〔公　開〕

TR－C－0058

ＰＶ－ＷＡＶＥ　拡張プロシジャー　マニュアル

宮脇　隆志

TAKASHI　MIYAWAKI

１９９０．１２．２８

ＡＴＲ通信システム研究所

PV-WAVEは、ワークステーション上で稼働するデータ解析、表示ソフトである。 主な機能として、多次元データの読み込み、表示、変換、解析（ヒストグラム、フーリエ変換など）、画像処理、グラフアウト（１次元、サーフェス、シェーディング）などがあげられる。また、表示した結果をポストスクリプトファイルに出力するのでdvi2ps等を用いて、TeXの文書中にグラフや、画像を論文に直接挿入できる。

PV-WAVEは、主にインタープリターにてコマンドを実行するが、プロシジャーファイルを作成することにより、プログラミングを行うこともできる。

本報告書は、画像の読み書きや、プリンターへの出力など頻繁かつ汎用的に使われれるコマンドを、使いやすいように拡張したプロシジャーの取り扱い説明書である。

なお、PV-WAVEのライセンスは、現状では（1990年12月末）2cpu分所有している（cs40およびcs33）。cs40においては、これらの拡張プロシジャーは、

/usr1/guest/pvlib

のディレクトリに置かれている。

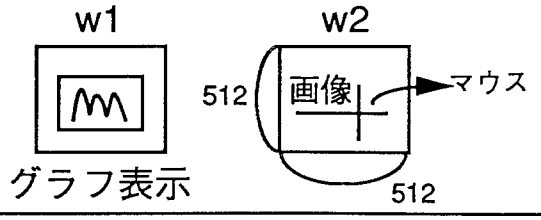
PV-WAVEの使用方法は、マニュアルを参照していただきたい。

内　容

◎　コマンドプロシジャー説明

PV-WAVEコマンドプロシジャー

プログラム名	プログラム名が aaa の時 ファイルは aaa.pro である。
機　能	機能が簡単に書かれている。
引　数	・プログラム実行時に必要な引数。 ・プロシジャー中で使われている名前と 一致する必要なし。
キーワード	・付加的な引数。　付けても付けなくてもよい。 ・num = 23 というように　"="　を付ける。 ・プロシジャーと名前を一致させること。
注意事項	使用上の注意。

プログラム名	CURS
機　能	インタラクティブに1スキャンラインのデータをグラフ化する。 
引　数	w1 : ウィンドウ1の番号 w2 : ウィンドウ 2 の番号 user_image : 画像名(メモリ上の)
キーワード	なし
注意事項	・512 x 512 のウィンドウを2つ (w1, w2) 作っておくこと。 ・w2 上で画像 "user_image" を表示しておくこと。

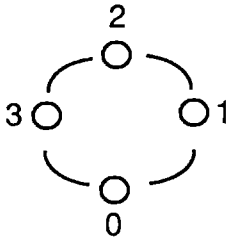
プログラム名	drpoly
機　能	多角形 (ぬりつぶし) 表示 マウスボタン　左 : 頂点指示 右 : 頂点指示終了
引　数	xxx : ポイントされた頂点の x 座標例 yyy : ポイントされた頂点の y 座標例 注 : 正規化された値　(0〜1.0)
キーワード	dens : 明度　0 〜 255
注意事項	

プログラム名	drrect
機　能	1) マウスで2点ピック 2) 対応する長方形領域において画像 img のデータをおきかえる。
引　数	img : 画像名 value : おきかえる値
キーワード	num : 長方形領域の個数 (デフォルトは1)
注意事項	

プログラム名	histmask
機　能	画像 image に対し、マスク画像 mask の255の画素のみのヒストグラフを作る。
引　数	image : 画像 mask : マスク画像 hist : ヒストグラム配列　(1次元) longint (256)
キーワード	なし
注意事項	PV-WAVE の特徴をよく表すプロシジャーである。 参考にされたし ";"の行はコメント行である。

プログラム名	drline
機　能	多角形のライン表示 (閉じていない)
引　数	XXX YYY 頂点データ配列
キーワード	linew : ラインの幅 1 以上
注意事項	

プログラム名	drtext
機　能	1) キーボードより文字入力 2) マウスにて表示位置指定
引　数	fontsize : フォントの大きさ ・省略可 ← デフォルト1 ・小数点入力可、例 1.5
キーワード	X Y　　(0 ～ 511 の整数) が指定されていれば 2) のマウス入力が省略され、X、Y の場所に表示される。
注意事項	ウィンドウサイズ → 512 x 512 であること

