MATLAB based PowerPoint to image converter

A MATLAB function to convert PowerPoint slides to images in a designated resolution

Adrian Etter

University of Zurich
Department of Economics
Winterthurerstrasse 30
CH-8006 Zurich

E-mail: adrian.etter@econ.uzh.ch
URL: http://www.econ.uzh.ch/faculty/etter.html

1 March 2013
Abstract

PowerPoint is a mighty tool to implement quickly a series of visual presentations. Most researchers already worked with PowerPoint and know how to use it. Alignment, emphasis, text style, basic geometric shapes, displaying of images and many other things are done very fast.

Why not use it to display visual stimuli?

One of the problems is that storing images out of PowerPoint is not so easy controllable. On Windows it’s only possible to change the resolution in steps of 10 by changes a specific key in the registry. But for the most laboratory computer the registry is not changeable. Besides it allows only one JPEG – format.

Whereas on PowerPoint on Mac OSX on one hand it is possible to set a custom resolution and on the other to set the designated file type.

Another problem is, that for different laboratory setups, different resolutions might be necessary. Therefor most researchers program also static visuals in their preferred programming language. Sometimes it takes a long time to calculate position, text spacing, alignment, font size and so on.

Wouldn’t it be very handy to benefit from the easiness of PowerPoint to create visual stimuli?

With the MATLAB based PowerPoint to image converter it is possible to dynamically generate images from PowerPoint slides on the fly. The MATLAB function opens an ActiveX Server which allows to access certain functionalities in PowerPoint that are not directly available in PowerPoint on Windows. Thereby it’s possible to export PowerPoint slides with the designated laboratory/experiment resolution and in the preferred file format.

For simple experiments it’s more than sufficient to create the visuals in PowerPoint and then export it to the required format of the stimuli screen and present it with the preferred visual stimuli software, e.g. Cogent / PsychToolBox, Mgl, ...

For more complicated experiments with dynamic content or random objects on random locations just code out the complicated stages and the rest could still be done with PowerPoint.

Copyright

Copyright © 2007-1013 Adrian Etter. All rights reserved.

This document may be copied, modified, reproduced and redistributed for educational and personal use as long as the original author is mentioned and cited.

MATLAB® is a registered Trademark of MathWorks, Inc.™ (http://www.mathworks.com).
Windows® is a registered Trademark by Microsoft® (http://www.microsoft.com).
PowerPoint® presentation graphics program is a registered Trademark by Microsoft® (http://www.microsoft.com).
Cogent 2000, developed at the FIL and the ICN, is a complete PC-based software environment for functional brain mapping experiments, containing commands useful for presenting scanner-synchronised visual stimuli (Cogent Graphics), auditory stimuli, mechanical stimuli, and taste and smell stimuli. It is also used in monitoring key presses and other physiological recordings from the subject. (LoN 2011, 2013)
Psyctoolbox or The Psychophysics Toolbox is a collection of Matlab functions that extend the language to give you exquisite control of your computer to test observers with research-grade dynamic stimuli. (Psychtoolbox, 2013)
Mgl is a suite of mex/m files for displaying visual psychophysics stimuli and writing experimental programs in MATLAB (GRU RIKEN Brain Science Institute, 2013)
Table of contents:

Abstract .................................................................................................................. 3

Terms and Conditions .......................................................................................... 5
  Acknowledgment ................................................................................................. 5
  License .................................................................................................................

Installation ........................................................................................................... 6
  Requirements ........................................................................................................

Usage Example ..................................................................................................... 6
  Basic Workflow ...................................................................................................
  Step 1: Define your Setup ..................................................................................
  Step 2: Convert your PowerPoint ........................................................................
  Step 3: Get the images ......................................................................................

4 different examples on how to display the images ......................... 10
  Example 1: Display with a MATLAB figure ......................................................
    Create - Open a figure or a window to display the images.........................
    Bring the images to a format that can be displayed in that figure/window. Create
    Textures/Sprites ..............................................................................................
    Display the images in that Figure/Window ..................................................
    Close that Figure/Window ............................................................................

  Example 2: Psychtoolbox ...................................................................................
    Create - Open a figure or a window to display the images..........................
    Bring the images to a format that can be displayed in that figure/window. Create
    Textures/Sprites ..............................................................................................
    Display the images in that Figure/Window ..................................................
    Close that Figure/Window ............................................................................

  Example 3: Cogent .............................................................................................
    Create - Open a figure or a window to display the images..........................
    Bring the images to a format that can be displayed in that figure/window. Create
    Textures/Sprites ..............................................................................................
    Display the images in that Figure/Window ..................................................
    Close that Figure/Window ............................................................................

  Example 3: mgl .................................................................................................
    Create - Open a figure or a window to display the images..........................
    Bring the images to a format that can be displayed in that figure/window. Create
    Textures/Sprites ..............................................................................................
    Display the images in that Figure/Window ..................................................
    Close that Figure/Window ............................................................................

