SOFTWARE TOOL ARTICLE

ImageM: a user-friendly interface for the processing of multi-dimensional images with Matlab [version 1; peer review: 2 approved with reservations]

David Legland ^{1,2}, Marie-Françoise Devaux ¹

¹INRAE, UR BIA, F-44316 Nantes, France

²INRAE, PROBE Research Infrastructure, BIBS Facility, F-44316 Nantes, France

V1 First published: 30 Apr 2021, 10:333
<https://doi.org/10.12688/f1000research.51732.1>

Latest published: 30 Apr 2021, 10:333
<https://doi.org/10.12688/f1000research.51732.1>

Abstract

Modern imaging devices provide a wealth of data often organized as images with many dimensions, such as 2D/3D, time and channel. Matlab is an efficient software solution for image processing, but it lacks many features facilitating the interactive interpretation of image data, such as a user-friendly image visualization, or the management of image meta-data (e.g. spatial calibration), thus limiting its application to bio-image analysis.

The ImageM application proposes an integrated user interface that facilitates the processing and the analysis of multi-dimensional images within the Matlab environment. It provides a user-friendly visualization of multi-dimensional images, a collection of image processing algorithms and methods for analysis of images, the management of spatial calibration, and facilities for the analysis of multi-variate images. ImageM can also be run on the open source alternative software to Matlab, Octave.

ImageM is freely distributed on GitHub:

<https://github.com/mattools/ImageM>.

Keywords

image processing, matlab, multivariate image analysis, segmentation, interactive segmentation

Open Peer Review

Approval Status ? ?

	1	2
version 1		
30 Apr 2021	view	view

1. **John Anthony Jose**, De La Salle University, Manila, Philippines

Janos Lance Tiberio, De La Salle University Manila, Manila, Philippines

2. **Curtis Rueden** , Laboratory for Optical and Computational Instrumentation, Center for Quantitative Cell Imaging, University of Wisconsin, Madison, USA

Any reports and responses or comments on the article can be found at the end of the article.

This article is included in the **NEUBIAS - the Bioimage Analysts Network** gateway.

Corresponding author: David Legland (david.legland@inrae.fr)

Author roles: **Legland D:** Conceptualization, Methodology, Software, Writing – Original Draft Preparation; **Devaux MF:** Conceptualization, Investigation, Writing – Original Draft Preparation

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2021 Legland D and Devaux MF. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Legland D and Devaux MF. **ImageM: a user-friendly interface for the processing of multi-dimensional images with Matlab [version 1; peer review: 2 approved with reservations]** F1000Research 2021, **10**:333 <https://doi.org/10.12688/f1000research.51732.1>

First published: 30 Apr 2021, **10**:333 <https://doi.org/10.12688/f1000research.51732.1>

Introduction

Modern imaging devices provide a wealth of data often organized as images with many dimensions, for instance 2D/3D, time and channel. This leads to new challenges in the exploration, visualization and processing of multi-dimensional data. In particular, multi-spectral images require the application of methods at the intersection of spectroscopy and image processing.

Matlab (The Mathworks, Natick, MA), is an efficient software solution for image and signal processing that provides native support for multi-dimensional arrays, a large number of image processing methods, and a great facility for adding custom developments. However, it lacks many features for facilitating the interactive interpretation of image data, such as a user-friendly visualization of multidimensional images, or the management of image meta-data (e.g. spatial calibration), thus limiting its application to bio-image analysis.

The ImageJ/Fiji software (Schneider et al., 2012) is an alternative that provides an intuitive graphical user interface for image exploration, and many image processing functions that take into account spatial or channel calibration. The basic functionalities can be extended by a large collection of plugins. The integration of custom developments is possible via macros or plugins, but this often requires advanced programming skills.

The ImageM software aims at providing a user-friendly interface for the interactive exploration, processing and analysis of multi-dimensional imaging data within the Matlab environment, while taking into account meta-data and facilitating the integration with custom algorithm developments.

Methods

Implementation

ImageM strongly relies on Matlab's Object-Oriented Programming. The core feature is an "Image" class that handles arrays up to five dimensions, corresponding to three spatial dimensions, the channels, and the time. It also encapsulates image meta-data such as spatial calibration, image name, or channel names.

The user interface relies on various image viewers adapted to the type (grayscale, color, label, multivariate) and the dimensionality of the image (Figure 1a). Interactive tools allow for quick image exploration, for example histograms, line profiles and 3D visualization. The architecture was kept modular to facilitate the inclusion of new features.

Figure 1. Result of a typical image processing workflow with ImageM. Original 3D image is visualized via adapted viewer (a). Image processing operators are applied (b), using contextual dialogs (c). Segmentation results are represented by label images (d,g). Analysis of the image results in table frame (f). Menu actions are transcribed into runnable Matlab commands (e). Original image courtesy of K. Belcram (Moukhtar et al., 2019).