プログラム名	imod (ファンクション)
機　能	サイクリックモード 例  ; mod (3,4) = 3 ; mod (6,4) = 2 ; mod (-1, 4) = 3
引　数	value : 数値 modnum : 底
キーワード	なし
注意事項	・色相のヒストグラムによる領域分割に役にたつ ・プロシジャーではない (プロシジャーとファンクションの違いは マニュアルを見ること)

プログラム名　　ohtsu
機　能　　オーツの領域分割手法
引　数
キーワード
注意事項

プログラム名	open win
機　能	ウィンドウ (512 x 512) を作成
引　数	windowid : ウィンドウ名 (変数名)
キーワード	なし
注意事項	openwin, w1 でウィンドウを作った時 そのウィンドウに移動する場合は wset, w1　でよい。

プログラム名	pause
機　能	プロシジャーの一時停止 リターン入力にて再開
引　数	なし
キーワード	なし
注意事項	なし

プログラム名	ps_set
機　能	デバイスをポストスクリプトに変更
引　数	なし
キーワード	bits : 画像のビット数 (デフォルトは8)
注意事項	・WAVE>.run ps というコマンドを実行した後使用すること ・ファイルの指定をしていないので wave.ps というファイルができる。

プログラム名	ps-cap
機　能	デバイスをカプセル化されたポストスクリプトに変更
引　数	なし
キーワード	bits : 画像のビット数 (デフォルトは8)
注意事項	ps_set と同じ

プログラム名	ps_vec
機　能	デバイスをポストスクリプトに変更 (ベクターフォント)
引　数	なし
キーワード	なし
注意事項	・ps_set と同じ ・ベクターフォントにセットされる ・日本語を含んでいる時はこれを使用すること

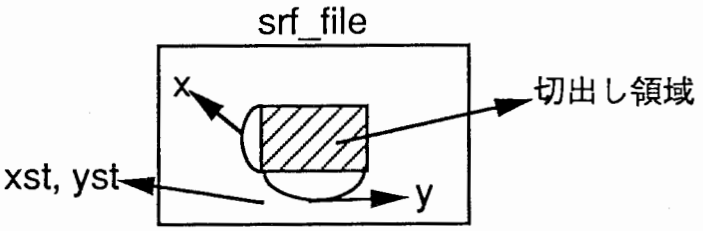
プログラム名	ps_print
機　能	1) ポストスクリプトデバイスを閉じる 2) wave.ps のプリントアウト 3) デバイスを sun にもどす
引　数	Newfile : ファイル名 (省略可) ファイル名の指定があれば wave.ps はこのファイルに変更される。
キーワード	なし
注意事項	・ps_set, ps_cap, ps_vecのどれかを実行しておくこと ・ファイル変更 → aaa.ps の時 WAVE>ps_print, 'aaa.ps'と入力

プログラム名	readimg
機　能	画像データ (1byte データ) の読み込み
引　数	filename : 画像ファイルの名前 img : 画像メモリの名前
キーワード	・rev : 画像の上下を反転する (rev = 1 の時) ・xsize, ysize　画像のサイズ (デフォルト　512 x 512)
注意事項	・データフォーマットは1byte/1pixel のべたファイル ・ファイル名が gazou.r, メモリ名が red の時 WAVE>reading, 'gazou.r', red とする ・ファイルがない時エラーになる そのときは WAVE>retall を実行すること

プログラム名	readxy
機　能	グラフアウト用ファイルの入力 (ファイルの編集必要)
引　数	filen : ファイル名 size : データ数 xx yy　) x,y 座標頂点数
キーワード	なし
注意事項	グラフアウト用ファイルはヘッダー部、 頂点ナンバーを取りのぞき x1 y1 x2 y2 x3 y3 . . というアスキーファイルに直すこと

プログラム名	srf2imagen
機　能	サンラスターファイル (ダンプファイル) を読み込み imagen プリンターに出力する。
引　数	srf_file : サンラスターファイル名
キーワード	bits : 出力する画像のビット数 (デフォルト = 8) nega : =1 の時白黒反転
注意事項	・解像度が半分 (576/450) になる (プリンターのメモリの制約のため) ・ビット数が1の時は、フルレゾリューションの srf2imagen ls を使うとよい

