

Akiraの手引書

中村貴英

平成 23 年 5 月 26 日

はじめに

可視化ツール Akira のマニュアルです。この文章は中村貴英によって書かれました。よって文責は中村貴英にあります。

目次

はじめに	i
第 1 章 Akira とは	1
1.1 Akira の構成	1
1.2 Akira の遍歴	2
第 2 章 インストール方法	3
2.1 Mac OSX/Linux の場合	3
2.1.1 準備	3
2.1.2 JAVA のインストール	4
2.1.3 JOGL のインストール	4
2.1.4 Akira.jar のインストール	4
2.1.5 シェルスクリプトのインストール	4
2.1.6 動作確認	4
2.2 Windows の場合	5
2.2.1 準備	5
2.2.2 JAVA のインストール	5
2.2.3 JOGL のインストール	5
2.2.4 Akira.jar のインストール	5
2.2.5 バッチファイルのインストール	6
2.2.6 動作確認	6
第 3 章 Converter の詳細	7
3.1 Akira フォーマットについて	7
第 4 章 Viewer の詳細	8
4.1 機能一覧	8
4.2 Key 操作	10
第 5 章 その他	12
5.1 ムービーを作る	12
5.1.1 ImageMagick について	12
5.1.2 QuickMovie	12
5.2 故障かなと思ったら	12

	iii
5.3 要望, 不満, バグ	12
付 録 A イースターエッグ	13
A.1 エンジョイモード	13

第1章 Akira とは

Akira とは計算機シミュレーションの結果を可視化するソフトウェアです。もともとは尾形先生の先輩である中野先生の研究室で、C 言語と OpenGL(glut) を使って開発された atoms viewer というソフトウェアが起源となっています。この atoms viewer を河野さんが JAVA に移植し KVS(Kouno Viewer System) の名前でしばらく使われていました。それをもとに中村がフルスクラッチで書き直し、現在に至ります。現在は、名古屋工業大学の尾形研の有志によって改良・保守されています。

また Akira は JAVA と OpenGL によって作られています。これは

1. JAVA が、プラットフォームに依存せず実行できること
2. JAVA が、多人数での開発に向いていると言われているオブジェクト指向を体現した言語であること
3. OpenGL が、描画ライブラリとして一般的で情報が得やすいこと

を理由としています。

1.1 Akira の構成

Akira では、描画時の高速化の為にデータを一旦 Akira 専用のバイナリに変換します。その為 Akira は Converter と Viewer の二つからなります。

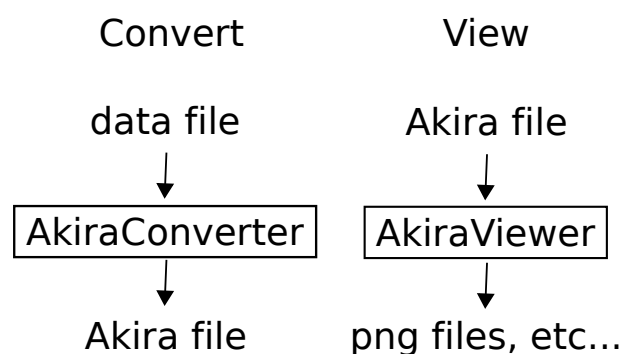


図 1.1: Akira で可視化するまで

AkiraConverter

描画に必要なデータのみを集約し、バイナリファイルとして保存します。読み込みファイルのフォーマットは [Sec.3.1](#) を参照してください。

AkiraViewer

AkiraConverter で変換された Akira ファイル (拡張子は.Akira) を読み込んで表示します。Viewer の設定ファイルは、`~/.Akira/` または `C:\Akira` に保存されます。

