Introduction

We are delighted that you have chosen Team Hindsight for your business needs and are excited for you to use our software. The Hindsight software is a productive system for analyzing dust density in abrasion images that has been developed to suit your needs.The purpose of this user manual is to help you, the client, successfully install, administer, and maintain the Hindsight software product in your actual business context going forward. Our aim is to make sure that you are able to use our product for the M2020 mission cycle.

How to install/Run

Requirements:

- Matlab version 2017b or higher
- Python 3.6
- Windows, linux, (Mac is buggy with user interface colors, but still technically works if you press the non-colored buttons)

Installation:

- Open command line as Administrator
    - In command line navigate to matlab root:
        - You should be able to find your matlab root directory by opening matlab and typing "matlabroot" into the console
    - In the regular command line(not in matlab) navigate to matlab root:

    Program Files\MATLAB\R2017b\extern\engines\python

    - Next run the setup.py to get matlab engine to run in python:

    Program Files\MATLAB\R2017b\extern\engines\python> python setup.py install

- Go to Hindsight GitHub repository at: https://github.com/Beckcjb/Hindsight
    - Select “Clone or Download” dropdown and select “Download”. This will download all the files from the repository.
    - Unzip Hindsight-master to location of choice
    - In setup.py file edit line 26 to show current location of MATLAB python engine
        - Replace matlab_path string in line 26 with location of MATLAB extern engines python

Program Files\MATLAB\R2017b\extern\engines\python

- From command line interface navigate to the folder in which the Hindsight software is located. (Default directory named Hindsight-master, can be anything you want)
- Run the setup.py file to get a complete installation of packages and environment path setup:
    - C:/User/Programs/Hindsight-master> python setup.py install

Running Software:

- From command line interface navigate to the folder in which the Hindsight software is located.
- Run the index.py file to start MatLab engine and run the Hindsight Software. The following command will start the software
    - C:/User/Hindsight-master> python index.py
- Software has exit button, but if something goes wrong or you want to exit the application before it completes its task type “Ctrl + c” in the command line interface. Not elegant to close this way, we recommend using the exit button on the software.

How to operate

This software was developed with the intention to analyze images of holes drilled by a rover that uses a dust removal tool inside an abrasion. The software utilizes image sets generated inside an atmospheric chamber that represents the Mars atmosphere. This software allows for the continuous testing and calibration of dust removal tools while remaining in the desired atmospheric pressure.

Tools:

The Hindsight software is developed in python using various libraries including Opencv, Tkinter, Pandas, Numpy, and Matplotlib. These tools are used to assist in image processing, data storage, and graphical user interface(GUI) display. The software also integrates MatLab libraries to carry out tasks such as image color segmentation for dust detection in Rock - E and a few other rock types.

How-To-Use:

The Hindsight Image Analyzer software is developed to be fast and easy to understand and is broken up into windows for the graphical user interface(GUI). This communicates with the pandas dataframe that houses the image data necessary to carry out image manipulations to detect dust in the image. To start we will go over the front end of the software which was developed using tkinter.

Browse Window:

The browse window is the initial window that the user will see upon starting the program. This window has two buttons on it and an entry bar. The "Browse..." button will open up a directory browser for the user to select a single image set or any number of image sets (we recommend for fastest results to run a range of 1 to 5 sets a time).
Once the user has highlighted their desired image sets and selected "open" in the directory the user will be returned to the browse window and will need to press the "Select" button to move on to the Analyzer window and send the file path data to the next section of code. The entry box can also be used to manually enter in file paths but after the desired path is enterd the user will still need to press "Select" to move forward. Next is the Analyze window

Analysis Window:

Once the user has selected a number of image sets and pressed "Select" in the Browse window they are directed to the Analyze window. This window is the main section of the software that will be interacted with the most during processing the image sets. On this window the user will see at the top of the frame the file basepath which houses the information for which file the images are in and then below that are the file paths for the individual images. Next to the base path is a "Change File" button that allows the user to select different images for a new run.

In the right of the frame the user is given a drop down menu that allows them to select the rock type that will be tested.On the left of the window there will be a drop down menu that has the option "Color Analysis". The color analysis returns the percentages and colored areas of the abrasion to signify differences in dust coverage.

The user will then be presented with two different detection shapes, the first and most accurate is the circular shape. This will search the abrasion using concentric circles that are defined by the "Band Size" value, we recommend a size of 20 for decent accuracy and good performance. It will take the program significantly longer time to process if the value is decreased, as it will increase the number of bands being created, analyzed, and drawn the the analysis image. The buffer size changes the size of the pixel area that is used to make an average of dust coverage, for this we recommend a value of 5. The next three entry boxes are the pixel position and size of the search area that our algorithms are bound by. The values to move the area left or right go more left closer to 0 more right further than zero. The values for moving the search area up and down correspond to up further from 0 and down closer to 0. Then finally the radius of the search circle can be changed to increase the searched area size.