プログラム名	srf2imagen\s
機　能	1) サンのダンプファイルの読み込み 2) imagen に出力 (ランドスケープ表示、ビット数 = 1, 解像度 1152 x 900)
引　数	srf_file : ダンプファイル名 (サンラスターファイル)
キーワード	nega : =1 の時反転表示
注意事項	このコマンド使用後はポストスクリプトの オフセットが xoffset = 0, yoffset = 0 となる

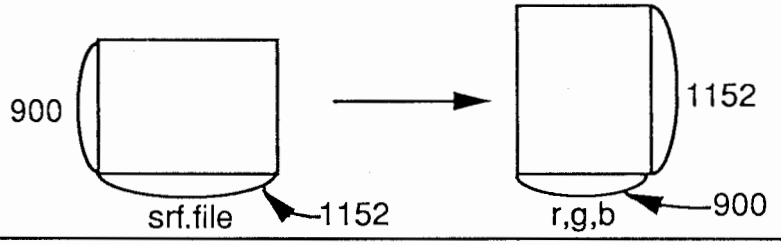
プログラム名	suncut
機　能	1) サンラスターファイル読み込み 2) 領域の切り出し 3) r.g.b ファイルとして書出し
引　数	・srf_filename ; サンラスターファイルネーム 
キーワード	なし
注意事項	r.g.b ファイルは srf_filename.r srf_filename.g srf_filename.b となる

プログラム名	sundump
機　能	サンの画像をダンプし、imagen にプリントアウトする。
引　数	
キーワード	bit : ビット数 (デフォルト = 8) nega : = 1 の時白黒反転
注意事項	・解像度半分 (576, 450) ・PV-WAVEの画像のみが出力される。

プログラム名	sundumpls
機　能	sundump と同じ (ただし、bit 数 = 1, ランドスケープ出力 フル解像度)
引　数	なし
キーワード	nega : = 1 の時反転
注意事項	sundump 参照のこと

プログラム名	writing
機　能	画像データのファイル出力
引　数	filename : 出力ファイル名 work : 画像メモリ名
キーワード	rev : = 1 の時上下反転してセーブする。
注意事項	file 名 = gazou.r メモリ名 = redの時 wave>writing, 'gazou.r', red とする。

プログラム名	writsun
機　能	1) サンラスターファイルの読込み 2) r,g,b データとして出力
引　数	srf_filename : サンラスターファイルネーム
キーワード	なし
注意事項	・r,g,b データは　srf_filename.r srf_filename.g srf_filename.b　になる。 ・出力データは縦、横が反転する。



付1.　news投稿記事抜粋

From: miyawaki@atr-sw.atr.co.jp (Takashi MIYAWAKI)
Newsgroups: atr.csrl.misc
Subject: kanji in pv-wave
Date: 16 Jul 90 02:45:09 GMT
Distribution: atr
Organization: ATR Communication Systems Research Laboratories

宮脇です。

"pv-waveの利用者へお知らせ … 漢字について"

pv-wave のグラフ機能で、アルファベットを用いてタイトルを入力する方法は先日の講習会で習いましたが、ここでは漢字によるタイトルの入力方法についてお知らせします。

漢字を直接入力する方法は２通り考えられます。１つは、JNIXや、JLE のフロントエンドを利用する方法、そしてもう１つは、nemacs上のシェルを使って pv-wave を立ちあげegg の漢字機能を用いる方法です。

漢字入力 その１（フロントエンドによる漢字入力）

- pv-waveを立ちあげます。
- パラメータ !edit_inputを、0にします。（コマンドリコール機能は使えません。）

  wave> !edit_input = 0

- フォントパラメータを、-1にします。

  wave> !p.font = -1

- フロントエンドを使って直接漢字を入れます。

  wave> plot, histogram(image), title='漢字のタイトル'

- パラメータ !edit_inputを、1にします。

  wave> !edit_input = 1

漢字入力 その２（nemacsによる漢字入力）

- nemacsを立ちあげます。
- ノーマルのshellを走らせます。飯田さんが以前投稿していたコマンドリコール機能つきは使えません。

  M-x shell

- 漢字プロセスコードを、EUCにします。

  Ctr-x Ctr-k p もしくは、set-kanji-process-code "euc"

- pv-wave を立ちあげます。

  % wave

- パラメータ !edit_inputを、0にします。

  wave> !edit_input = 0

注意：kanji code は EUC!!