1.2 Akira の遍歴

2008 以前 atomviewer が使われる

2008-12-04 KVS 開発スタート

2009-07-08 KVS 開発終了。KVS2 開発スタート

2010-11-28 外部公開に向けて、KVS から Akira へ改名。マニュアル整備。

2010-12-26 ホスティングサーバーを Google Project Hosting へ。

第2章 インストール方法

Akira は JOGL(<http://jogamp.org>) と JAVA があれば動きます。JOGL とは OpenGL を JAVA から使う為のライブラリです。JAVA は大抵の環境でインストール済みと思われるので、JOGL のみ新たにインストールする必要があります。Akira は jar ファイル (JAVA の実行形式) で配布されます。Akira の最新版は <http://code.google.com/p/project-akira/downloads/list> からダウンロード可能です。インストールは

- JOGL の設置、および設置したディレクトリに PATH を通す
- Akira.jar の設置
- Akira.jar を呼び出すスクリプトの設置

を行うのみです。アップデートがある場合、Akira.jar を入れ替えるのみで、JOGL を入れ替える必要はありません。

2.1 Mac OSX/Linux の場合

このセクションでは JOGL を `~/myLocal/javaliib/jogl` に、Akira.jar を `~/myLocal/Akira/` に置くとして話を進めます。¹

2.1.1 準備

以下を実行してディレクトリを作っておきます。

```
mkdir ~/myLocal
mkdir ~/myLocal/bin
mkdir ~/myLocal/javaliib
mkdir ~/myLocal/javaliib/jogl
mkdir ~/myLocal/Akira
mkdir ~/.Akira
```

また以下を `.bashrc` に書き込みます。

¹これらのディレクトリが気に入らない上級者は、自由に変更して下さい。ただし Akira の設定ファイルの置き場所は、`~/.Akira/` で固定なので注意して下さい。

```
export JOGL_LIB=~myLocal/javallib/jogl
export DYLD_LIBRARY_PATH=$JOGL_LIB:$DYLD_LIBRARY_PATH
export LD_LIBRARY_PATH=$JOGL_LIB:$LD_LIBRARY_PATH
export PATH=$PATH:~/myLocal/bin
```

また以下を.bash_profile

```
source ~/.bashrc
```

に書きこんで、ターミナルを再起動して、設定を反映させます。

2.1.2 JAVA のインストール

Mac OSX の場合、標準で JAVA がインストールされています。むしろ、画面左上リンゴマークソフトウェアアップデートで最新版にしてください。Linux を使う人はエキスパートと思われるので、各自インストールを行ってください。

2.1.3 JOGL のインストール

<http://download.java.net/media/jogl/builds/archive/jsr-231-2.0-beta10/> から JOGL ² をダウンロードします。解凍し、lib の中身を~/myLocal/javallib/jogl に置いて下さい。³

2.1.4 Akira.jar のインストール

Akira.jar を~/myLocal/Akira/に置いて下さい。⁴

2.1.5 シェルスクリプトのインストール

自分のプラットフォームにあわせた shell-script-***の中身を~/myLocal/bin/に置きます。

2.1.6 動作確認

<http://project-akira.googlecode.com/files/sample.zip> から sample.zip をダウンロード、解凍してその中で以下を実行します。

² 64bit CPU の Linux の人は jogl-2.0-linux-amd64.zip, 32bit CPU の Linux の人は jogl-2.0-linux-i586.zip, Mac OSX の人は jogl-2.0-macosx-universal.zip です。

³ OSX の場合は環境変数 `DYLD_LIBRARY_PATH`, Linux の場合は環境変数 `LD_LIBRARY_PATH` に ~/myLocal/javallib/jogl を設定します。が、準備でこれらの設定は済んでいます。

⁴ Akira のアップデートは、~/myLocal/Akira/にある Akira.jar を入れ替えることで完了します


```
Akira.sh
```

問題なく表示されれば、インストールは完了です。うまく表示されない場合はエラー文を参考に問題解決してください。

2.2 Windows の場合

windows において、Akira の設定ファイルは C:Akira に保存されます。故にディレクトリ C:Akira は必須です。windows では、Akira.jar も C:Akira に置く事を推奨します。尚、JOGL は C:jogl に置くとして話を進めますが、JOGL の置き場所は自由です。

