

サルでもわかる i_rec マニュアル Ver.2.0.2

これは、ド素人がとりあえず i_rec を動かせるようになるまでの記録に基づいたマニュアルです。

Linux についての知識がゼロ、工学的なハードの知識もない、Windows と Mac ならば事務用などに使っているが、プログラミングなんてとんでもない。というヒトがなんとか i_rec を使っておサルの視線計測ができるようになるまでになった・・・という記録です。

いろいろ試行錯誤し、いろいろ躓きましたが、現在はなんとか i_rec が動いています。

i_rec の作者である産総研の松田さんには、何度もばかばかしい質問をして、お手数をおかけしました。この場を借りて御礼申し上げます。

詳しいことは判りませんが、「とりあえずこうすれば動くよ」というメモ書き程度にお読みください。

なお、このマニュアルは i_rec ver2.82 用に作成してあります。他のバージョンでは、多少異なる部分があります。

ここに記載してある方法は、富山大学大学院医学薬学研究部システム情動科学で行った方法であり、必ずしもこの方法で動作するとは限りませんので、参考程度に見て頂ければと思います。

目次

1. 機器を用意する。
 - 1) パソコン
 - 2) CCD カメラユニット
 - 3) レンズ
 - 4) 赤外線 LED
 - 5) カメラステイ
 - 6) 各種ボード
 - 6-1) ビデオキャプチャボード
 - 6-2) デジタル入出力ボード
 - 6-3) デジタル・アナログ変換ボード

2. Vine Linux をインストールする。
 - 2-1) パソコンにボード類を全部挿します。
 - 2-2) Vine Linux 3.2 の入手
 - 2-3) Vine Linux のインストール
 - 2-4) Vine Linux 3.2 の起動
 - 2-5) ビデオキャプチャボードの確認

3. i_rec 用のフォルダを作成する。

4. i_rec をダウンロードする。

5. i_rec をインストールする。

6. i_rec を自分の環境に合わせてカスタマイズする。
 - 6-1) i_rec に含まれるファイルの編集
 - 6-2) Video for Linux の編集
 - 6-3) i_rec のコンパイル

7. i_rec を使用してみる。
 - 7-1) カメラと赤外線 LED の位置関係など
 - 7-2) i_rec のセッティング

8. キャリブレーションを行う。

- 8-1) キャリブレーションのための視標準備
- 8-2) 視標と被験者の位置関係
- 8-3) 視標の数値化
- 8-4) Object Map ファイル
- 8-5) サルに視標を注視させ、キャリブレーションを取る。

9. 外部機器への出力確認をする。

- 9-1) DA ボードからの出力
- 9-2) D I/O ボードからの出力

10. i_rec メニュー一覧

11. i_rec のセットアップ時に使用するターミナルエミュレータコマンド一覧

1. 機器を用意する。

必要な機器は、松田さんの Web ページに記載されているとおりです。

<http://staff.aist.go.jp/k.matsuda/eye/>

ちなみに、現在我々が使用している機器は以下のとおり。これ以上のスペックのものを用意しましょう。

1) パソコン

CPU : Pentium III, 500MHz

RAM : 256MB

HDD : 10GB

PCI 空きスロット : 3つ以上

の手作りパソコン。

*我々は、OS は後で入れるので、OS なしで注文しました。でも、とりあえず Windows が入っていれば、ハードウェアの動作確認がしやすいので、Windows を入れておいたほうが無難なようです。

2) CCD カメラユニット

ソニー B&W Camera Module XC-75

3) レンズ

ソニー 45 mm マクロレンズ VCL-M45YM

*赤外線フィルタ（赤っぽいガラス板）が入っているので、はずして使います。

4) 赤外線 LED

パーツ屋さん（富山には無線パーツしかない）においてある赤外線 LED。

5V 仕様であるが、そこに 12V 入れて 2 個並列でつかっちゃってます。

850nm くらいの赤外線 LED であれば結構な照度が得られます。

12V 仕様のものもあります。

5) カメラステイ

（有）メルクエストにこしらえてもらいました。

6) 各種ボード

6-1) ビデオキャプチャボード

カメラからの画像を取り込むのに使います。

我々は、Artray の ART-878 を使用しています。

が、市販はされていない模様です。

パソコン機器を扱っている代理店に相談しましょう。

（富山では、ワキタさんが持ってきてくれます。）

6-2) デジタル入出力ボード

外部機器（タスク制御用パソコン）との信号（TTL）制御に使います。

CONTEC の PIO-16/16L(PCDH)

このボードは負理論を使用しているため、ON-Low, OFF-High です。

TTL 出力を行うためには、ソフト的あるいはハード的にインバートさせる必要があります。

6-3) デジタル・アナログ変換ボード

目の座標や瞳孔径をアナログデータとして出力します。

CONTEC の DA12-4(PCI)

2. Vine Linuxをインストールする。

2-1) パソコンにボード類を全部挿します。

パソコンの蓋を開けたことがない場合には、そういう経験のあるヒトにやってもらい、必ず脇で見てください。

簡単ですから、すぐに覚えられます。

また、ボード類には金色の楕円のような部分があります。

この部分はとても大事なので、素手で触れてはいけません。

使用するボードはいずれもPCIと呼ばれるスロットに挿すものです。

PCIのスロットは白ですから、すぐにわかると思います。

2-2) Vine Linux 3.2 の入手

Linux (リナックス、ライナックス、リヌクス等と読むらしい) を使ったことが無くても大丈夫です。

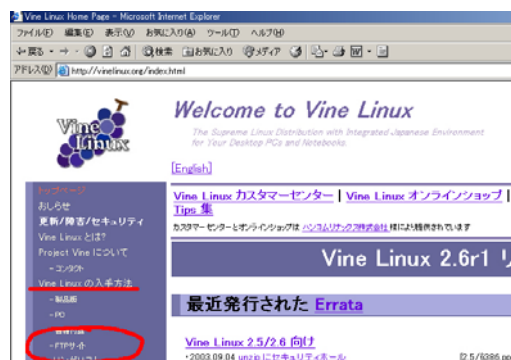
まずは、WindowsでもMacでもいいので、普段使用しているパソコンで、CD-RもしくはCD-RWを搭載しているマシンから次のWebサイトへアクセスしてみましょう。