Operation

To have the whole functionalities, Matlab (v2020a) is required, as well as the Image Processing and the Statistics toolboxes. The Gui Layout Toolbox (Sampson, 2021) is required to benefit from flexible image viewers. ImageM runs on a typical workstation. The application is launched by typing “ImageM” from the command line.

Menus provide quick access to common image processing operators, organized in a way to make them easily discoverable by a non-expert user. When necessary, intuitive dialogs allow for setting up parameters. Preview of the results facilitates the choice of relevant settings. Region of interest may be defined interactively to perform interactive measurements.

To facilitate usage during scripting and integration with custom developments, images can be easily imported from and exported to the workspace. Applying an operator results in a log entry that can be integrated into a script for quickly generating a workflow from the succession of operations.

Segmentation and analysis of regions

ImageM integrates a large family of image processing operators (linear filtering, mathematical morphology, segmentation). Intuitive dialogs allow tuning of parameters and preview of the result, facilitating the choice of relevant settings (Figure 1c). The type of the result image is automatically inferred to provide the most adequate visualization (Figure 1d and e).

Segmentation algorithms result in binary or label images that can be used for quantitative analysis of regions. The spatial calibration of images is considered when available. Features can be visualized as geometric overlays on the image or exported to data tables (Figure 1f).

Multivariate image analysis

Multi-channel images or microscopy images contain pixels represented of several values. Multivariate image analysis provides a convenient formalism to analyze such images by taking account both the spatial and the spectral dimensions (Geladi & Grahn, 2006). ImageM can convert multi-variate images into data tables, allowing their exploration through multi-variate analysis tools such as Principal Component Analysis or k-means clustering (Figure 2). Resulting tables can be easily back-converted to images to facilitate the visual interpretation.

Octave

ImageM can also run within the open-source platform, Octave, after some modification of the code (see *Extended data* (Legland & Devaux, 2021)).

Use case

We present here a use case describing image segmentation and region analysis of the 3D image presented in Figure 1. The *Extended data* (Legland & Devaux, 2021) contains the sample files, and more detailed use cases.

- First, run the application by typing “ImageM” from Matlab command line.
- In the menu, select “File->Open Image ...” and choose the file “arabidopsis-embryo.tif”. This opens an image viewer for the 3D image.

Figure 2. Multivariate image analysis of a macrofluorescence image. Left panel: original multivariate image (from Devaux, 2008). Middle panel: application of K-means clustering on principal components. Right panel: color representation of k-means classes.

- Reduce the noise in the image, by selecting “Process->Gaussian Filter ...”. This opens a dialog.
- Choose value 1 for each of the “sigma” values, and click “OK”. The filtered image appears in a new viewer.
- Perform 3D segmentation of the cells. Select menu entry “Process->Extended Min Watershed ...”. Choose a basin dynamic equal to 6, the “C6” connectivity, and result type as “basins”. This results in a new 3D label image in a new viewer, where each label is displayed with a different color.
- Some regions do not correspond to cells. They can be removed manually. Select the menu “Process->Replace Value(s) ...” and fill in with “1, 2, 28, 29” (values can be identified by moving the mouse cursor and inspecting the status bar).
- Morphometric features can be computed from 3D label images, by selecting the menu “Analyze->Analyze Regions ...”. After choosing the features and clicking “OK”, the selected features appear in a new data table frame that can be saved as a text file or exported to the workspace.
- The 3D representation of the [Figure 1g](#) can be obtained by selecting the menu entry “View -> Show 3D isosurface”, choosing a “smoothing radius” value equal to 1 and checking the options “Reverse Z-axis” and “Rotate Ox”.

Summary

The ImageM application proposes a convenient user interface for the visualization and exploration of multi-dimensional images within Matlab. It also provides basic support for regions of interests, the management of data tables. Its modular architecture should facilitate the future incorporation of new functionalities. The current work focuses on a better management of regions of interest, and on the inclusion of user plugins.

Software availability

Packaged application available for direction download from: <https://www.mathworks.com/matlabcentral/fileexchange/45847-imagem>

Source code available from: <https://github.com/mattools/ImageM>

Archived source code as at time of publication: <http://doi.org/10.5281/zenodo.4674326> (Legland, 2021).

License: BSD 2-Clause “Simplified” License

Data availability

Underlying data

All data underlying the results are available as part of the article and no additional source data are required.

Extended data

Zenodo: Extended data for the manuscript “ImageM: a user-friendly interface for the processing of multi-dimensional images with Matlab”, <http://doi.org/10.5281/zenodo.4705240> (Legland & Devaux, 2021).

This project contains the following extended data:

- User manual
- Octave compatibility notice
- Two detailed use cases
- Images used in use cases

Data are available under the terms of the [Creative Commons Attribution 4.0 International license](#) (CC-BY 4.0).