また windows に疎い人間が書いている文章なので、もっとスマートな設定方法があれば教えてください。

2.2.1 準備

C ドライブの直下に jogl および Akira という名前のフォルダを作っておいて下さい。

2.2.2 JAVA のインストール

JRE をインストールしてください。

2.2.3 JOGL のインストール

まず C ドライブの直下に jogl というフォルダを作ります。そして <http://download.java.net/media/jogl/builds/archive/jsr-231-2.0-beta10/> から JOGL ⁵ をダウンロードします。解凍して、lib の中身を C:jogl に置いて下さい。

更にコントロールパネル-システム-詳細設定-環境変数にて、PATH に

```
C:jogl
```

を追加します。

2.2.4 Akira.jar のインストール

C ドライブの直下に Akira というフォルダを作ります。そして Akira.jar を C:Akira に置いて下さい⁶。Akira の設定ファイルを C:Akira に保存する様にプログラムされている為、このディレクトリが無ければ正常に動作しません。

⁵64bit CPU を使っている人は jogl-2.0-windows-amd64.zip, 32bit CPU を使っている人は jogl-2.0-windows-i586.zip です。

⁶Akira のアップデートは、C:Akira にある Akira.jar を入れ替えることで完了します

2.2.5 バッチファイルのインストール

好きなところへ置いて下さい。

2.2.6 動作確認

<http://project-akira.googlecode.com/files/sample.zip> から sample.zip をダウンロード，解凍してその中で以下を実行します。

```
AkiraConverter.bat  
AkiraViewer.bat
```

第3章 Converterの詳細

現在, AkiraConverter では

- Akira フォーマット
- CHGCAR
- Gaussian cube

を読み込むことができます.

3.1 Akira フォーマットについて

Akira フォーマットを出力するサンプルコードを Fig.??および Fig.??に示します. これらは tar.gz¹ tar.bz2² で形式で圧縮されているならば, 解凍する事無く読み出しが可能です.

¹tar czvf a.tgz out*

²tar cjvf a.tbz2 out*

第4章 Viewerの詳細

AkiraConv にて変換された，専用の Akira ファイル (拡張子は.Akira) を読み込んで表示します。

4.1 機能一覧

原子を点，球で描く

最も早い描画は点で描くことです。また最終的には球で描くと思います。その場合は，緯度と経度方向のスライス指定して球の滑らかさを決定します。もちろんスライスが増えるに連れて，描画は遅くなります。それらは atoms パネルで指定します。またそれぞれの原子が持つデータを基に，色を決定するモードもあります。

ボンドを線，円筒で描く

これもやはり早い描画は線で描くときです。

ベクトル

任意のベクトル量を矢印として描画出来ます

ライトの設定

背景色，文字色などの設定

印刷する場合は背景色を白にした方が，インクの節約になるでしょう。

View point の保存

現在の視点を保存することが可能です。次回の Akira 起動時に同じ視点が再現出来ます。

静止画像として出力 (png,jpg,bmp)

デフォルトでは png 形式で出力します。各フレームを画像ファイルに出力し，連結すればムービーが出来上がります。

任意の面でカット可能

任意の面でカットして、断面を見ることが可能です。面の指定は、その法線ベクトルと面が通る点を指定することで行います。

Volume Rendering

additional data は原子一つ一つのデータであるのでバラツキが大きく、なんらかの平均をしなければ、見るに耐えません。しかし、滑らかな描画の為には高い解像度が必要です。この相反する要求を満たすために、まず平均用のメッシュ(data mesh)で additional data を平均します。そして描画用のメッシュ(draw mesh)を data mesh から生成し、draw mesh に基づいてデータを描画します。

等値面

指定したデータを持つ面を描きます。

等高線の2次元射影

指定した面の情報を二次元に射影します。一応等高線もかけますが、ファイルに出力して gnuplot で等高線を描くのがよいでしょう。

動径分布関数の計算

動径分布を見ます。ファイル出力にも対応しているので、簡易解析にどうぞ。

averager

現在表示されている原子のデータを単純平均します。

コンボ

定型処理を登録しておくと、自動的に処理出来ます。

special export

eps ファイル、POV-ray ファイルで出力できるように努力中です。

ピック

情報

ピックされた原子の情報を出力します.