- フォントパラメータを、-1にします。

  wave> !p.font = -1

- eggの機能で、漢字を入力します。

0, unseen,,
*** EOOH ***
From: miyawaki@atr-sw.atr.co.jp (Takashi MIYAWAKI)
Newsgroups: atr.csrl
Subject: pv-wave 3.01
Date: 12 Dec 90 10:26:55 GMT
Distribution: atr
Organization: ATR Communication Systems Research Laboratories

宮脇です。

PV-WAVEが表題のごとくバージョンアップしました。

X対応になったので、実行はCS40・表示は自分のWS、ということができます。

が、いくつが問題があります。

1．UserIDが1000番以下でないと動かない。

2．サーバーがOpenwindowでないと動かない。

1については、年内もしくは今年始めに解決可能です。2については、Openwindowに付いているXのライブラリを使って実行モジュールを作ったためと考えられます。X11R4を使った時エラーが出ました。R3の場合やIrisの場合は、まだ確認していません。これからベンダーに問い合わせて対応策を考えます。

現状において使う方法（サーバ、クライアント方式で）は、

1．サーバー（表示したいWS）をOpenwindowで立ち上げ、

```
% xhost cs40
```

を実行する。

2．guestでCS40にloginした後、Xのサーバーを指定する。

```
% setenv DISPLAY cs○○:0.0
```

3．pv-wave,openwindowの環境設定をする。

```
% source wvsetup
```

4．pv-waveを実行する。

```
% wave
```

5．deviceをXに変更する。

```
WAVE> set_plot,'x'
```

6．例えばデモを走らせる。

```
WAVE> demo
```

なお、wvsetupを行わなければ従来のpv-wave（version2.2）が動きます。

--

（株）ＡＴＲ通信システム研究所　知能処理研究室
宮脇　隆志（miyawaki@atr-sw.atr.co.jp）
TEL 07749-5-1211 Ex. 268

0, unseen,,
*** EOOH ***
From: miyawaki@atr-sw.atr.co.jp (Takashi MIYAWAKI)
Newsgroups: atr.csrl
Subject: Re: pv-wave 3.01
Date: 13 Dec 90 06:03:19 GMT
Distribution: atr
Organization: ATR Communication Systems Research Laboratories
In-reply-to: miyawaki@atr-sw.atr.co.jp's message of 12 Dec 90 10:26:55 GMT

宮脇です。

>>　　２．サーバーがOpenwindowでないと動かない。
>>
>>　　２については、
>>Openwindowに付いているXのライブラリを使って実行モジュールを作ったため
>>と考えられます。X11R4を使った時エラーが出ました。R3の場合やIrisの場合
>>は、まだ確認していません。

　確認しました。

○Sun3の場合

・X11R3表画面（cgfour0）は正常に稼働

・X11R3裏画面（1bitプレーン）は、白黒反転する。

1bitプレーンで画像を表示する場合は、例えば8bitの画像名がimg1とすると、

```
WAVE> tv, byte(255 - img1)
```

とすれば、あとはpvwaveがかってにディザをかけるので、何とか見れるようになります。

○Irisの場合

　おまじないが必要です。ホームディレクトリに .xSGINeWS.cmd というファイルを作り、その中に以下のような一行を書いておきます。

```
/usr/bin/X11/Xsgi -bs -pseudo
```

　上のファイルをはじめて作った場合は、もう一度loginしなおします。そして、twmを走らせるかktermを立ちあげた後、（Xのウインドウから）以前私が投稿した手順にしたがって実行します。

　注意点 ： xhost cs40 を行う時は、必ずXのクライアント（kterm, xclock等）を走らせた後にして下さい。

I thank Dr.takemura for his kind advice.

--

（株）ＡＴＲ通信システム研究所　知能処理研究室
宮脇　隆志（miyawaki@atr-sw.atr.co.jp）
TEL 07749-5-1211 Ex. 268