Linuxに関する情報がいっぱい・・・ <http://www.linux.or.jp/>

Vine Linuxの情報がいっぱい・・・ <http://vinelinux.org/index.html>

多分、何が何やら判らないと思います。

とりあえず、OSとしてVine Linuxと言うのを入れますので、上記下段のWebサイトへアクセス。



i_recは、2008年8月現在、Vineのver 4以降に対応しておりません。必ず、ver 3をインストールしてください。

左側にVine Linuxの入手方法というのがあるので、そこに記載されている適当なftpサイトを開くと、Vine 3.2 というフォルダがあります。その中に、CDIMAGE

というフォルダがあり、さらにその中にVine32-i386.isoと言うのがあります。この、Vine32-i386.isoをダウンロードして、CDに焼きましょう。

2-3) Vine Linuxのインストール

インストールする前に、インストール方法を記載したWebサイトを見てみましょう。

<http://www.vinelinux.org/install-guide/index.html>

図が豊富で、判りやすいですね。

起動ディスクを作成するフロッピーやIPアドレス、モニタの形式など、準備に必要な情報をまとめておきましょう。

まずは、ボードを挿したパソコンの電源を入れ、Vine Linuxの入ったCDを入れてみま

す。CDから勝手にVineのインストーラが立ち上がればラッキーです。

もし、立ち上がらなかつたら、BIOSというのをいぢる必要があります。

BIOSをいぢると、CDから立ち上げることができるようになります。

パソコンを良く知っているヒトに、設定してもらいましょう。

その際も、隣でよく見ておきましょう。

BIOSは、下手にいぢるとパソコンの調子が悪くなりますが、変なことをしなければ大丈夫です。

CDから立ち上がったら、<http://vinelinux.org/manuals/install-vine.html>に記載されているとおりに進めます。

しかし、注意があります。

マウスのボタンは、Macでは1つ、Windowsでは2つですが、Linuxでは3つです。でも、2つボタンのマウスでも大丈夫です。「3 ボタンエミュレーション」をチェックしておけば左右のボタンを同時にクリックすることで、真ん中ボタンを押したものとして扱われるようになります。

インストールする条件を開発者かワークステーションかを聞いてくるので、必ず「開発者」あるいは「全部インストールする」を選択しましょう。

ワークステーションを選択すると、後でi_recを動かさせません。

また、テキストモードとグラフィックモードを選べますので、グラフィックモードを選択しておいたほうが初心者にはわかりやすいです。

ユーザの設定もしておきましょう。

普段から全部の権限をもったroot状態で作業するのは、好ましくありません。

我々は、rootのほかにmonkeyというユーザ名を作成し、普段はそれを使用しています。

2-4) Vine Linux 3.2CRの起動

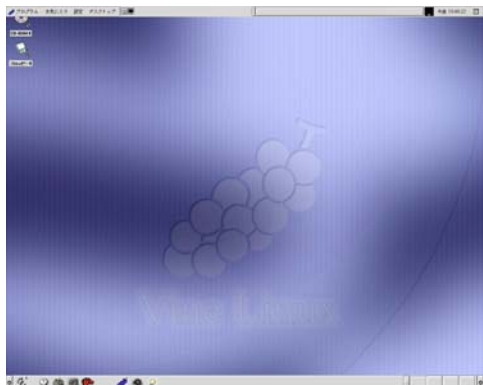
さて、インストールが終わったら、CDを抜いてパソコンを再起動してみましょう。

Vine Linuxには、ver3.2 とver3.0 が両方入っています。

どちらを立ち上げるか選べる画面が出てきますが、ver3.2 を使用します。

放って置けば、ver3.2 が立ち上がります。

取り敢えずはrootではなく、先ほど設定したユーザ名とパスワードで立ち上げてみます。



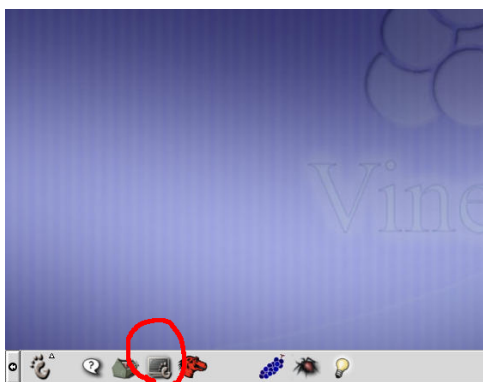
Vine Linuxは立ち上がったでしょうか？

もし、上手く立ち上がらないときには、<http://vinelinux.org/index.html> へアクセスして原因を調べてみましょう。

2-5) ビデオキャプチャボードの確認

ここでは、ビデオキャプチャボードをVine Linuxが認識しているかどうかを確認します。

Vine Linuxが立ち上がった状態ですと、下のほうに左図のようなアイコンがあるはずですが。



これは、ターミナルエミュレータと言い、コマンド（命令文）を入力してパソコンを動かすモノです。

巻末に、i_recのセットアップに使用するコマンド一覧を載せてありますので、興味のある人は参考にしてください。

これをたちあげて、

`lspci`

と入力してみます。

ここで注意。今後、このマニュアル内でタ

ーミナルエミュレータ上での文字入力には、

```
% lspci
```

の様に頭に%をつける事とします。

```
% lspci
```

の結果、

```
00:0C.0 Multimedia video controller: Brooktree Corporation Bt878 (rev 11)
```

```
00:0C.1 Multimedia controller: Brooktree Corporation Bt878 (rev 11)
```

というメッセージが確認できればボードがちゃんと刺さっていることを意味します。

もし、上記メッセージが確認できないときには、`lspci`の存在している場所に、パスが通

っていないためです。

```
% /sbin/lspci
```

としてみてください（もしかしたら、`/usr/sbin/lspci`かもしれません）。

また、

```
% ls -la /dev/video*
```

も同様に試してみましょう。

```
lrwxrwxrwx 1 root root 6 Apr 30 2003 /dev/video/video ->video0
```

```
crw----- 1 root root 81, 0 Jan 29 2000 /dev/video0
```

```
crw----- 1 root root 81, 1 Jan 29 2000 /dev/video1
```

