Subject: Running this MATLAB code in IDL
Posted by mikejohnryan08 on Thu, 24 Apr 2014 10:39:24 GMT
View Forum Message <> Reply to Message

Hello,

What would be the best way to run this MATLAB code in IDL (http://www.ee.cuhk.edu.hk/~snli/MATLAB_code.zip) which creates a 3d red/blue anaglyph from two images?

Also, since this is a code from an academic source, would I be able to use this to build onto for a more involved program for a masters thesis as long as I source the code's author in my report?

Thanks!

Subject: Re: Running this MATLAB code in IDL
Posted by Michael Galloy on Thu, 24 Apr 2014 20:25:25 GMT
View Forum Message <> Reply to Message

On 4/24/14, 4:39 AM, mikejohnryan08@gmail.com wrote:
> Hello,
>
> What would be the best way to run this MATLAB code in IDL
> (http://www.ee.cuhk.edu.hk/~snli/MATLAB_code.zip) which creates a 3d
> red/blue anaglyph from two images?

You want to launch Matlab and run your code? SPAWN is the only way I
know of doing that.

I have a very simple class to create red/cyan or red/blue anaglyphs in
my library (github.com/mgalloy/mglib) at:

https://github.com/mgalloy/mglib/blob/master/src/vis/objectg
raphics/mggr3dconverter__define.pro

Call the ::_combineImages method to create the anaglyph image. But if
you already have the two images, there is not much to it -- I would just
use my code as a guide and write your own.

> Also, since this is a code from an academic source, would I be able
> to use this to build onto for a more involved program for a masters
> thesis as long as I source the code's author in my report?

For questions like these, check the license of the code which is either
in comments at the top of the code or in a LICENSE/COPYING/README type
file. In your case, the code has this comment:

%Copyright(c) 2013 Li Songnan
%All Rights Reserved.

Do not use, although you could always ask the author for special permission.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Running this MATLAB code in IDL
Posted by Craig Markwardt on Fri, 25 Apr 2014 03:42:18 GMT
View Forum Message <> Reply to Message

On Thursday, April 24, 2014 4:25:25 PM UTC-4, Mike Galloy wrote:
>> Also, since this is a code from an academic source, would I be able
>> to use this to build onto for a more involved program for a masters
>> thesis as long as I source the code's author in my report?
>
>
>
> For questions like these, check the license of the code which is either
> in comments at the top of the code or in a LICENSE/COPYING/README type
> ...

Beyond simple licensing rights, you really need to speak with your Master's advisor about what is the appropriate level of work for your project.

As a purely practical matter, I think it would be a little nuts to mix IDL and Matlab. These are not languages that were designed to talk to each other, and both languages have strange quirks. And then there is the question of audience. Few people have Matlab, even fewer have IDL, and a vanishingly small number of people have both. I recommend sticking with one language.

CM

Subject: Re: Running this MATLAB code in IDL
Posted by mikejohnryan08 on Mon, 28 Apr 2014 01:00:55 GMT
View Forum Message <> Reply to Message

On Thursday, April 24, 2014 4:25:25 PM UTC-4, Mike Galloy wrote:
> On 4/24/14, 4:39 AM, mikejohnryan08@gmail.com wrote:
>

>> Hello,
>
>>
>
>> What would be the best way to run this MATLAB code in IDL
>
>> (http://www.ee.cuhk.edu.hk/~snli/MATLAB_code.zip) which creates a 3d
>
>> red/blue anaglyph from two images?
>
>
>
> You want to launch Matlab and run your code? SPAWN is the only way I
>
> know of doing that.
>
>
>
> I have a very simple class to create red/cyan or red/blue anaglyphs in
>
> my library (github.com/mgalloy/mglib) at:
>
>
>
> https://github.com/mgalloy/mglib/blob/master/src/vis/objectg
raphics/mggr3dconverter__define.pro
>
>
>
> Call the ::_combineImages method to create the anaglyph image. But if
>
> you already have the two images, there is not much to it -- I would just
>
> use my code as a guide and write your own.
>
>
>
>> Also, since this is a code from an academic source, would I be able
>
>> to use this to build onto for a more involved program for a masters
>
>> thesis as long as I source the code's author in my report?
>
>
>
> For questions like these, check the license of the code which is either
>
> in comments at the top of the code or in a LICENSE/COPYING/README type

>
> file. In your case, the code has this comment:
>
>
>
> %Copyright(c) 2013 Li Songnan
>
> %All Rights Reserved.
>
>
>
> Do not use, although you could always ask the author for special permission.
>
>
>
> Mike
>
> --
>
> Michael Galloy
>
> www.michaelgalloy.com
>
> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

Thanks for the responses! I agree that it would not be a good idea to attempt using the MATLAB code in any way towards thesis research, and that it would be better to write my own simple IDL function to do this.