付2.　拡張プロシジャー

プログラムリスト

```
pro curs, w1, w2, user_image
  wset, w2
  xx = 256
  yy = 256
  tvcrs, xx, yy, /device
  !c = 0
  !err = 0
  while (!err eq 0) do begin
    wset, w1
    if yy lt 0 then yy = 0
    if yy gt 511 then yy = 511
    !p.title = 'Profile of row' + string (511-yy)
    plot,/xstyle, user_image(0:511, yy), yr=[0,255]
    wset, w2
    cursor, xx, yy, 2, /device

  endwhile
  !err = 0
  tvcrs, 0

  !p.title = ''

  wset, w1
end
```

```
pro drpoly, xxx, yyy, dens = dens
;+
; draw polygon
; usage : drpoly (, xxx, yyy, dens = color density)
;-
;
; if you want to use non-interactive, do the following command,
;
;    WAVE> polyfill, /normal, xxx, yyy, col = dens * (!d.n_colors - 1)
;
;   where xxx and yyy are arrays made by the process before.
;
if not(keyword_set(dens)) then dens = 1.
print, ' '
print, '   left-button  : input'
print, '   right-button : close'
print, ' '
pause
;
xx = fltarr(2)
yy = fltarr(2)
xwrk = fltarr(100000)
ywrk = fltarr(100000)
x = 256
y = 256
for i=0, 1000000 do begin
   tvcrs, x, y, /device
   !c = 0
   !err = 0
   while (!err eq 0) do begin
     cursor, x, y, 2, /device
   endwhile
;
   if(!err eq 1) then begin
      xx(1)=x/511.
      yy(1)=y/511.
      if(i ne 0) then plots, /normal, xx, yy
      xx(0)=xx(1)
      yy(0)=yy(1)
      xwrk(i) = xx(1)
      ywrk(i) = yy(1)
   endif
;
   if (!err eq 4) then begin
      !err = 0
      tvcrs, 0
      if (i gt 2) then begin
         xxx = fltarr(i)
         yyy = fltarr(i)
         for j=0, i-1 do begin
             xxx(j) = xwrk(j)
             yyy(j) = ywrk(j)
         endfor
      endif
      polyfill, /normal, xxx, yyy, col = dens * (!d.n_colors - 1)
      return
   endif
;
   !err = 0
endfor
end
```

```
pro drrect, img, value, num = num
;+
; pick two point
; and fill with value
;
; usage : drrect, image-name, value, num = number of rectangle
;-
if not(keyword_set(num)) then num = 1
for kkk=1, num do begin

xx = intarr(2)
yy = intarr(2)
xx0 = 256
yy0 = 256
tvcrs, xx0, yy0, /device
!c = 0
!err = 0
while (!err eq 0) do begin
 cursor, xx0, yy0, 2, /device
endwhile
print, xx0, yy0
xx(0)=xx0
yy(0)=yy0
!err = 0
while (!err eq 0) do begin
 cursor, xx0, yy0, 2, /device
endwhile
!err = 0
tvcrs, 0
print, xx0, yy0
xx(1)=xx0
yy(1)=yy0
xmn = min(xx)
xmx = max(xx)
ymn = min(yy)
ymx = max(yy)

img(xmn:xmx, ymn:ymx) = value

endfor

tv, img
end
```

```
pro drline, xxx, yyy, linew = linew
;+
; draw line
; usage : drline (, xxx, yyy, linew = line-width-value)
;-
;
; if you want to use non-interactive, do the following command,
;
;    WAVE> plots, /norm, xxx, yyy, thick = line-width
;
;   where xxx and yyy are arrays made by the process before.
;
print, ' '
print, '   left-button  : input'
print, '   right-button : quit'
print, ' '
pause
if not(keyword_set(linew)) then linew = 1.
;
xx = fltarr(2)
yy = fltarr(2)
xwrk = fltarr(100000)
ywrk = fltarr(100000)
x = 256
y = 256
for i=0, 1000000 do begin
  tvcrs, x, y, /device
   !c = 0
   !err = 0
   while (!err eq 0) do begin
     cursor, x, y, 2, /device
   endwhile
;
   if(!err eq 1) then begin
      xx(1)=x/511.
      yy(1)=y/511.
      if(i ne 0) then plots, /normal, xx, yy, thick=linew
      xx(0)=xx(1)
      yy(0)=yy(1)
      xwrk(i) = xx(1)
      ywrk(i) = yy(1)
   endif
;
   if (!err eq 4) then begin
      !err = 0
      tvcrs, 0
      if (i gt 1) then begin
         xxx = fltarr(i)
         yyy = fltarr(i)
         for j=0, i-1 do begin
             xxx(j) = xwrk(j)
             yyy(j) = ywrk(j)
         endfor
      endif
      return
   endif
;
   !err = 0
endfor
end
```