日付部分は異なるでしょうが、この様なメッセージが確認できればOKです。

マザーボードによっては、上記のメッセージが確認できない場合があります。

念のため、別のftpサイトからダウンロードしたOSを入れなおしてみましょう。

また、これらのメッセージはデバイスドライバ（ボードを動かすためのソフトのようなもの）がインストールされていることを意味するものではありません。

そこで、`xawtv`というソフトをインストールすると、`/dev/video0`などのデバイスファイルがない場合は作成されるうえに、`video for linux`の動作確認も出来ます。

もし、上記のメッセージが確認できないときには、

- 1) Vine Linuxを別のFTPサイトからダウンロードし、入れなおしてみる。
- 2) `xawtv`というソフトをインストールしてみる。

後者については、以下の手順で行います。

---modules.conf の編集---

`/etc/modules.conf`をエディタで編集し、以下の行を書き足し保存します。

左下の足跡マークから行けるプログラムのアプリケーション内に`gedit`と言うのがありますので、それを立ち上げて`modules.conf`を開いて書き足しましょう。

```
alias char-major-89 i2c-dev
```

```
post-install i2c-dev modprobe "i2c-algo-bit"
```

```
alias char-major-81 bttv
```

```
alias char-major-81-0 bttv
```

---xawtvのインストール---

<http://bytesex.org/xawtv>から`xawtv_3.88.tgz`をダウンロードします。

ダウンロード先は`/usr/local/src`に指定します。

ターミナルエミュレータ上で下記のコマンドを実行し、`/usr/local/src`にてファイルを展開してインストールします。

```
% cd /usr/local/src
```

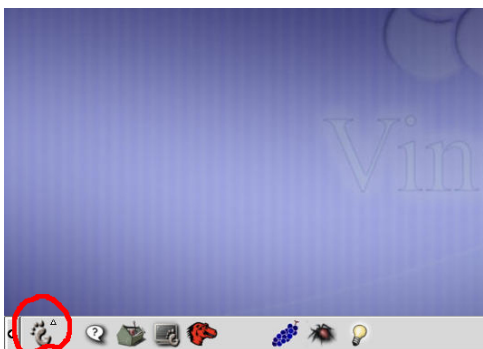
```
% tar zxvf xawtv_3.88.tgz
% cd xawtv-3.88
% ./MAKEDEV.v4l
% ./configure --prefix=/usr/X11R6
% make
% make install
```

xawtvを動かして、v4lが動いているのを確認します。

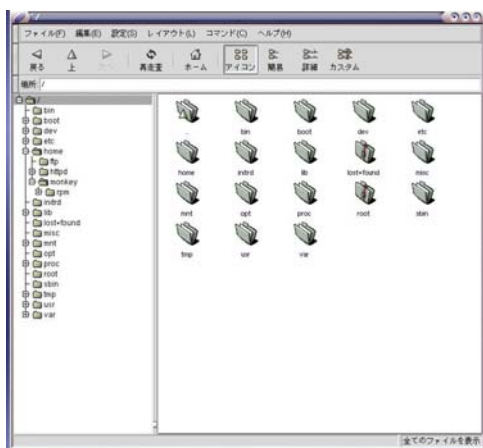
3. i_rec 用のフォルダを作成する。

Windows ユーザーにはわかりやすいと思いますが、Vine Linux にはファイルマネージャと言うのがあります。

左下の足跡マーク (Win で言うところのスタートボタン) をクリックすると、プログラムと言うのが見えます。



そこにカーソルを合わせると、いろいろ出てきますが、ファイルマネージャと言うのを選んで起動します。



Windows のエクスプローラとそっくりですね。左側がディレクトリで、右側がその内容です。

usr の中の local の中の src の中に、i_rec というフォルダを作成してください。

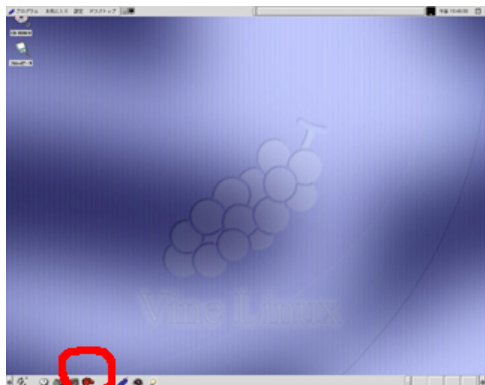
作成方法は、左側で src を選んだ後、右側には src の内容が表示されます。右側で右クリックをして、「新規ディレクトリ」を選びます。

名前を i_rec にすれば OK です。

*後のことを考えると、usr/local/src/i_rec の中にバージョン毎のディレクトリを作成しておいた方がいいでしょう。

4. i_rec をダウンロードする。

Vine Linux には、Mozilla と呼ばれる Web ブラウザが標準でついてきます。



これをご覧の方ならば、どこから i_rec を入手するかはご存知だと思いますので、それは省きます。

ダウンロードするファイルは Video for Linux version 2.82 です。

ダウンロード先は、先ほど作成した `usr/local/src/i_rec/` を選択します。

5. i_rec をインストールする。

ダウンロードした i_rec は、圧縮ファイルです。これを解凍する必要があります。

まず、ターミナルエミュレータを開き、

```
% su
```

と入れます。

root のパスワードを聞かれますので、パスワードを入力します。

次に、

```
% cd /usr/local/src/i_rec/
```

と入力します。

次に、

```
% tar zxvf ver2.82.tar.gz
```

と入力します。

万が一、ISA バスに刺す DA ボードをお使いの場合は、

```
% cd ver2.82/source/i_rec
```

と入力してディレクトリを ver2.82 内の i_rec に移動し、

```
% cp dacprog/contec.da.c da.c
```

と入力します。

```
# 上書きしてもよろしいですか(yes/no)?
```

と聞かれますので、y と入力してリターンします。

とりあえずこれで、i_rec のソースコードの展開ができました。

しかし、このままでは上手く動きません。

自分の環境に合わせて、カスタマイズする必要があります。

ここまでの流れをまとめると、ターミナルエミュレータ上で入力するのは

```
% su
```

```
% cd /usr/local/src/i_rec/
% tar zxvf ver2.82.tar.gz
[ % cd ver2.82/source/i_rec
  % cp dacprog/contec.da.c da.c ]