Acknowledgements

This publication was supported by COST Action NEUBIAS (CA15124), funded by COST (European Cooperation in Science and Technology).

References

-
- Devaux M-F: **UV and visible fluorescence images of maize stem: macroscopy and confocal microscopy**. [Data set]. Published 2018 via *Perscido-Grenoble-Alpes*.
[Reference Source](#)
- Geladi P, Grahn HF: **Multivariate Image Analysis**. In: *Encyclopedia of Analytical Chemistry* (eds Meyers RA, Meyers RA). 2006.
- Legland D: **mattools/ImageM: ImageM v1.3.2.1 (Version v1.3.2.1)**. *Zenodo*. 2021, April 9.
[Publisher Full Text](#)
- Legland D, Devaux M: **Extended data for the manuscript "ImageM: a user-friendly interface for the processing of multi-dimensional images with Matlab" (Version 1.0)**. *Zenodo*. 2021, April 20.
[Publisher Full Text](#)
- Moukhtar J, Trubuil A, Belcram K, *et al.*: **Cell geometry determines symmetric and asymmetric division plane selection in Arabidopsis early embryos**. *PLOS Computational Biology*. 2019; **15**(2): 1–27.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Sampson D: **GUI Layout Toolbox**. *MATLAB Central File Exchange*. Retrieved April 13, 2021. 2021.
[Reference Source](#)
- Schneider CA, Rasband WS, Eliceiri KW: **NIH Image to ImageJ: 25 years of image analysis**. *Nat Methods*. 2012; **9**(671–675): 2012.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 09 September 2021

<https://doi.org/10.5256/f1000research.54926.r92537>

© 2021 Rueden C. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Curtis Rueden

Laboratory for Optical and Computational Instrumentation, Center for Quantitative Cell Imaging, University of Wisconsin, Madison, Wisconsin, USA

ImageM is a MATLAB-based application, backed by a modular collection of MATLAB libraries, for performing image processing and analysis on multidimensional scientific images with the common dimensions of X, Y, Z, channel, and time. Modeled closely after ImageJ, ImageM integrates many powerful routines into a central main window, making them more easily accessible, especially to those less versed in writing scripts.

The use cases do a nice job of quickly showcasing some of the great things you can do with ImageM. However, the article's introduction and rationale needs work, and the program needs to be better documented and tested. Nonetheless, this software is certainly worth advertising to the scientific community.

> Is the rationale for developing the new software tool clearly explained?

Partly. I have two problems with the rationale as written:

- The claim that ImageM addresses novel visualization challenges:

> Modern imaging devices provide a wealth of data often organized as images with many dimensions, for instance 2D/3D, time and channel. This leads to new challenges in the exploration, visualization and processing of multi-dimensional data. In particular, multi-spectral images require the application of methods at the intersection of spectroscopy and image processing.

Challenges in visualization of 3D, time and channel are not "new". ImageJ's core gained 5D support in 2007 with version 1.39I, and there were many earlier programs with ≥ 5 D support as well (e.g. BioImageXD, VisBio, ImageJ via the Image5D plugins, OME Server).

Highlighting multi-spectral as a challenge makes sense, but it is hardly the only dimension-

related challenge of scientific imaging. Multiple angles? Fields of view? Fluorescence lifetime?

ImageM is limited to 3D+time+channel. Therefore, it does not facilitate handling of any "new challenges" in dimensionality beyond this paradigm.

Rather than argue that ImageM addresses new challenges in scientific imaging, the article would be better served explaining that ImageM builds on the wealth of functionality available in MATLAB, enabling more seamless integration of other MATLAB-based tools with its 5D image paradigm. The two use cases, especially use case 2, illustrate this fact well.

- The lack of discussion of the technical merit of a MATLAB-based implementation of an ImageJ-like program:

While this is a short article, it would be ideal—if sufficient space—to mention ImageJ's MATLAB integrations ImageJ-MATLAB (<https://imagej.net/scripting/matlab>) and MIJ/Miji (<https://imagej.net/plugins/miji>), and how ImageM compares to them. This is an important part of the justification for essentially reimplementing ImageJ in MATLAB: what are the downsides of using the existing ImageJ within MATLAB via its built-in JVM? How does ImageM overcome those downsides?

I do believe there is technical merit to ImageM. But the article needs to do a clearer job explaining the benefits.

> it lacks many features for facilitating the interactive interpretation of image data, such as a user-friendly visualization of multidimensional images or the management of image meta-data (e.g. spatial calibration), thus limiting its application to bio-image analysis.

I am not well versed in the ecosystem of MATLAB add-ons, so I cannot evaluate whether this statement is really true, but I am skeptical. Consider rewording to frame things more positively, saying that ImageM provides user-friendly capabilities that build upon, enhance, and complement MATLAB's existing image processing offerings.