```
pro drtext, fontsize, x=x, y=y
;+
; draw text
; usage : drtext, fontsize, x=xval, y=yval
;-
if(n_elements(fontsize) eq 0) then fontsize = 1
!p.font = -1
text = ''
print, 'Enter Text !'
read, text
;
if not(keyword_set(x)) then begin
   x = 256
   y = 256
   tvcrs, x, y, /device
   !c = 0
   !err = 0
   while (!err eq 0) do begin
     cursor, x, y, 2, /device
   endwhile
   !err = 0
   tvcrs, 0
endif
;
print, 'x = ', x, '   y =', y
xyouts, /norm, x/511., y/511., text, size = fontsize
end
```

```
pro histmask, image, mask, hist
;+
; make histogram table in mask region
; usage: histmask, image, mask, histogram-table-name
;-
hist = lonarr(256)
hist = histogram(image(where(mask eq 255)))
;for i=0,511 do begin
;    z = where(mask(i,*) eq 255)
;    checkz= size(z)
;    if(checkz(0) ne 0) then begin
;      for j=0, checkz(1) -1 do begin
;        hist(image(i,z(j))) = hist(image(i,z(j))) +1
;      endfor
;    endif
;endfor
end
```

```
function imod, value, modnum
;+
; integer cyclic mod
;  if modnum = 4, imod(3, 4) = 3, imod(4,4) = 0, imod(-1,4) = 3
;-
ival  = indgen(fix(modnum))
ivals = shift(ival,-fix(value))
return,ivals(0)
end
```

```
pro lg, im, ans, sigma=sigma, thresh=thresh, mode=mode
;+
; laplasian gausian operation
; usage :
;  lg, input-image, operated-image,
;-
; T. Miyawaki '90 4/19

timea = systime(1)
spawn, 'echo $$', pid
tmpif = '~/tmp/lgi' + pid
tmpof = '~/tmp/lgo' + pid

writimg, tmpif(0), im

print,'laplasian gausian operation'

com1= 'rsh ciris2 /usr/local/bin/lg'

if not(keyword_set(mode)) then $
   com1 = com1 + ' -m0'
if (keyword_set(mode)) then $
   com1 = com1 + ' -m' + strcompress(string(mode),/remove_all)
if (keyword_set(mask)) then com1 = com1 + ' -ma' + mask
if (keyword_set(thresh)) then $
   com1 = com1 + ' -t' + strcompress(string(thresh),/remove_all)
if (keyword_set(sigma)) then $
   com1 = com1 + ' -s' + strcompress(string(sigma),/remove_all)

com2= '<' + tmpif + '>' + tmpof

spawn,  com1 + com2
readimg, tmpof(0), ans
spawn,'/bin/rm ' + tmpif
spawn,'/bin/rm ' + tmpof

timeb = systime(1)
print, 'elapsed time = ',timeb-timea,' sec',(timeb-timea)/60.,' min'

end
```

```
pro ohtsu, img, his, sig, smth=smth
;+
; ohtu's threshold method
; usage :
;   pro ohtsu, img-name, his-arry, sig-arry (, smth=smooth parm.)
;-
; 5/1 T. Miyawaki
;
if keyword_set(smth) then his = smooth(histogram(img),smth)
if not(keyword_set(smth)) then his = histogram(img)
nomhis=his/total(his)
sig=dblarr(256)
gray=dblarr(256)
for i=0,255 do gray(i)=i*his(i)
for i=0,254 do begin
  if (total(his(0:i)) gt 0) and (total(his(i+1:255)) gt 0) then $
    sig(i)=total(nomhis(0:i))*total(nomhis(i+1:255)) * $
        ( total(gray(0:i))/total(his(0:i)) - $
          total(gray(i+1:255))/total(his(i+1:255)) )^2
endfor

maxsig = where(sig eq max(sig))
print, maxsig
end
```

```
;window open
;
pro openwin, windowid
win = !D.WINDOW
if (win eq -1) then win = 1 else win = 11 - win
winchr = string(win)
wintitle = 'PV-WAVE WINDOW ' + winchr
WINDOW, /FREE, XSIZE=512, YSIZE=512, XPOS=540, YPOS=350, $
        TITLE=wintitle
windowid = !D.WINDOW
;
end
```

```
pro pause, dummy
;       HAK is a procedure that performs "Hit any key to continue".  It waits
;       for keyboard input, clears the type-ahead buffer and allows the
;       application to continue.
print, 'Hit any key to continue !'
;
dumb = get_kbrd (1)
empty
repeat dumb = get_kbrd (0) until dumb eq ''
empty
;
return
end
```

```
;
;   FOR POSTSCRIPT SETTING AND PRINTOUT
;
; PS_SET   -> DEVICE SET TO POSTSCRIPT
; PS_CAP   -> DEVICE SET TO POSTSCRIPT(ENCAPSULATED)
