Comments on the recently revised IUImageFindMax() function.

D. R. G. Mitchell, Mar 29, 2025.

At some point between GMS 3.5.1 and GMS 3.6.1 Gatan revised the IUImageFindMax() function. This function can find the maximum within an image with sub-pixel precision.

Note: In GMS 3.6.1, this revised function only reported the shifts with single pixel precision rather than sub-pixel precision. I am not sure if this is a bug or intentional. If your work relies on sub-pixel precision, then I would use the original (3.5.1) version of the script.

Typically, it is employed for analysing cross-correlation images to find the maximum intensity, and from this the location of the image centre and the offset between two shifted images can be determined. A number of my scripts use this function, and users running very recent GMSes have reported such scripts no longer running, such as my Stack Alignment script (now updated to address this issue). The error will manifest itself when the script is run (in GMS 3.6.1 and later) as: “Error in Line XYZ, Unable to match this argument to an existing function”, and the offending line containing IUImageFindMax() in the script will be highlighted.

The problem with Gatan’s change is that if I revise a script to work with the new command format, then the script will no longer work on earlier versions of GMS. The solution is then to create two versions of the script. I have done this for my Stack Alignment script, while working out how the revised IUImageFindMax() function works. However, I do not propose updating other scripts. This note is intended to help users who encounter this error in other scripts fix it for themselves.

If the script exhibiting the error is a standalone (.s) script file, then you can open in it in DigitalMicrograph and directly edit it. If the script is part of a package, you will not have direct access to the script file. In which case email me (via my website) and I will make the source file available.

Syntax for IUImageFindMax() in GMS 3.5.1 and earlier:

Boolean IUImageFindMax(img, top, left, bottom, right, xshift, yshift, origin)

Boolean – can be ignored – presumably returns a 1 if the centre is found – 0 if not
(note img is an image variable, all other variables are number)

img: the image (or sub-area therein) being analysed.
top, left, bottom, right: the pixel coordinates of either the image, or a sub-region within that image, to be analysed. Note top left in Gatan image coordinates is (0,0) and bottom right is (image y size, image x size)
xshift, yshift: this is how much (in pixels) the measured centre is shifted from the origin (geometric centre of the image).
origin: set this to 1

Revised syntax for IUIMageFindMax() in GMS 3.6.1:

Boolean IUImageFindMax(img, top, left, bottom, right, halfWidth, xshift, yshift, maxValue)
Boolean: as above
img: as above
top, left, bottom, right: as above

halfwidth: this is a new variable and is half the x size of the image (or half the y size – image must be square) ie if the image is 1024 pixels wide, then the halfWidth is 512). When editing a pre-existing script, if it is not apparent what the image x size is from the code immediately above the command, you can add the following code immediately above the IUImageFindMax() command ie to source it (note comments are italicized – code is bold):

> // Add this to get the halfWidth (which is the image x size / 2). Choose unique variable names for the x and y size variables you are adding. If you choose something obvious like xsize and ysize, such names will almost certainly have been used earlier in the code and an error will occur (you cannot declare the same variable twice with the same section of code).
>
> **number myuniquexsize, myuniqueysize** // *declare new image size variables*
>
> **getsize(img, myuniquexsize, myuniqueysize)** // *replace img with the image variable name used in the IUImageFindMax() command*
>
> **number halfWidth= myuniquexsize/2** // *determine the halfWidth*

xshift, yshift: these are different to the previous function. Here, the reported shift values are in absolute pixel coordinates, ie relative to (0,0) and not to the image centre. This necessitates some revision of these values when editing pre-existing scripts (see below).
maxValue: this returns the maximum value located at the xshift, yshift position. This is new to the revised function. In cross-correlation scripts this value is the cross-correlation coefficient and ranges between 0 (no cross-correlation) and 1 (excellent correlation). In my Stack Alignment script I used an alternative command to get this value:

maxval=max(crosscorrimg, posx, posy)

This command will find the position of the maximum in an image, but only to pixel precision. So I used it to get the peak intensity (maxval – the cross-correlation coefficient) and ignored the location it returned for this maximum (posx, posy), instead using IUImageFindMax() to get the position with higher precision. In this case a variable (maxval) has already been declared and can simply be reused in the IUIMageFindMax() function which immediately follows. However, in a script where no such variable existed, then one would need to be declared above the IUIMageFindMax() function ie

number myuniquemaxval

Putting all this together in the case of my Stack Alignment script – starting at line 643:

Original Code:

maxval=max(crosscorrimg, posx, posy)
number xshift, yshift
IUImageFindMax(crosscorrimg, 0, 0, refysize, refxsize, xshift, yshift, 1)
xshift=-xshift
yshift=-yshift // *the original function returns the centre position as a shift relative to the geometric centre (origin) of the image.*

Revised Code to be compatible with GMS 3.6.1 and later:

maxval=max(crosscorrimg, posx, posy)
number xshift, yshift
number halfWidth=refxsize/2
IUImageFindMax(crosscorrimg, 0, 0, refysize, refxsize, halfWidth, xshift, yshift, maxval)
xshift=halfWidth-xshift // *the revised function returns the centre position in*
yshift=halfWidth-yshift // *absolute image pixel coordinates (not relative to geometric centre)*

If you get stuck editing one of my scripts while trying to incorporate this revised function – please get in touch through my website. If you are not game to have a go at editing code, you can always run the script on an older i(GMS 3.5.1 and earlier) instance of GMS – most folk have an old laptop or desktop they can use. If you run any kind of virtualisation eg Parallels, Proxmox etc, you can simply create multiple Windows 10 virtual machines and install various GMSes on them.