How to export PowerPoint slides on PowerPoint for Mac OSX 14

Acknowledgment ................................................................................................. 17

Bibliography ......................................................................................................... 17
Terms and Conditions

Acknowledgment
You are allowed to use this software for free, but please acknowledge if you are using this software to perform your experiment with the following statement:

The visuals of this experiment were exported from PowerPoint® with the MATLAB® ppt2image function programmed by Adrian Etter at the University of Zurich.

License
The ppt2image function is Licensed under the BSD 2 License.

Copyright
Copyright (c) 2013, Adrian Etter
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Neither the name of the UNIVERSITY OF ZURICH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Installation

Requirements
To successfully run the ppt2image function ActiveX must be enabled and PowerPoint and MATLAB must be installed.

Usage Example
Use the “usageExample.m” to apply it for your experiment. Here follows a detailed description of the usage example.
On PowerPoint for Mac OSX follow first the guide in chapter: How to export PowerPoint slides on PowerPoint for Mac OSX

Basic Workflow
A basic workflow on how an experiment could be programmed can be described with the following figure Fig. 1

![Diagram of Basic Workflow](image)

Fig. 1 Basic workflow

After creating the experiment, switch to MATLAB and export all slides using the ppt2image function. The time spent programming in MATLAB could be reduced dramatically as the presenting of images always follows the same procedure.
A 7 step procedure

This procedure can be described in 7 steps:

1. Step 1: Define your Setup
2. Step 2: Convert your PowerPoint
3. Step 3: Get the images
4. Step 4: Open a figure / screen / window with the required toolbox
5. Step 5: Prepare the images for displaying with the specified toolbox
6. Step 6: Show the images with that toolbox
7. Step 7: Close / free the figure / screen / window used

Find on the next pages the usage example described in detail.
**Step 1: Define your Setup**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWERPOINT</td>
<td>'StroopExample.pptx'</td>
</tr>
<tr>
<td>IMAGEFOLDER</td>
<td>'images'</td>
</tr>
<tr>
<td>SPRITEWIDTH</td>
<td>1440</td>
</tr>
<tr>
<td>SPRITEHEIGHT</td>
<td>900</td>
</tr>
<tr>
<td>FILETYPE</td>
<td>'jpg'</td>
</tr>
<tr>
<td>DISPLAYDURATION</td>
<td>1</td>
</tr>
</tbody>
</table>

**Step 2: Convert your PowerPoint**

On Mac OSX follow the guide in chapter: How to export PowerPoint slides on PowerPoint for Mac OSX

```matlab
if ispc
    ppt2image(PowerPoint, SPRITEWIDTH, SPRITEHEIGHT, IMAGEFOLDER, FILETYPE);
end
```

**Step 3: Get the images**

```matlab
if ispc
    fileType = upper(FILETYPE); % For some reason under Windows, % the export function makes capital % filetype, e.g: JPG
else
    fileType = lower(FILETYPE); % On Mac Office the filetype is lowered
end

c = [pwd filesep IMAGEFOLDER]; % Store foldername in a % variable
c = dir({c filesep '*.Kat'}); % get all the created % images

if numel(filenames) < 1
    if ismac
        % code
    end
```
open('PowerPoint2011Export.pdf');
open(POWERPOINT);
warning('ppt2image:numel', ['There are no images in the provided ' ...
    'folder. Please read instructions ' ...
    'on how to export Powerpoint slides ' ...
    'on a mac!']);
else
    warning('ppt2image:numel', ['There are no images in the provided ' ...
    'folder. Was your Powerpoint empty?']);
end
return;
end
for currentImage = 1 : nrImages
    images{currentImage, 1} = double(imread([folder filesep ...
        filenames(currentImage).name]));
end
4 different examples on how to display the images

All the examples follow the same procedure that can be described in the following four steps:

- Step 4: Create - Open a figure or a window to display the images
- Step 5: Bring the images to a format that can be displayed in that figure/window
- Step 6: Display the images in that Figure/Window
- Step 7: Close that Figure/Window

**Example 1: Display with a MATLAB figure**

Create - Open a figure or a window to display the images.

```matlab
% % Create a figure/window
handle = imagesc(zeros(size(images{currentImage, 1})));
axis off ij image;
% without image processing toolbox the images are to big to display in
% figure with imshow, therefor we display them in grayscales
colormap('gray');
```

Bring the images to a format that can be displayed in that figure/window. Create Textures/Sprites

```matlab
% Create Textures/Sprites
Textures = cell(nrImages, 1);
for currentImage = 1 : nrImages
    textures{currentImage, 1} = mean(images{currentImage, 1}, 3);
end
```