; PS_VEC   -> DEVICE SET TO POSTSCRIPT(VECTOR FONT)
; PS_PRINT -> PRINTOUT, CLOSE PS_FILE,
;             AND DEVICE RESET
;
;-------------------------------------------------------------
pro ps_set, bits=bits
set_plot, 'ps'
!p.font = 0
;device, /times, /bold, /italic
device, /times, /italic
if not(keyword_set(bits)) then bits = 8
device, bits_per_pixel = bits
end

;-------------------------------------------------------------
pro ps_cap, bits=bits
set_plot, 'ps'
!p.font = 0
device, /times, /italic
device, /encapsulated
if not(keyword_set(bits)) then bits = 8
device, bits_per_pixel = bits
end

;-------------------------------------------------------------
pro ps_vec
set_plot, 'ps'
!p.font = -1
device, bits_per_pixel = 8
end



;-------------------------------------------------------------
; close postscriptfile and printout to imagen printer
; (if you specify New_file, 'wave.ps' will move to the file)
; and set device to sun
;
; usage example
;     WAVE> ps_print
;  or
;     WAVE> ps_print, '~/pvwave/file_name'
;
Pro ps_print, New_file
;old_device = !d.name
set_plot, 'ps'
device, /close
cmd = 'lpr -Pimagen -Lultrascript wave.ps'
if n_elements(New_file) gt 0 then         $
   cmd = [cmd, 'mv wave.ps '+ New_file]
spawn, cmd
!p.font = -1
set_plot, 'sun'
end
```

```
pro readimg, filename, img, rev = rev, xsize=xsize, ysize=ysize
:+
; image data read program
;
; usage
;   readimg, '~/imagedata/file-name', image-name, xsize=xsize, ysize=ysize
;-
if not(keyword_set(xsize)) then xsize=512
if not(keyword_set(ysize)) then ysize=512
temp = assoc (1, bytarr(xsize,ysize))
print, '...Reading file : ' + filename
openr, 1, filename
if not(keyword_set(rev)) then begin &
     work = temp(0)            &
     img = rotate(work, 7)     &
  endif else begin             &
     img = temp(0)             &
endelse
close, 1
end
```

```
pro readxy, filen, size, xx, yy
;+
;   read xy data and plot out
;   usage : readxy, 'file-name', number-of-data, xarray, yarray
;
;-
openr, 1, filen
dat = dblarr(2,size)
xx = dblarr(size)
yy = dblarr(size)
on_ioerror, end_of_file
readf, 1, dat
end_of_file: close, 1
xx(*) = dat(0,*)
yy(*) = dat(1,*)
plot, xx, yy,  /xstyle
end
```

```
pro srf2imagen, srf_file, bits=bits, nega=nega
;+
; Read srf, and printout to imagen printer.
;   This is half resolution of the screen and 256 gray scale.
; If you want full resolution, and if the image can be 1 bit,
; you can use the command 'srf2imagenls', ls means LandScape.
;
; usage : srf2imagen, 'srf-file-name' (, bits = 1 ~ 8, /nega )
;-
; read_srf, tranlate from rgb to black&white, and write to file
;
read_srf, srf_file, im, r, g, b
imm = rotate(im, 7)
im = bytscl(r(imm)*.3 + g(imm)*.59 + b(imm)*.11)
;
set_plot, 'ps'
if not(keyword_set(bits)) then bits = 8
device, bits_per_pixel = bits
device, filename='~/tmp/scr.ps'
if keyword_set(nega) then begin
   tv, 255 - rebin(im, 576, 450)
endif else begin
   tv, rebin(im, 576, 450)
endelse
device, /close
spawn, 'lpr -Pimagen -Lultrascript ~/tmp/scr.ps'
set_plot, 'sun'
;
spawn, '/bin/rm ~/tmp/scr.ps'
;
end
```

```
pro srf2imagenls, srf_file, bits=bits, nega=nega
;+
; Read srf, and printout to imagen printer.
;   This is full resolution of the screen and 1 bit scale.
;
; usage : srf2imagenls, 'srf-file-name' (, /nega )
;-
;
read_srf, srf_file, im, r, g, b
im = rotate(im, 6)
imr = r(im)
img = g(im)
imb = b(im)
im  = bytscl((imr ge 128) or (img ge 128) or (imb ge 128))
;
set_plot, 'ps'
device, bits_per_pixel = 1
device, filename='~/tmp/scr.ps'
device, /inches, xsize=8.4, ysize=11.6
device, xoffset=0, yoffset=0
