Instructions to Just Download the MATLAB Files:

1. Click on the ‘Files’ heading at <https://sourceforge.net/projects/spoketoolbox/>
2. Download the .zip file labeled ‘High Speed Image Analysis’
3. Unpack the .zip file in your MATLAB user directory
   1. for most Windows users this will be the MATLAB folder in ‘My Documents’
   2. If you’re not sure, type ‘userpath’ at the MATLAB command prompt.
4. In MATLAB, go to File 🡪 Set Path…, click the ‘Add with Subfolders…’ button, and select the location where you unzipped the High Speed Image Analysis package. Click ‘Save’ and ‘Close’. You’re now ready to start using the files! If you have questions, contact [msmcdon@umich.edu](mailto:msmcdon@umich.edu).
5. The .zip file will periodically be updated as the code is improved, so check back occasionally!

Instructions to Analyze the Sample Thruster Video

1. Run ‘IntegratedFASTCAMAnalysis.m’. There are three ways to do this:
   1. Open the file in the Editor window and click the‘Run’ button (the green arrow icon in the toolbar)
   2. Or, open the file in the Editor window and press the F5 key
   3. Or, just type ‘IntegratedFASTCAMAnalysis()’ (no quotes!) at the command line
2. The program will prompt you to select a video file. Download the .MAT video file in the Sample Data File folder on the project SourceForge page and select the downloaded file.
   1. The program first crops the image to analyze only the region of the actual discharge channel. Pre-set options are included for some thrusters, you may select a default option to analyze between 75% and 125% of the mean discharge channel diameter, or you may perform your own fit using a built-in GUI. All cropping is aided by the function ‘AutoCircleFit.m’
   2. After selecting a thruster and if necessary selecting the discharge channel region to identify, four windows will pop up:
      1. First, a view of the mean image from the whole video overlaid with the calculated thruster center as a large yellow dot, mean channel diameter in black, and inner and outer diameters for analysis in red.
      2. Second, the first millisecond of the video’s spoke surface
      3. Third and fourth, the broken-out 2D DFTs of the spoke surface, calculated in both the clockwise and counterclockwise directions. Since spokes switch directions with B-field polarity, it is worth it to always compute both directions just to make sure you’re capturing any spokes.
3. The sample video is short to allow for easy downloading. However, this also makes the DFT resolution very coarse. To illustrate a finer resolution DFT, the ‘Sample Data Files’ directory also includes a longer spoke surface from a 9000-frame video. To look at this spoke surface:
   1. Open ‘Video2DDFT.m’ and run it. The program will prompt you to select the spoke surface file. Make sure to select the downloaded spoke surface file from the Sample Data File directory, and not the raw video file we analyzed earlier.
   2. If desired, change the value for ‘reverse’ around line 9 of the code to compute the DFT in the reverse direction.

Instructions to Analyze Your Own Thruster Video

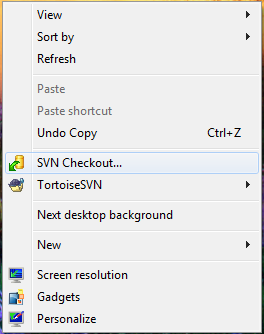
1. The toolkit was designed to import .MAT files primarily. The code can accept Photron .MRAW files and Photron- or Phantom-generated multipage TIFF files and convert them to the MATLAB .MAT format automatically. Once you’ve imported into the .MAT format, it will be faster to use that file format for future analysis since MATLAB is optimized to import .MAT files quickly. Speed can be important for multi-gigabyte datasets.
2. As with the sample files, run IntegratedFASTCAMAnalysis.m on your .MAT, .MRAW or .TIF video file. The conversion to .MAT can take a while; be patient. It will print status updates as it goes.
   1. If you would rather not isolate the channel or AC couple the video, or would like to normalize out the breathing mode in the spoke surface, those options may be selected near lines 40-44 of ComputeSpokeSurface.
3. Finally, as you analyze your own files, you can get an idea of which of the toolbox files call which others using MATLAB’s dependency reports and profiling tools, or by the table below which summarizes the major parent functions and their child function calls:

|  |  |
| --- | --- |
| **Parent Function** | **Child Function(s)** |
| IntegratedFASTCAMAnalysis | ComputeSpokeSurface  Video2DDFT  File Import and Conversion\ImportMatVideo  Utilies\RemoveFileExtension |
| [ComputeSpokeSurface](matlab:%20edit%28%27ComputeSpokeSurface.m%27%29) | [AutoCircleFit](matlab:%20opentoline%28%27ComputeSpokeSurface.m%27,105%29) [ChannelThetaBin](matlab:%20opentoline%28%27ComputeSpokeSurface.m%27,126%29)  thetafnX  Utilities\RemoveFileExtension  File Import and Conversion\ImportMatVideo  GUI\PixelPolarCoords |
| AutoCircleFit | [GUI\TaubinSVDThresholdFit.m](matlab:%20opentoline%28%27AutoCircleFit.m%27,17%29) |
| ChannelThetaBin | [GUI\TaubinSVDThresholdFit.m](matlab:%20opentoline%28%27AutoCircleFit.m%27,17%29),  [GUI\PixelPolarCoords.m](matlab:%20opentoline%28%27ChannelThetaBin.m%27,13%29) |
| Video2DDFT | do\_FFTs  Utilities\lorentzianfit  Filtering\ConfidenceLevelCrop  Filtering\ConfidenceLevelInterval  Filtering\ConfidenceLevelPlot |

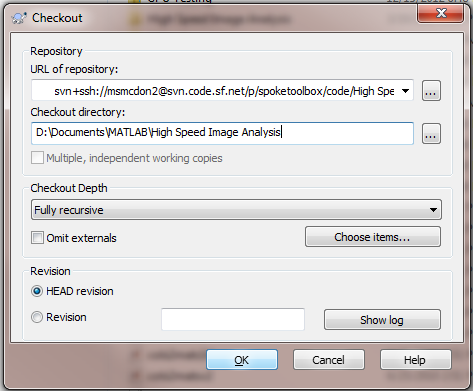
Instructions for Contributing to Development of the High Speed Image Analysis Toolbox:

This toolkit grew out of the dissertation of now-Dr. Michael McDonald at the University of Michigan Plasmadynamics and Electric Propulsion Laboratory. However, its continued usefulness to the electric propulsion and plasma community will depend on users to add features they see as necessary to the package and to share them with their peers. If you have made changes to the code that you feel make it more useful, easier or more efficient, please contribute your changes to the repository to aid future users. To do so:

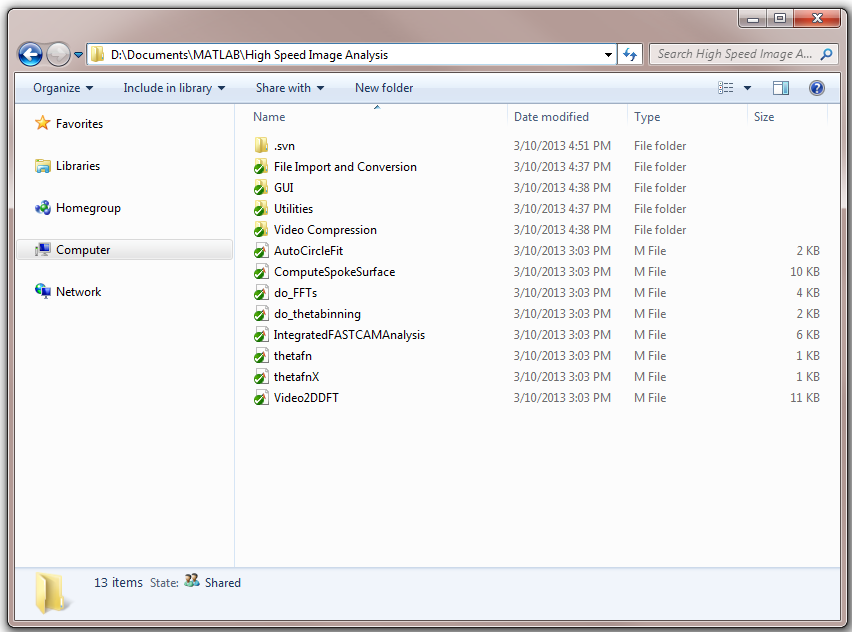
1. Create an account at [www.sourceforge.net](http://www.sourceforge.net) (SourceForge is a hosting site for code projects)
   1. E-mail [msmcdon@umich.edu](mailto:msmcdon@umich.edu) with your username for read/write access to the project
2. Download and install TortoiseSVN from <http://tortoisesvn.net/downloads.html>. TortoiseSVN is a revision control program allowing multiple contributors to collaborate on software without erasing each others’ work.
3. Download the PuTTY Key Generator (puttygen.exe) and Pageant Key List (pageant.exe) from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. These programs let TortoiseSVN talk to SourceForge to retrieve and update files without a lot of really annoying password re-entry.
4. Generate an SSH key using PuTTY:
   1. Execute the PuTTY Key Generator (puttygen.exe) program.
   2. Select "SSH2 DSA" as the desired key type in the "Parameters" section.
   3. Click "Generate"
   4. Follow the instructions by jiggling your mouse randomly in the blank area of the window. Key generation will occur once PUTTYGEN has collected sufficient random data from your movement.
   5. Enter *YourSourceforgeUsername*@svn.code.sf.net for the key comment, entering your actual username in place of *YourSourceforgeUsername*
   6. (Optional) Enter a passphrase in the "Key passphrase" and "Confirm passphrase" fields. This makes it so no one could hijack your computer and use your private key to do mischief.
   7. Click "Save private key". Choose a suitable directory and optionally rename the key to something memorable, like "SourceForge-Shell.ppk". The .ppk extension is used for PuTTY Private Key files.
   8. Go to the SSH key posting page (<https://sourceforge.net/account/ssh>). Copy your public key data from the "Public key for pasting into OpenSSH authorized\_keys file" section of the PuTTY Key Generator window and paste it into the provided form on the SourceForge.net site. Click the "Update" button to complete the posting process.
   9. Exit the PuTTY Key Generator (puttygen.exe) program.
   10. The key data will sync from the SourceForge.net site after a short delay (minutes)
5. Start pageant.exe and add your just-generated private key
6. Open your “My Documents\MATLAB” folder
7. Make sure your SourceForge username has been added to the High-Speed Imaging Project (i.e., make sure you’ve completed step 1). Then right-click in the MATLAB directory and select “SVN Checkout…”



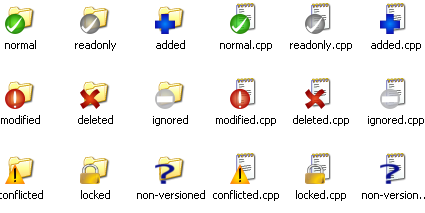
1. In the window that comes up, the repository URL is: svn+ssh://*YourSourceforgeUsername*@svn.code.sf.net/p/spoketoolbox/High Speed Image Analysis



1. You should now have a directory that looks like below. The checkbox icons may not appear until after you reboot.



TortoiseSVN icons:



* Newly checked-out working copies get a green check. This is *Normal*.
* Once you change a file, the status is *Modified* and the icon changes to a red exclamation point. This indicates which files have changed since last updating your working copy from the repository. To save your changes to the repository, right-click the files and select ‘Commit’.
* If a conflict occurs during an update (say you and someone else checked out a copy at the same time and changed the same parts of the code), then the icon becomes an exclamation mark with a triangular yellow shield. In this case you’ll need to manually tell Subversion what changes to accept.
* If you delete a file, it will get a red X and the next time you commit to the repository it will be removed.
* If you create new file, it will get a blue exclamation point until you right-click to add it to the version controlled code
* Once you add a file, it gets a blue plus sign until you commit the changes to the repository, at which point it gets a green check mark (normal)