```

ということになります。

6. i_recを自分の環境に合わせてカスタマイズする。

6-1) i_recに含まれるファイルの編集

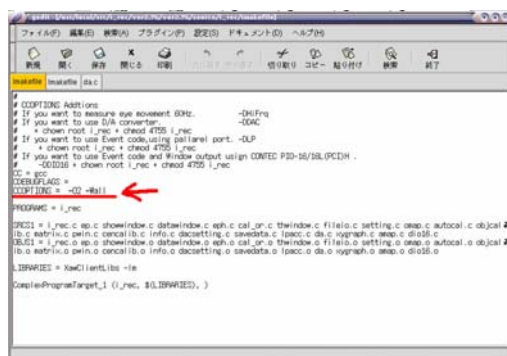
まず、ファイルマネージャを使って
/usr/local/src/i_rec/ver2.82/source/i_rec
という場所へ移動します。

その中に、Imakefileというのがありますから、これを開きます。

ダブルクリックすると、そのまま開いて編集できる状態ではありません。

足跡マークから行けるプログラムのアプリケーション内にgeditと言うのがありますので、それを立ち上げて開いてみましょう。

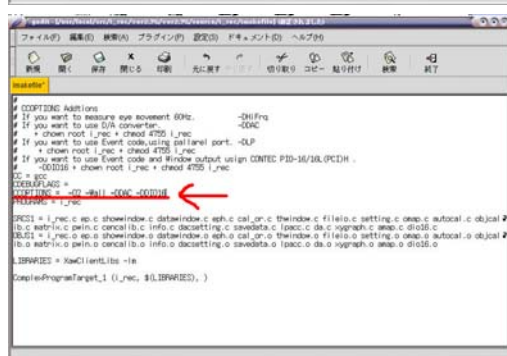
Imakefileの中身を見ると、CCOPTIONという部分があります。



そこに

`-DDAC -DDIO16`

という一文を追加します。



6-2) Video for Linuxの編集

ファイルマネージャを使って

/etc/modules.conf

の存在を確認後、geditでもってこのファイルを開いて、以下のコマンドを追加します。

```
alias char-major-89 i2c-dev
```

```
post-install i2c-dev modprobe "i2c-algo-bit"
```

```
alias char-major-81 bttv
```

```
alias char-major-81-0 bttv
```

これらは、ビデオキャプチャボードをきちんと動かすために必要な記述です。

これらの記述がないと、撮影した目がi_rec上に現れてきません。

6-3) DIO、DAの設定

i_recでは、被験者が任意の場所を見た時にTTL信号を出力させる事ができます。

例えば、モニタ上の注視点をサルを見た時にTTLシグナルを出すようにし、そのシグナルを受けたら課題を開始するようにしておけば、いわゆるFixation課題をさせることが出来ます。

ここでは、CONTECのPIO-16/16L(PCDH)を使用した場合の設定について述べます。

このDIOボードは、負理論を採用しているので、ON時にはLOW、OFF時にはHIGHと なってしまいます。

これを逆転させるために、/usr/local/src/i_rec/ver2.82/source/i_recにあるdio16.cの内容を編集します。

```
void DIO16_Out(int data)
{
    unsigned short int o;
    o=(unsigned short)(data &0xffff);
    outw_p(o,DIO16_ADR+2);
}
```

oのビットを反転させる一行を追加します。

```
void DIO16_Out(int data)
{
    unsigned short int o;
    o=(unsigned short)(data &0xffff);
    o=~o;          ←この一行を追加します。
    outw_p(o,DIO16_ADR+2);
}
```

さらに、デバイスのIDやボードのアドレスを調べて、dio16.cの内容を編集します。

まず、

```
% /sbin/lspci -v
```

と入力します。その結果、なんだか色々则表示されると思いますので、その中から下記のような部分を探してください。

02:09.0 Multimedia controller: Contec Co., Ltd: Unknown device 8183 (rev 01)

Subsystem: Contec Co., Ltd: Unknown device 8183

Flags: medium devsel, IRQ 21 I/O ports at b400 [size=32]

この中で、device **8183** という部分とI/O ports at **b400** という部分が重要です。

dio16.cの以下の箇所を上記の赤字に編集する必要があります。

```
#define DeviceID_DIO 0x9726この下線部分を、上記 8183 に相当する数字と入れ替えます。
```

また、

```
#define DIO16_ADR 0xece0の下線部分を、上記b400 に相当する文字と入れ替えます。
```

*現在我々が使用している*i_rec*は、D I/O出力にちょっとだけ手が加わっています。

*Fixation Task*のために、1500 ミリ秒以上固視しないとTTLの出力が行われないう、松田さんに改造してもらいました。

*i_rec*ディレクトリ内にある*dio16.c*をgEditなどで開くと、

```
#define Duration 45
```

という一文があります。

眼球の画像を得ているフレームレートが1/30 秒なので、上記のDuration 45 は

$1/30\text{sec} * 45 = 1.5\text{sec} = 1500\text{ms}$ となります。

固視させる時間を変えたければ、*dio16.c*内のdefine *Duraion*の数字を変えて保存し、コンパイルしなおします。

このモジュールは、通常の*i_rec*には含まれておりません。必要な方は、松田さんにお問い合わせして下さい。

*我々は、松田さんからDIO16 チャンネルの出力をチェックするプログラムをいただきました。以下に、その使用方法を示します。

圧縮ファイル*dio16test.tgz*を/var/tmpの下にコピーし、ターミナルエミュレータで

```
% cd /var/tmp
```

```
% tar zxvf dio16test.tgz
```

としてファイルを解凍します。すると*dio16test*というディレクトリが出来上がります。

一度gEditなどで解凍したファイルの中にある*dio16test.c*をひらき、*diviceID*の部分を下記の通り書き換えます。

```
#define deviceID_DIO 0x9182 ←この数字をlspci -vで調べた値にする。
```

もう一度ターミナルエミュレータを開き、*dio16test*ディレクトリの中に移動します。

```
% cd /var/tmp/dio16test
```

```
% make
```

```
% su
```

```
% chmod 4755 dio16test
```

とします。これで使用できるので、DIO出力のテストを行うときは、root権限でターミナルエミュレータから

```
% cd /var/tmp/dio16test
```

```
%. /dio16test
```

