamps=250
ScalarProperty name=.ops;interpretation=;datatype=uint8_t;arraysize=1;numsamps=250
ScalarProperty name=.vals;interpretation=;datatype=float64_t[16];arraysize=16;numsamps=250
ScalarProperty name=visible;interpretation=;datatype=int8_t;arraysize=1;numsamps=250
Object name=/Cube/CubeShape
CompoundProperty name=.geom;schema=AbcGeom_PolyMesh_v1
ScalarProperty name=.selfBnds;interpretation=box;datatype=float64_t[6];arraysize=6;numsamps=1
ArrayProperty name=P;interpretation=point;datatype=float32_t[3];arraysize=8;numsamps=1
ArrayProperty name=.faceIndices;interpretation=;datatype=int32_t;arraysize=24;numsamps=1
ArrayProperty name=.faceCounts;interpretation=;datatype=int32_t;arraysize=6;numsamps=1
CompoundProperty name=.userProperties;schema=
ScalarProperty name=meshtype;interpretation=;datatype=bool_t;arraysize=1;numsamps=1
CompoundProperty name=.arbGeomParams;schema=
CompoundProperty name=uv;schema=
ArrayProperty name=.vals;interpretation=vector;datatype=float32_t[2];arraysize=14;numsamps=1
ArrayProperty name=.indices;interpretation=;datatype=uint32_t;arraysize=24;numsamps=1
ArrayProperty name=N;interpretation=normal;datatype=float32_t[3];arraysize=8;numsamps=1
```

- abcechobounds

.abc 파일 내부 바운딩 박스의 사이즈 크기를 반환하는 명령어 입니다.

```
$ abcechobounds box.abc
```

Output:

```
/Cube/CubeShape (-1 -1 -1) (1 1 1)
/ (-1 -1 -1) (1 1 1)
```

- abcls

.abc 파일 내부에 있는 오브젝트 이름을 반환합니다.

```
$ abcls box.abc
```

Output:

```
Cube
```

- abcestitcher

각 프레임마다 연산된 .abc 파일을 묶어서 하나의 .abc파일로 만들어줍니다. 프레임영역이 겹치면 `ERROR: overlapping frame range between` 에러가 발생합니다. 보통 개별로 bake된 시뮬레이션 파일을 합칠 때 자주 사용됩니다.

```
$ abcestitcher output.abc input.0001.abc input.0002.abc input.0003.abc ...
```

- abctree

.abc 파일의 구조를 Tree 형태로 그려주는 명령어 입니다.

```
$ abctree box.abc
```

Output:

```
ABC
  --Cube
    --CubeShape
```

abc뷰어

3D 소프트웨어가 없더라도 .abc 파일을 볼 수 있다면 편리합니다. 그 역할을 하는 abcvview를 컴파일해보겠습니다.

- usd를 컴파일하면 usdview가 설치됩니다. usdview에서는 abc파일도 같이 볼 수 있습니다.
- Gaffer 를 설치해도 abc를 볼 수 있습니다.
- 뉴크에서도 abc를 로딩하여 데이터를 확인할 수 있습니다.

```
$ usdview input.abc
```

boost 컴파일

VFX Platform CY2019 기준 Boost는 1.66을 사용합니다. abcvview를 컴파일 하기 위해서는 Boost가 필요합니다.

오래걸림. 수업 초반에 진행할 것

```
$ cd ~/app
$ wget https://dl.bintray.com/boostorg/release/1.66.0/source/boost_1_66_0_rc2.tar.gz
$ tar -zxvf boost_1_66_0_rc2.tar.gz
$ mv boost_1_66_0 boost_1_66_0_src
$ mkdir boost_1_66_0
$ cd boost_1_66_0_src
$ ./bootstrap.sh --prefix=$HOME/app/boost_1_66_0
$ ./b2 install
```

OpenGL 설치

```
# yum install freeglut-devel
```

Python library 설치

```
# yum install python-devel
```

abcvview 컴파일 (아직 컴파일되지 않음.)

참고 : usdview에는 이미 abc파일을 볼 수 있습니다. 개인적으로 usd가 컴파일이 더 잘되어서 abcvview 대신 usdview를 사용하고 있습니다.

소스코드 : <https://github.com/alembic/abcvview>

컴파일이 아직 되지 않는다. 기존에 컴파일한 리스트중 다 활성화 하여 컴파일 해보기.

```
$ cd ~/app
$ git clone https://github.com/alembic/abcvview abcvview_src
$ mkdir abcvview
$ mkdir abcvview_build
$ cd abcvview_build
$ scl enable devtoolset-6 bash
$ cmake ../abcvview_src -DBOOST_ROOT=$HOME/app/boost_1_66_0 -DILMBASE_ROOT=$HOME/app/IlmBase -DALEMBIC_LIBRARIES=$HOME/app/alembic -DPYTHON_ROOT=/usr/lib64/python2.7/config -DPYTHON_LIBRARY=/usr/lib64/python2.7/config/libpython2.7.so -
$ make -j2
$ make install
$ make help
```

실습

- 3D툴을 이용해서 abc 파일 제작.
- 뉴크에 로딩해보기
- 명령어 실습
- centos/script에 필요한 코드 설치 자동화 추가하기