Trajectory 表示

選んだ原子の軌跡を描きます.

2点ピックで距離計算

連続してピックすると, その二点間の距離を計算します.

3点ピックで角度計算

連続して3点ピックすると, 3点で作られる角度を計算します.

平面の表示

任意の面をかけます.

4.2 Key 操作

AkiraViewer 実行時にも Infomation Window に key の使用方法が表示されます.

表 4.1: List of parameters only for NIT version.

操作	効果
esc	終了
arrows	回転
alt+arrows	z 軸回転
shift+arrows	平行移動
meta+arrows	少量回転
meta+shift+arrows	少量平行移動
0	原子番号に基づいた原子色
1,2,3,...	i 番目のデータに基づいた原子色
alt+a	Axis の表示非表示
shift+a	tics タイプの変更
alt+b	box の表示非表示
shift+b	bond タイプの変更
b	bond の表示非表示
alt+c	コンボモードの起動
shift+c	アルファタイプの変更

table continued on next page

continued from previous page

操作	効果
c	カラータイプの変更
shift+d	fps を落とす
d	fps を上げる
alt+h	save view-point
shift+h	revert view-point
h	revert home view-point
i	ViewConfigWindow を手前に
shift+l	ラベルタイプの変更
l	ラベルの表示非表示
shift+m	トラックボールモード
m	平行投影/透視投影の切替え
n	次のフレームへ
meta+o	開く
p	1 フレーム戻る (previous frame)
alt+r	動径分布
shift+r	原子の表示非表示
r	原子タイプの変更
s	アニメーションのスタート, ストップ
alt+t	時間の表示非表示
shift+t	legend の表示非表示
t	軌跡モード
shift+v	ベクトルタイプの変更
v	ベクトルの表示非表示
w	静止画出力
W	連続的に静止画を出力 (動画用)
x	x 軸から見る
y	y 軸から見る
shift+z	zoom out
z	zoom in
左マウスドラッグ	回転
右マウスドラッグ	平行移動
中央マウスドラッグ	拡大, 縮小
マウススクロール	zoom in/out

第5章 その他

5.1 ムービーを作る

Wを押すと、全フレームをpngで出力されます。これをなんらかのソフトで結合すればムービーの出来上がりです。例えば以下のソフトがあります。

image magick コマンドラインツール

QuickMovie MacOSX 用 (シェアウェア)

以上で紹介したソフト以外にも、画像を結合して動画を作るソフトは無数に有るので各自で素敵なソフトを探してください。

5.1.1 ImageMagick について

<http://www.imagemagick.org/script/index.php>. GimpかInkscapeについてくるMacならMacPorts, Linuxならyumやapt-getでインストールできます。WindowsでもGimpをインストールすると、自動的にインストールされるので、使えると思います。

実際にムービーを作る場合、例えば以下のコマンドです。

```
convert -delay 20 *.png a.mpg
```

5.1.2 QuickMovie

尾形先生愛用!?のソフト。たしかに直感的で使いやすいです。

5.2 故障かなと思ったら

一先ず、~/Akira/またはC:Akiraにある設定ファイルを消して下さい。

5.3 要望, 不満, バグ

バグがあったらエラー文とともに教えてください。

付 録 A イースターエッグ

A.1 エンジョイモード

「モード反転, 裏コード ザ・ビースト」なんてのがあるんです (元ネタわかりますか?).

```
java -Xmx1024m -Xdock:icon=$AkiraDIR/Akira.icns -Xdock:name=Akira \  
-cp $AkiraDIR/Akira.jar viewer.AkiraView -enjoy
```

のように`-enjoy`をオプション引数にして実行してみてください. いろいろな遊び要素/実験機能が追加されたモードで起動します.