と入力してテストプログラムを立ち上げます。

何かを入力するたびに、1ビットずつ出力を行います。

DAについても、DIOと同様にボードのアドレスやデバイスIDを設定する必要があります。% /sbin/lspci -vで調べたデバイスIDなどを元に、da.cを編集します。geditなどでda.cを開き、DIOと同様に

```
#define DAC_ADR 0x300この部分と
```

```
#define DeviceID_DAC 0x8186この部分を書き換えます。
```

6-4) i_recのコンパイル

i_recを、実際に使用する形に仕上げます。

ターミナルエミュレータを開いて、まずはディレクトリを変更します。

```
% cd /usr/local/src/i_rec/ver2.82/source/i_rec
```

その後、

```
% su
```

でrootのパスワードを入力して権限を獲得し、

```
% xmkmf -a
```

```
% make
```

と入力します。

これは、インストールしたi_recをコンパイル（実際に使用する形に仕上げる）する命令文です。

すると、いろいろ作業を開始します。

makeが終わったら、

```
% chown root i_rec
```

```
% chmod 4755 i_rec
```

```
% cp i_rec /usr/local/bin
```

と入力します。

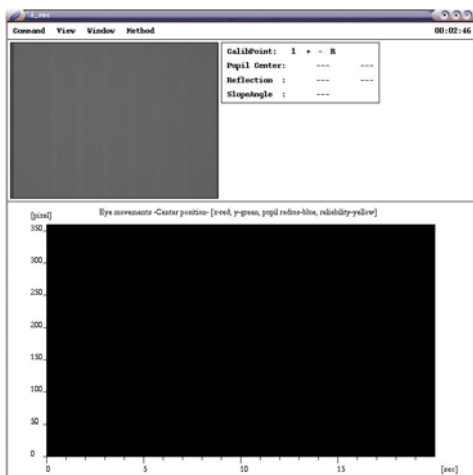
この状態で一度パソコンを再起動してみましよう。

素直にVine Linuxが立ち上がったら、ターミナルエミュレータを開き

```
i_rec
```

と入力てみます。

i_recは起動できたでしょうか？



7. i_rec を使用してみる。

7-1) カメラと赤外線 LED の位置関係など

カメラから取り込んだ画像を元に i_rec を使うためには、いくつかの条件があります。1)サル
の目とカメラの距離

・・・使用するレンズにもよりますが、現在は
110 mm です。

2)サルの目と照明の距離

・・・これもレンズの性能に左右されますが、170 mm で使用しています。

照明は、カメラのレンズの下側に2個取り付けられています。

3)カメラとビデオキャプチャボードの接続

・・・これが間違っていると、当然ですが画像入力がありません。

カメラユニットからのケーブルをビデオカードの1番に接続します。

4) 赤外線 LED の明るさと位置

・・・充分明るくないと、ちゃんと検出してくれません。

また、LED の証明範囲は比較的狭いので、向きも問題になります。

条件をいろいろ変えて、目が上手く撮影できる条件を見つけましょう。

7-2) i_rec のセッティング

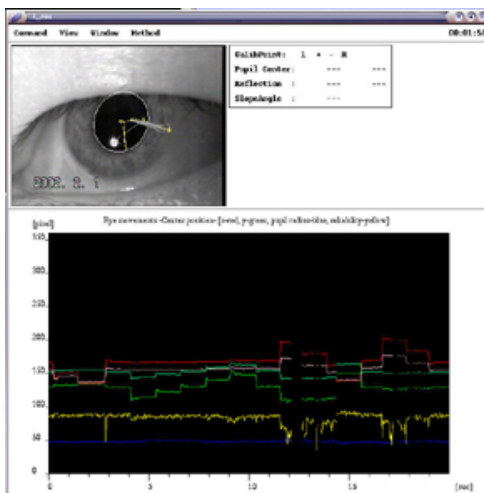
i_rec のセッティングについては、解説書1および2を参照しましょう。

参考までに、メニューについての不完全な解説を巻末に載せておきます。

とりあえず、目を撮影してみましょう。

いきなりサルで条件設定を行うのはサルが待たされてストレスになります。

まずは、ヒトで確認してみましょう。



カメラと目の位置関係は動かないようにしないとイケません。

左の図のように、i_rec の画面に、目はちゃんと映っていますか？

何も画像が映っていない場合は、カメラの接続が間違っていないか、確認しましょう。

目が映っていたら、松田さんの解説書を見ながら i_rec のセッティングを行います。

i_rec の下段には目の位置や瞳孔系を示すグラフが出ています。

十分な証明強度があり、かつ焦点が合っていないと、このグラフが途切れ途切れになります。

条件がばっちり合えば、赤、緑、黄色および青のラインのほか、ピンク色のラインが出てくるはずですが。

また、RIO の設定が正しくされていないと、これらのラインは出てきませんばかりか、明るさやコントラストの設定も出来ません。

メニューバーの **Command->Pause** をしたときに表示される、四角い枠が ROI です。マウス左ボタンを押しながら、ドラッグすると ROI の変更が出来ますので、眼球を十分にカバーするような領域に設定しましょう。

8. キャリブレーションを行う。

i_rec のでは、サルがどこを見ているのかを正確に記録するために、キャリブレーションの機能が用意されています。

ここでは、コンピュータモニタを使った視覚刺激呈示により、キャリブレーションを行う方法を解説します。

まず、コンピュータモニタに指標を呈示するファイルを作ります。

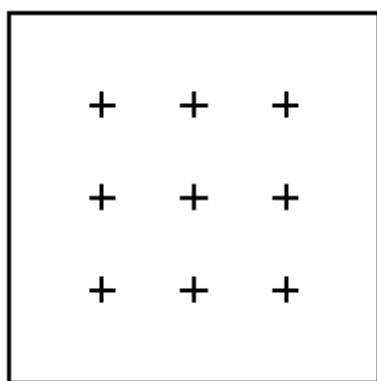
我々の研究室では、心理実験用のフリーソフト「PsyScope」を使用しています。