> The integration of custom developments is possible via macros or plugins, but this often requires advanced programming skills.

I disagree about MATLAB programming being easier than ImageJ macro programming. Both are designed to be as accessible as possible, while still being powerful. Instead of arguing that extending ImageJ is too hard, why not instead argue that extending ImageM via MATLAB code enables scripters to easily harness MATLAB's power, flexibility, and extensive suite of toolboxes and add-ons?

- > Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes. Use case 1 was fully reproducible for me. Use case 2 was also fully reproducible, but:

- There appears to be some randomness to the K-Means clustering output. My results, while functionally identical, differed from the screenshot regarding the cluster numbering.

- While following the steps, the following error is emitted to the command window:

Error using convhull

Error computing the convex hull. The points may be collinear.

Error in Table/scatterGroup (line 204)

```
inds2 = convhull(xdata(inds), ydata(inds));
```

Error in imagem.actions.table.process.TableKMeans/run (line 88)

```
scatterGroup(tab(:,1), tab(:,2), k);
```

Error in imagem.gui.FrameMenuBuilder>@(src,evt)action.run(obj.Frame) (line 475)

```
'Callback', @(src, evt) action.run(obj.Frame));
```

Error while evaluating Menu Callback.

- Folding the k-means score table back to an 800x800 image does not emit a line of script for programmatic reproduction.

All of that said, the use case of K-means clustering to segment tissue types is very cool!

Relatedly, ImageM is hard to install:

- I could not get the ImageM Octave fork working in Octave. I recommend removing the claim that ImageM works with Octave. (Which is a shame, because a major problem with MATLAB is that it is not open source.)
- I could not get ImageM working in MATLAB Online. This isn't too surprising, but may want to specifically acknowledge that it doesn't work in the online app.
- I could not get ImageM working in a local MATLAB R2021a after installing it from the Add-On Manager, even after also installing the add-on dependencies MatImage, geom2d, geom3d, and MatStats. Is there really no dependency management system for MATLAB add-ons? If not, that is a huge downside of using MATLAB compared to Java, Python, JavaScript, Ruby, or other popular language ecosystem.
- I was finally able to get ImageM to work by opening the ImageM-1.3.2.1-full.mlappinstall bundle downloaded from the GitHub releases page. It added a working ImageM icon to MATLAB's APPS tab. The documentation—especially the GitHub repository's README.md and ImageM page on MATLAB Central—needs to explain in much more detail how to install this software.
- I continued to receive startup errors about the MATLAB path due to backslashes in the path strings. Based on the errors I received trying to run it with backslashes in paths, it seems like ImageM is only tested on Windows. I was testing ImageM on Ubuntu Linux 20.04 LTS.

More generally, ImageM's documentation is lacking:

- Why is the manual only a PDF? Why not online as HTML or Markdown?
- There is no way to quick-launch commands from the menu. A command finder or search bar would be very helpful, especially as people start developing ImageM extensions that

add to the menu structure.

- The Help menu should offer links to important resources, especially the online docs.

Finally, a couple of miscellaneous comments:

- Mathworks named the program MATLAB in all caps ("MATLAB"), not title case ("Matlab").
- From the ImageM manual:
 - > An element of an image is usually associated to a square region centered on its coordinates.
 - > Its bounds are therefore ± 0.5 around each dimension.

A pixel is not a little square. http://alvyray.com/Memos/CG/Microsoft/6_pixel.pdf

As mentioned above, I had a hard time getting ImageM installed. Here are the issues I filed on GitHub to give some technical detail:

- MATLAB Online – <https://github.com/mattools/ImageM/issues/1>
- Octave – <https://github.com/mattools/ImageM/issues/2>
- MATLAB R2021a – <https://github.com/mattools/ImageM/issues/3>

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Software engineering, software architecture, image visualization, image processing, image analysis.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 23 July 2021

<https://doi.org/10.5256/f1000research.54926.r89355>

© 2021 Jose J et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

John Anthony Jose

Electronics and Communications Engineering Department, De La Salle University, Manila, Philippines

Janos Lance Tiberio

De La Salle University Manila, Manila, Philippines

The authors are able to clearly explain the problem and contribution of the paper. However, the method of how they validated their contribution is quite limited. Specifically, they mentioned that the problem they would like to solve is that the existing approaches on visualizing and exploring multi-dimensional images lacks features in interactive interpretation. Concretely, it is about the lack of user-friendly visualization and management of image meta-data. However, their results didn't provide any validation that their contribution is has better user-friendly approaches compared to the existing one.

The authors can improve on their article by concretely discussing on how their results was able to directly address the problem that they are trying to solve. Additionally, it will be helpful if they provide comparisons with existing software for multi-dimensional image analysis.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Computer Vision, Image Processing

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research