Mike, thanks for pointing me towards your code library. The "combineImages" method from your anaglyph class seems to be EXACTLY what I am trying to implement in IDL, and seems like it would be a very basic function; since as you said, I already have the left and right images. But since I am completely new to IDL, I can't even begin to think about how that method could be turned into a standalone function. It should be very basic, right?

Thanks again!

Subject: Re: Running this MATLAB code in IDL
Posted by Michael Galloy on Mon, 28 Apr 2014 01:20:16 GMT

View Forum Message <> Reply to Message

On 4/27/14, 7:00 pm, mikejohnryan08@gmail.com wrote:

> Thanks for the responses! I agree that it would not be a good idea
> to attempt using the MATLAB code in any way towards thesis research,
> and that it would be better to write my own simple IDL function to do
> this.
>
> Mike, thanks for pointing me towards your code library. The
> "combineImages" method from your anaglyph class seems to be EXACTLY
> what I am trying to implement in IDL, and seems like it would be a
> very basic function; since as you said, I already have the left and
> right images. But since I am completely new to IDL, I can't even
> begin to think about how that method could be turned into a
> standalone function. It should be very basic, right?

Methods have access to the instance variables of the object and can call other methods of the object using the "self" variable, so watch out for them. In the _combineImages example, there was only the self.color variable, so how about something like:

```
function mg_create_anaglyph, leftImage, rightImage, color=color
  compile_opt strictarr

  ; define combined_image to the correct size
  combinedImage = leftImage * 0B
  dims = size(leftImage, /dimensions)

  if (keyword_set(color)) then begin
    combinedImage[0, *, *] = leftImage[0, *, *]
    combinedImage[1, *, *] = rightImage[1, *, *]
    combinedImage[2, *, *] = rightImage[2, *, *]
  endif else begin
    _leftImage = byte(total(fix(leftImage), 1) / 3)
    _rightRight = byte(total(fix(rightImage), 1) / 3)

    combinedImage[0, 0, 0] = Reform(_leftImage, 1, dims[1], dims[2])
    combinedImage[1, 0, 0] = Reform(_rightRight, 1, dims[1], dims[2])
    combinedImage[2, 0, 0] = Reform(_rightRight, 1, dims[1], dims[2])
  endelse

  return, combinedImage
end
```

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: Running this MATLAB code in IDL
Posted by mikejohnryan08 on Tue, 29 Apr 2014 02:04:09 GMT
View Forum Message <> Reply to Message

On Sunday, April 27, 2014 9:20:16 PM UTC-4, Mike Galloy wrote:
> On 4/27/14, 7:00 pm, mikejohnryan08@gmail.com wrote:
>
>> Thanks for the responses! I agree that it would not be a good idea
>
>> to attempt using the MATLAB code in any way towards thesis research,
>
>> and that it would be better to write my own simple IDL function to do
>
>> this.
>
>>
>
>> Mike, thanks for pointing me towards your code library. The
>
>> "combineImages" method from your anaglyph class seems to be EXACTLY
>
>> what I am trying to implement in IDL, and seems like it would be a
>
>> very basic function; since as you said, I already have the left and
>
>> right images. But since I am completely new to IDL, I can't even
>
>> begin to think about how that method could be turned into a
>
>> standalone function. It should be very basic, right?
>
>
>
> Methods have access to the instance variables of the object and can call
>
> other methods of the object using the "self" variable, so watch out for
>
> them. In the _combineImages example, there was only the self.color
>
> variable, so how about something like:
>
>
>
> function mg_create_anaglyph, leftImage, rightImage, color=color
>
>   compile_opt strictarr
>
>
>

```
>    ; define combined_image to the correct size
>
>    combinedImage = leftImage * 0B
>
>    dims = size(leftImage, /dimensions)
>
>
>
>    if (keyword_set(color)) then begin
>
>      combinedImage[0, *, *] = leftImage[0, *, *]
>
>      combinedImage[1, *, *] = rightImage[1, *, *]
>
>      combinedImage[2, *, *] = rightImage[2, *, *]
>
>    endif else begin
>
>      _leftImage = byte(total(fix(leftImage), 1) / 3)
>
>      _rightRight = byte(total(fix(rightImage), 1) / 3)
>
>
>
>      combinedImage[0, 0, 0] = Reform(_leftImage, 1, dims[1], dims[2])
>
>      combinedImage[1, 0, 0] = Reform(_rightRight, 1, dims[1], dims[2])
>
>      combinedImage[2, 0, 0] = Reform(_rightRight, 1, dims[1], dims[2])
>
>    endelse
>
>
>
>    return, combinedImage
>
>  end
>
>
>
>  Mike
>
>  --
>
>  Michael Galloy
>
>  www.michaelgalloy.com
>
```

> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
> Research Mathematician
>
> Tech-X Corporation

Thanks, that makes much more sense now. I was confused about where the self.color was coming from but now I understand. This is very helpful advice for kickstarting my research project, I certainly appreciate it.