8-1) キャリブレーションのための視標準備

キャリブレーションのための視標の数は、実際には5個または9個を使います。

図には、9個の場合の視標呈示を模式的に表しています。

5個の場合は、図から上中央、下中央、左中央、右中央を除きます。



我々はめんどくさいので5個の視標を使っています。

モニタの中心および実際の実験で呈示する視覚刺激を充分カバーできるモニタ四隅に指標を呈示しています。

サルには、左上から右下へ順に呈示します。

8-2) 視標と被験者の位置関係

当然ですが、モニタの中心と測定するサルの眼球の中心は垂直になるようにします。

その状態で、モニタの中心とサルの眼球の中心の距離を測ります。

我々の研究室では、700mmです。

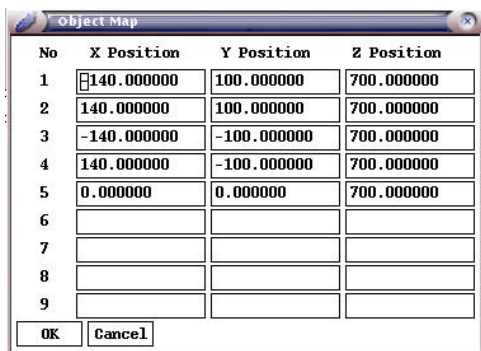
8-3) 視標の数値化

モニタの中心を0として、四隅の指標がどれだけずれているかを数値にします。

実際にモニタ上で測るもよし、指標を呈示しているプログラム上で見るもよしです。キャリブレーションに必要な数値は、それぞれの指標の位置関係、およびモニタの中心から測定するサルと眼球までの距離です。

指標の座標を x および y で表し、サルとモニタの距離を z で表します。

それぞれの数値化ができれば Object Map を作ります。



No	X Position	Y Position	Z Position
1	140.000000	100.000000	700.000000
2	140.000000	100.000000	700.000000
3	-140.000000	-100.000000	700.000000
4	140.000000	-100.000000	700.000000
5	0.000000	0.000000	700.000000
6			
7			
8			
9			

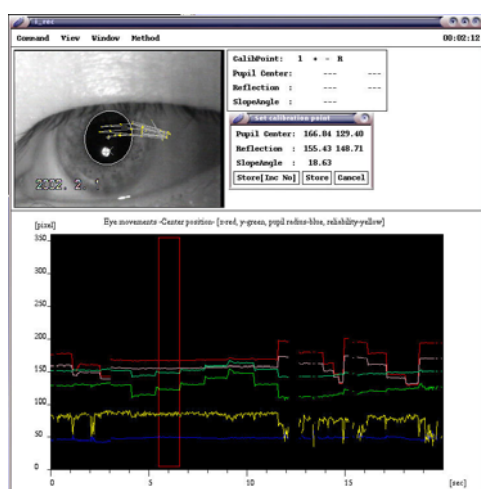
8-4) Object Map ファイル

Object Map ファイルは、視標の座標を呈示順に列挙したもののなので、上記を Object Map ファイルにあらわすと次のようになります。これらの座標は、プログラムの中ではベクトルとして処理されるため、単位をそろえてあれば、cm 単位でも、小数が入っても大丈夫です。

指標の番号は、呈示する順番です。

i_rec のメニューから Window を選び、Object map を開いて値を入力します。

8-5) サルに視標を注視させ、キャリブレーションを取る。



いよいよキャリブレーションです。

が、サルには事前に注視点を注視するようにトレーニングしておく必要があります。明るさやコントラストなどがきちんとしていれば、図のような表示になります。

この状態で視標 1 を注視させます。

サルが視標を注視すると、グラフがまっすぐになるので、この部分を左クリックします。

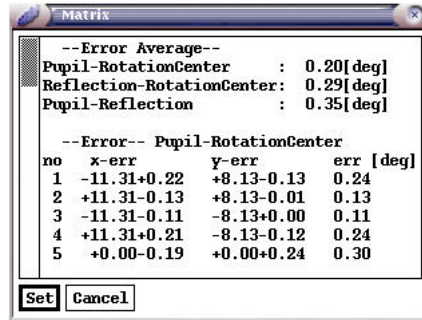
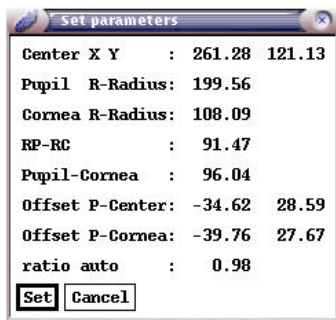
すると、走引が止まり図のように一定の値

を示す部分が赤く囲まれますので、Store[Inc No]を押すと視標 2 に移りますから、右クリックで走引を再開させます。

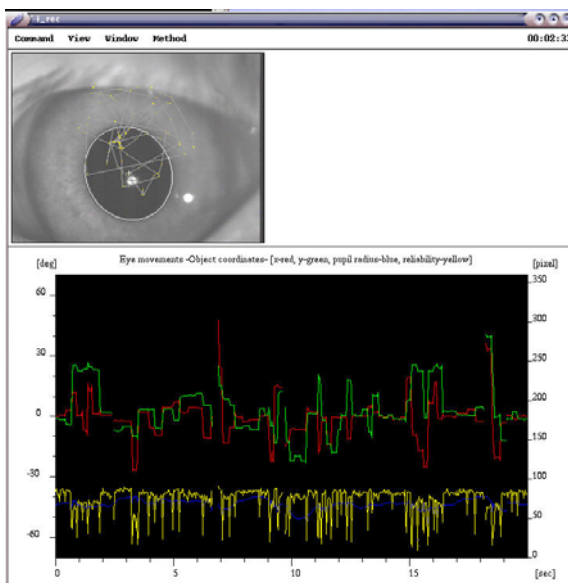
サルには視標 2 を呈示し、注視させます。

これを、視標 5 まで繰り返します。

もし、上手くいけば Set parameter のウインドウが開くのでそのまま set をクリックして、次の Matrix ウインドウでも set をクリックします。



最後にもう一度グラフ画面を右クリックすると、キャリブレーションの取れた状態で測定が始まります。



9. 外部機器への出力確認をする。

i_rec から出力できるのは、アナログデータとして x, y および瞳孔半径、デジタルデータとして XY Graph で設定した範囲に視線がきた時の TTL 信号です。