if keyword_set(nega) then begin
   tv, 255 - im, /inches, .25, .5, xsi=7.8125, ysi=10
endif else begin
   tv, im, /inches, .25, .5, xsi=7.8125, ysi=10
endelse
device, /close
spawn, 'lpr -Pimagen -Lultrascript ~/tmp/scr.ps'
set_plot, 'sun'
;
spawn, '/bin/rm ~/tmp/scr.ps'
;
end
```

```
pro suncut, srf_filename, xst, yst, x, y
;+
; read srf( sun raster file format ) data
; extract region  and write to file (srf_name + .r .g and .b)
;
; usage :
;  suncut, 'srf_filename', xstart-point, ystart-point, xsize, ysize
;                          (from lower-left)
;-
print, '...Reading sun raser file : ' + srf_filename
read_srf, srf_filename, im, r, g, b
im = rotate(im,7)

print, '...Format changing'
im = im(xst : xst +x -1, yst : yst +y -1)

rr = r(im)
gg = g(im)
bb = b(im)

writimg, srf_filename + '.r', rr
writimg, srf_filename + '.g', gg
writimg, srf_filename + '.b', bb

end
```

```
pro sundump, bits=bits, nega=nega
;+
;   Screendump of pvwave screen to sun-raster-file(srf),
; read srf, and printout to imagen printer.
;   This is half resolution of the screen and 256 gray scale.
; If you want full resolution, and if the image can be 1 bit,
; you can use the command 'sundumpls', ls means LandScape.
;
; usage : sundump (, bits = 1 ~ 8, /nega )
;-
;
spawn, 'screendump ~/tmp/scr'
print, 'printing to imagen printer'
;
read_srf, '~/tmp/scr', im, r, g, b
im = rotate(im, 7)
;
set_plot, 'ps'
if not(keyword_set(bits)) then bits = 8
device, bits_per_pixel = bits
device, filename='~/tmp/scr.ps'
if keyword_set(nega) then begin
   tv, 255 - rebin(im, 576, 450)
endif else begin
   tv, rebin(im, 576, 450)
endelse
device, /close
spawn, 'lpr -Pimagen -Lultrascript ~/tmp/scr.ps'
set_plot, 'sun'
;
spawn, '/bin/rm ~/tmp/scr ~/tmp/scr.ps'
;
end
```

```
pro sundumpls, nega=nega
;+
;   Screendump of pvwave screen to sun-raster-file(srf),
; read srf, and printout to imagen printer.
;   This is full resolution of the screen and 1 bit scale.
;
; usage : sundumpls (, /nega )
;-
;
spawn, 'screendump ~/tmp/scr'
print, 'printing to imagen printer'
;
read_srf, '~/tmp/scr', im, r, g, b
im = rotate(im, 6)
;
set_plot, 'ps'
device, bits_per_pixel = 1
device, filename='~/tmp/scr.ps'
device, /inches, xsize=8.4, ysize=11.6
device, xoffset=0, yoffset=0
if keyword_set(nega) then begin
   tv, 255 - im, /inches, .25, .5, xsi=7.8125, ysi=10
endif else begin
   tv, im, /inches, .25, .5, xsi=7.8125, ysi=10
endelse
device, /close
spawn, 'lpr -Pimagen -Lultrascript ~/tmp/scr.ps'
set_plot, 'sun'
;
spawn, '/bin/rm ~/tmp/scr ~/tmp/scr.ps'
;
end
```

```
pro writimg, filename, work, rev = rev
print, '...Writing file : ' + filename
openw, 1, filename
if not(keyword_set(rev)) then begin &
     img = rotate(work, 7)      &
endif else begin                &
     img = work                 &
endelse
writeu, 1, img
close, 1
end
```

```
pro writsun, srf_filename
;+
; read srf( sun raster file format ) data,
; rotate 90 degree , and
; wirte to r, g, b data ( not srf data)
; usage : writsun, srf_filename
;-
print, '...Reading sun raser file : ' + srf_filename
read_srf, srf_filename, im, r, g, b
print, 'rotate 90 degree and transform to r, g and b data'
im = rotate(im, 1)
rdata = r(im)
gdata = g(im)
bdata = b(im)
rfile = srf_filename + '.r'
gfile = srf_filename + '.g'
bfile = srf_filename + '.b'
print, '...Writing to ' + rfile
openw, 1, rfile
writeu, 1, rdata
close, 1
print, '...Writing to ' + gfile
openw, 1, gfile
writeu, 1, gdata
close, 1
print, '...Writing to ' + bfile
openw, 1, bfile
writeu, 1, bdata
close, 1
print, 'data size of r, g, b -> H : 900, V : 1152 '
end
```