Display the images in that Figure/Window.

```matlab
% Display Textures/Sprites
for currentImage = 1 : nrImages
    set(handle, 'CDATA', textures{currentImage, 1});
    drawnow();
    pause(DISPLAYDURATION);
end
```

Close that Figure/Window.

```matlab
% Close Figure/Screen
close(get(get(handle, 'Parent'), 'Parent'));
```
Example 2: Psychtoolbox

Create - Open a figure or a window to display the images.

```matlab
% Create a figure/window
[windowHandle, rectangle] = Screen('OpenWindow', ...
  max(Screen('Screens')), [], ...
  [0 0 SPRITEWIDTH SPRITEHEIGHT]);
```

Bring the images to a format that can be displayed in that figure/window. Create Textures/Sprites

```matlab
% Create Textures/Sprites
textures = cell(nrImages, 1);
for currentImage = 1 : nrImages
  textures{currentImage, 1} = Screen('MakeTexture', ...
    windowHandle, images{currentImage, 1});
end
```

Display the images in that Figure/Window.

```matlab
% Display Textures/Sprites
for currentTexture = 1 : nrImages
  Screen('DrawTexture', windowHandle, textures{currentTexture, 1});
  Screen('flip', windowHandle);
  pause(DISPLAYDURATION);
end
```

Close that Figure/Window.

```matlab
% Close Figure/Screen
Screen('CloseAll');
```
Example 3: Cogent

Create - Open a figure or a window to display the images.

% Create a figure/window
BLACK = [0, 0, 0];
WHITE = [1, 1, 1];
BACKBUFFER = 0;
MONITOR = 0; % Window
% Display with Cogent
cgopen(SPRITEWIDTH, SPRITEHEIGHT, 0, 0, MONITOR);

Bring the images to a format that can be displayed in that figure/window. Create Textures/Sprites

% Create Textures/Sprites
for currentSprite = 1 : nrImages
    image = uint8(images{currentSprite, 1});
    image = repmat(image, [1,1,3]);
    [pixelArray, width, height] = im2cgpixval(image);
    cgloadarray(currentSprite, SPRITEWIDTH, SPRITEHEIGHT, ...
                pixelArray, width, height);
end

Display the images in that Figure/Window.

% Display Textures/Sprites
for currentSprite = 1 : nrImages
    cgdrawsprite(currentSprite, 0, 0);
    cgflip(BLACK);
    pause(DISPLAYDURATION);
end

Close that Figure/Window.

% Close Figure/Screen
cgshut();
**Example 3: mgl**

Create - Open a figure or a window to display the images.

```
% Create a figure/window
BLACK       = [0, 0, 0];
WHITE       = [255, 255, 255];
MONITOR     = 0;
% Display with Mgl
mglOpen(MONITOR, SPRITEWIDTH, SPRITEHEIGHT, 0, 0);
mglClearScreen(BLACK);
mglScreenCoordinates();
```

Bring the images to a format that can be displayed in that figure/window. Create Textures/Sprites

```
% Create Textures/Sprites
textures = cell(nrImages, 1);
for currentImage = 1 : nrImages
    textures{currentImage, 1} = ...
        mglCreateTexture(images{currentImage, 1});
end
```

Display the images in that Figure/Window.

```
% Display Textures/Sprites
for currentSprite = 1 : nrImages
    mglBltTexture(textures{currentSprite}, [SPRITEWIDTH/2, ...
                  SPRITEHEIGHT/2, ...
                  SPRITEWIDTH, ...
                  SPRITEHEIGHT]);
    mglFlush();
    pause(DISPLAYDURATION);
end
```

Close that Figure/Window.

```
% Close Figure/Screen
mglClose();
```
How to export PowerPoint slides on PowerPoint for Mac OSX

Fig. 2 A finished Experiment produced in PowerPoint, ready to export

After finishing your experiment in PowerPoint, click in the menu bar on File \(\Rightarrow\) Save as Pictures...

Fig. 3 Save as Pictures exports the PowerPoint in an image format
Fig. 4 In the "Options-Menu" the required resolution can be given

Click on "Options..." to change the resolution according to the screen resolution in the laboratory where the experiment will be conducted.

Fig. 5 Change Resolution

Enter the required width and height and apply with the OK-Button. Sometimes the monitors in the laboratory have a different screen aspect ratio. If not the default aspect ratio (4:3) is required, the aspect ratio can be changed before step 2 in the menu: Menubar → File → Page Setup → Slides sized for: 16:10... 16:9 ... Custom and so on.
Fig. 6 Apply the file format required

**Folder Name**
Select the file format required for the experiment, e.g. png, jpg, bmp, gif, tif. The **Save As** name specifies the folder where the slides will be stored.

**Import to MATLAB**
Import pictures to Matlab. See Examples → analog to Windows Version!
Acknowledgment

Many thanks to Stefan Schmid for revising, spotting typos and contribute with helpful hints.

Bibliography