9-1) DA ボードからの出力

0ch x

1ch y

2ch 瞳孔の半径

が出力されます。

オシロスコープで確認した後、データ記録装置（我々の講座では PLEXON）へ入れてみましょう。

*瞳孔径については、イメージのピクセル数に応じた電圧が出力されるようになっています。従って、カメラやレンズが異なると、非常に小さな数値しか出力されなかったり、逆に飽和してしまって振り切れてしまう場合があります。

ます。これを補正するには、*da.c* の以下の部分を編集します。

static double XOFFSETP=0.0; . . . 瞳孔の大きさの下限、

static double RANGEP=320.0 . . . 瞳孔の大きさの可変範囲

なので、*XOFFSETP+RANGEP* が、瞳孔の大きさの上限になります。

従って、

static double XOFFSETP=0.0

static double RANGEP=160.0

とすると、

瞳孔の半径が 0 ピクセルの時、*-10V* (もしくは*-5V*)

瞳孔の半径が 160 ピクセルの時、*+10V* (もしくは*+5V*)

となり、振幅の幅が 2 倍になります。

9-2) D I/O ボードからの出力

後述する WindowMap で指定した *ch* から出力されます。

詳細は、後述の WindowMap の項を見てください。

タスク制御のコンピュータなどに入れて、**Fixation Task** をやりましょう。

10. *i_rec*メニュー一覧

Command

Pause . . . 画像の取り込みを一時停止します。

Calc parameters [-] . . . なんだろう？

Calc matrix . . . なんだろう？

Reset parameters . . . 設定したパラメータを全てリセットします。うかつに押さない方が身のためです。

Quit . . . *i_rec*を終了するときに使います。

View

Center Position . . . なんだろう？これを選択して記録しているのですが . . .

Camera coordinates . . . なんだろう？

Object coordinates . . . なんだろう？

Window

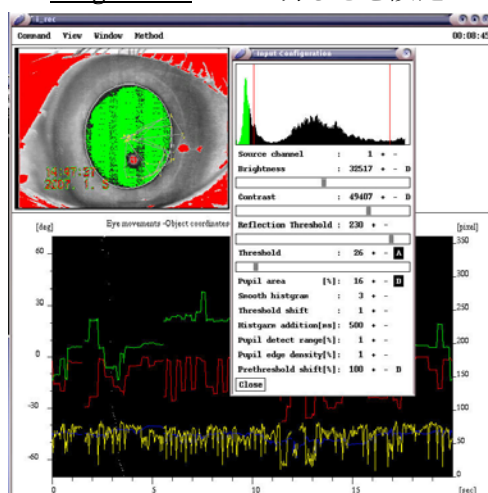
Input configuration . . . 明るさやコントラストなどを調節するウインドウです。

Brightness・・・明るさを設定します。スライダーを左右のマウスボタンで同時に持つ（3つボタンマウスの場合は真中のボタンで持つ）と、任意の位置にスライド

できます。

Contrast・・・コントラストを調節します。スライダーを左右のマウスボタンで同時に持つ（3つボタンマウスの場合は真中のボタンで持つ）と、任意の位置にスライドできます。

Reflection Threshold・・・赤外光が眼球で反射したものを検出する閾値を設定します。スライダーを左右のマウスボタンで同時に持つ（3つボタンマウスの場合は真中のボタンで持つ）と、任意の位置にス



で同時に持つ（3つボタンマウスの場合は真中のボタンで持つ）と、任意の位置にスライドできます。

Threshold・・・測定に使用する光学データの閾値を設定します。右端のAを反転させておけば、自動で設定されます。スライダーを左右のマウスボタンで同時に持つ（3つボタンマウスの場合は真中のボタンで持つ）と、任意の位置にスライドできます。瞳孔だけが緑色で、反射点が赤になるように設定しましょう。

Pupil areas・・・瞳孔の範囲を設定します。画像を見ながら、どこを瞳孔として捉えるのか、決めてください。

Smooth histograms・・・Input Configurationで表示するヒストグラムのスムージングを行います。

3くらいが適当でしょう。

Threshold shift・・・閾値を自動設定した場合に、自動設定された閾値を変移させる値を指定します。

+1くらいが適当でしょう。

Histogram addition・・・閾値を自動設定した場合に、自動設定された閾値の加算時間を指定します。

300～600msec程度が無難でしょう。

Pupil detect range・・・睫や瞼など、邪魔者が入った場合にどの程度で計測を断念するかを決めます。

50%くらいが適当です。

Pupil edge density・・・瞳孔として検出する割合を指定します。1%でオフになり

ます。

Prethreshold shift・・・瞳孔検出に用いる閾値（実は2段階で検出している）を決める際に使用します。

通常は100%です。

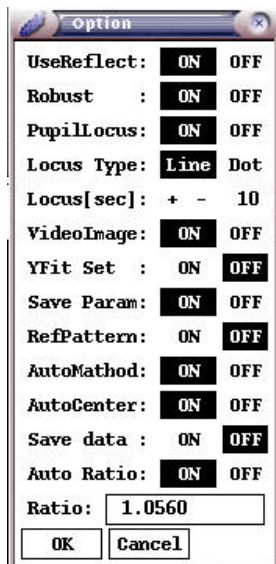
Option・・・ここでもいろいろなセッティングを行います。

UseReflect・・・目とカメラの位置関係がほぼ固定されているならばoffにします。サルが暴れる場合は、onにします。

Robust・・・斜めから目を撮影する場合にオンにします。

PupilLocus・・・視線の軌跡表示のオン・オフです。

Locus Type・・・視線の位置を表示する方法を選びます。



Locus [sec]・・・視線の位置を残像として表示する長さを設定します。

Video Image・・・ビデオ入力を表示するかどうかを選びます。

YFit Set・・・YFit Setは、瞳孔を検出する際に左右方向に加えて、上下方向のエッジを検出するスイッチです。エッジ数は増えるので精度の向上が期待されるのですが、実際には、まぶたの影響を受けやすくなり、場合によっては使わない方が性能はよいようです。

Save Param・・・一回のキャリブレーションで設定したパラメータを保存します。なるべくならオフにしておき、毎回キャリブレーションを行います。

RefPattern・・・キャリブレーションと異なる赤外光反射があった場合に、そのデータを検出不能にします。

AutoMethod・・・検出方法としてPupil-Reflectionを使用している場合に、データの欠損を防ぎます。が、まだ不具合があり、使用しない方が良さそうです。

