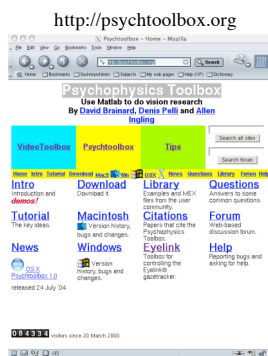


Computer exercise: Psychophysics Toolbox

Volker Franz
University of Giessen
Germany

Psychophysics Toolbox



Exercise 1

Type:

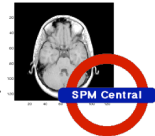
```
>> help psychtoolbox  
>> help psychdemos
```

Try one or two of the following demos:

- ClutMovieDemo
- ColorGratingDemo
- ContrastThreshDemo
- DriftDemo
- GetNumberDemo
- GratingDemo
- LineDemo
- MovieDemo
- NoiseDemo
- PhaseDistortDemo
- TextDemo

Matlab

- high-level language for technical computing.
- interactive!
- vector oriented!
- extensible: toolboxes!
- MEX: interface to C, Fortran, etc.



Starting Matlab without Java Virtual Machine

Matlab from DOS:

```
>matlab -nojvm
```

Matlab editor from DOS:

```
>meditor
```

... or create shortcut on desktop...

Exercise 2

- Get all example files from the fileserver:
 \public\psychtoolbox
- Run and inspect:
 >> help pthello_blank
 >> pthello_blank
 >> type pthello_blank

(Note: this is not hello world yet...)

Exercise 3

- Copy pthello_blank.m to myhello_error.m
- Rewrite myhello_error.m such that a Matlab error is simulated when you press the escape key (instead of closing the window).
- Use the Matlab function error for this:
 >> help error
 >> error('this simulates an error')

Try to get your Matlab window back after the error...

Exercise 4

- Copy pthello_blank.m to myhello_text.m
- Rewrite myhello_text such that
 “Hello World”
 is written to the window.
- Use the command:
 >> SCREEN(window,'DrawText','Hello world',100,100)
- Get help with:
 >> SCREEN('DrawText?')
- Get my solution from the fileserver:
 pthello_text.m

Exercise 5

- Copy pthello_blank.m to myfechner.m
- Read the image pict_fechner.tif into Matlab:
 >> fechner=imread('pict-fechner.tif')
- ... and display it on the screen:
 >> SCREEN(window,'PutImage',fechner)

- Get my solution from the fileserver:
 ptfechner.m
- Inspect the picture you read into Matlab:
 >> fechner=imread('pict-fechner.tif')
 >> size(fechner)
 >> class(fechner)
- Determine the minimal and the maximal grey-values of the picture.
 >> help min
 >> help max

Exercise 6

```
>> max(fechner)
>> max(max(fechner))
>> size(fechner)
>> size(fechner(:))
>> max(fechner(:))
```

- Copy ptfechner.m to ptfechnerinv1.m
- Invert the picture of Fechner (i.e., create a negative).
- Tips:

```
>> class(fechner)
>> help uint8
>> fechner_d = double(fechner)
>> class(fechner_d)
```

Exercise 7

- fechner=double(fechner)
- fechnerinv=255-fechner

Get my solution from the fileserver:

- ptfechnerinv1.m

- Copy ptfechnerinv1.m to myfechnerinv2.m .
- Rewrite myfechnerinv2.m such that the picture of Fechner and its negative are shown 10 times in succession (with a delay of 1 sec in between).
- Tips:

```
>> help pause
```

(Danger: Naïve approach!!!)
- Get my solution from the fileserver:
- ptfechnerinv2.m

Exercise 8

- Copy ptfechnerinv1.m to myfechnermask1.m .
- Create a simple, random mask with the same average luminance as the fechner picture.
- Display Fechner for ~20 msec directly followed by the mask which is shown for 1000 msec.
- Tips:

```
>> help randperm
```

Get my solution from the fileserver:

- ptfechnermask1.m

Exercise 9

- Copy ptfechnermask1.m to myfechnermask2.m
- Measure the timing accuracy of your Fechner presentation.
- Tips:

```
>> help GetSecs
```

Get my solution from the fileserver:

- ptfechnermask2.m

Demo 1

Get from the fileserver:

- ptfechnermask3.m
- Compare the accuracy of the timing to ptfechnermask2.m
- Inspect the code in the time-critical part (e.g.,: 'WaitBlanking')

Demo 2

Get from the fileserver:

- ptfechnermask4.m
- Compare the accuracy of the timing to ptfechnermask3.m
- What is different?

Demo 3

Get from the fileserver:

- ptmask2.m

>> help ptmask2

Summary...

- Timing of stimuli is difficult (and typically machine – dependent). PsychToolbox helps with this!
- Ideally use oscilloscope to check timing.
- Get all remaining files from the fileserver (including this presentation).

Thank you for your attention...