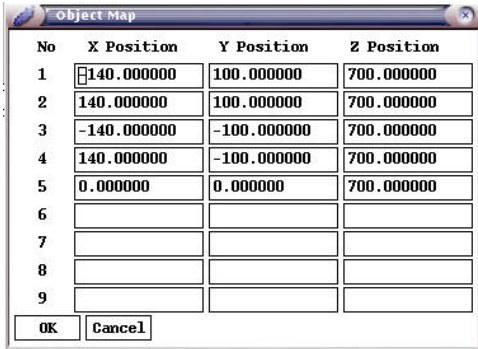
AutoCenter・・・検出方法としてPupil-Rotation Centerを使用している場合に、被験者の動きによるずれを補正します。これも、通常は使用しない方が良さそうです。

Save Data・・・DAコンバータに出力しているデータをファイルとして保存します。i_recを終了しないと保存されません。

Auto Ratio・・・入力画像のピクセル微妙な変形を補正します（長方形→正方形）。

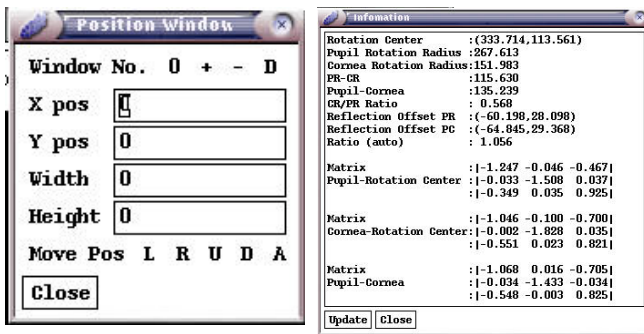
Ratio・・・入力画像の縦横比の補正值を入力します。

Object map・・・キャリブレーションに使う視標の位置を表した表です。



Position window・・・視線の位置に依存してアナログ出力をする時に使用します。キャリブレーションを取っていないなくても、ここで設定したWindow範囲に眼球の中心が来れば、出力が行えます。

Windowは数種類作れますが、右上のDを押してやらないと動作しません。

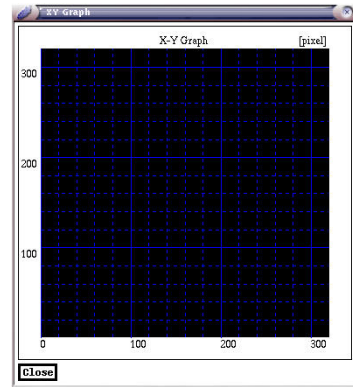
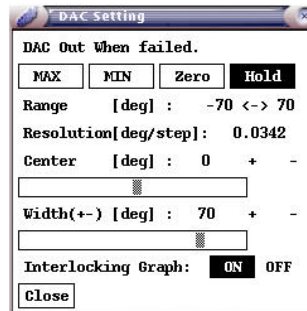


X pos, Y pos, Width, Heightなどに適当な数値を入れ、下段のMoveで動かしてみましよう。すぐにWindowのきり方がわかると思います。

Information window・・・多分、設定情報などを表示するウインドウだと思う。

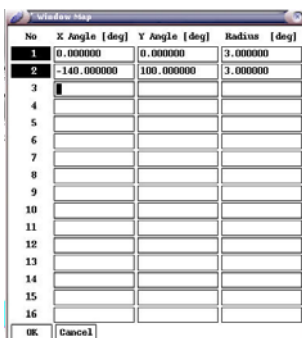
DAC setting・・・多分、アナログ出力の設定を行うウインドウだと思う。

XY Graph・・・検出している視線をリアルタイムに見ることが出来ます。



Fixation taskを行うときには、注視点と許容範囲を示

す緑色の円が描かれます。眼球の中心がその中入ると、その円が赤くなります。



Window map・・・ここで設定したWindow内に視線が来ると、D I/OボードからTTLシグナルが出せます。キャリブレーションを行わないと、動作しません。

例えば、上記ObjectMapに従ってキャリブレーションを取

った後に、被験者が中心を見た時にDIOのCH1 から出力したければ、No1 の行に

X=0, Y=0, Radius=3

と入力します。ここで、**Radius**とは誤差の許容範囲で、この場合は3度までの範囲に入ったら出力するように指定した事になります。

これらの設定を有効化させるためには、左側のチャンネルを示す数字をクリックして白黒反転させます。また、ウインドウを閉じるときには必ず左下の“OK”をクリックします。

もし、キャリブレーションを行った1番の場所を見た時にCH2 から出力するように指定したければ、No2 の行に

X=-140, Y=100, Radius=3

と入力します。

SaveSetting・・・設定を保存します。

Method

Pupil-RotationCenter・・・被験者の頭部が固定されている場合はこの方法を使用します。

Reflection-RotationCenter・・・なんだろう？

Pupil-Reflection・・・被験者の頭部が固定されておらず、動く場合にはこの方法を使用します。

11. i_rec のセットアップ時に使用するターミナルエミュレータコマンド一覧

ターミナルエミュレータ上で

```
% man ls
```

の様に入力すれば、lsに関するマニュアルが開きます。

ls --- List directory contents : ディレクトリに含まれるものを表示します。

su --- Run a shell with substitute user and group IDs : ユーザの変更をします。

tar --- The GNU version of the tar archiving utility : ファイルを解凍(展開)します。

cd --- Change directory? : ディレクトリを変えます。たぶん。

cp --- Copy files and directories : ファイルをコピーします。

xmkmf --- Create a Makefile from an Imakefile : Imakefile から Makefile を作ります。

chown --- Change file owner and groups : ファイルの所有者を変えます。

chmod --- Change file access permissions : ファイルのアクセス許可を変更します。

12. 付記

2003年5月20日	ver. 1 のマニュアル作成開始
2003年7月28日	ver. 1 のマニュアル作成完成
2003年8月10日	ver. 2 のマニュアル作成開始
2003年9月29日	ver. 2 のマニュアル Web 上に試験的アップ
2007年1月14日	ver. 2 のマニュアルを Web 上にアップ
2007年1月17日	ver. 2.0.1 のマニュアルを Web 上にアップ
2008年8月26日	ver. 2.0.2 のマニュアルを Web 上にアップ