For the previously mentioned three values we have set base values that are consistent with the positioning of JPL's abrasion images. To check the position and size of the search region pressing the "Apply" button will bring up an image from the set of abrasions selected and have a ring showing where the program will search. Under the "Change File" button there is a "Send Configuration" that is used once all desired image analysis options, values and image sets have been selected. Once the "Send Configuration" button is selected the GUI will create a config class object that allows the Pandas dataframe to access the data needed to analyze the images. This data contains the rock type, run type, base path, and file names. With using the base path and file names the Pandas dataframe will place the image sets into groups. If an image set has more than one "after" image, an image set will be created for each after image.

```
For example, and image set that has two "after" images:
(abrasion020_abraded.jpg, abrasion020_after1.jpg)
(abrasion020_abraded.jpg, abrasion020_after2.jpg)
```

After the image pairs have been generated in the Pandas dataframe and the configuration has been sent to the backend, the user will be able to press the "Run" button. Once this button is pressed the backend will read from the configuration data what type of rock type is being run and what image analysis type is being run. The Pandas dataframe will then apply the necessary functions to the image pairs and then place the result in the set of images.

```
For example:
(abrasion020_abraded.jpg, abrasion020_after1.jpg, analyzed_image_data.jpg)
```

Once the data frame has completed analyzing, the image sets will be displayed with the "after" image and the "analyzed image data" next to each other for each image pair that was run using Matplotlib. Below the analyzed image there are three percentage values that are labeled 'G' for green, 'Y' for yellow, and 'R' for red. This is done to aid in readability and understanding of program output.

If the user is satisfied with the outcome of the test the user can select the "Save" button to save all the "analyzed image data" that was created during the most recent run. This saves the "analyzed image data" to the file where the image sets where selected from. The “Exit” button in the top left of the analysis window closes all windows and resets the command line.

Image Manipulations Overview:

For the purpose of this project we have elected to use OpenCV and MatLab for alterations to the image data so that we can identify dust and its density in multiple rock types. Here we will go over what image manipulations we are doing in order to complete our task. These include: Color Segmentation and Heat Map Analysis

Color Segmentation:

Takes in a colored image converted to the HSV. The MatLab color segmentation script takes out any pixels that are not matching a specific color range, hue, and saturation specified by various thresholds.

Heat Map Analysis:

Takes in a binarized image and colors sections of pixels yellow, green , or red according to the buffer specified in the source code.

Maintenance

Because this product is relatively straightforward (applying mathematical functions to images), maintenance is relatively low. As long as the correct version of python is used and Matlab continues to support python this should work the same in the future. There is a separate section on improving and adding to our software.

Debugging

Running: If you attempt to run the images through the analysis before sending the configuration it will not work. You need to first send the configuration then select run.

Exiting: The exit button should handle closing all windows and resetting the cmd. If there is an error and it messes up upon exit, you can close all windows and reset the cmd by typing “Crtl-c” in the command line interface. Again the exit button on the software should handle this for you.

How to add new rock type analysis

To add a new rock type there are few things you will need to do.

- First: Add new rock type entry to the rock type dropdown menu in the index.py file, Analysis window class, line 172
- Second: Add new function for image processing script of choice in image.py file
- Third: Add new function for new rock type analysis in conrol_funcs.py file
- Optional(fourth): Add function for new rock type analysis types or values to be displayed for the new rock type instead of using the same ones as rock type-E.

Rock type E nuances

In our testing we found that our two types of analysis (circular or square) were more accurate depending on the kind of image we were analyzing.

For example, abrasions that are shallow, and the dust layer is thin, tend to work better with the circular region analysis. (eg abrasion 30, courtesy of NASA/JPL-Caltech)

We believe this is because, although the dust is thin, the color of the pixel still registers as “dust” to our program. However, in your analysis, you tend to ignore dust when it is mostly cleared out of a given region. Furthermore, what you define as a “region” is more fluid than our methods. Instead of concentric circles for example, your regions tend to “move” to make room. That is to say, if the corner of a given region is super dense, but the rest of it is clear, you may choose to make that super dense part its own region. Where as our program would just happily calculate it as being relevant to the entire region.

This is important as the shallower abrasions tend to scatter dust everywhere, and so there will be many regions with “a little bit of dust” scattered in every region. The circle detection has a more reasonable analysis as it just takes the average of large regions, so those small blips of dust don’t have a large effect on the region as a whole (kinda like how you analyze regions).

On the other hand, abrasions that are deep, and have a thicker (easier to see) layer of dust, the square analysis tends to produce better more accurate results (eg abrasion 14, courtesy of NASA/JPL-Caltech).

We believe this is because of a few things. In the images that are better for square analysis, they tend to have shapes that are not concentric circles. So the cleared region is not concentric, the dust tends to clump, and when cleared, it doesn't necessarily clear out evenly in every direction. Meaning that a more fine grained analysis should be better as it looks at smaller regions and calculates the dust coverage of those smaller regions.

General good practice

Reinstallation: To re-install or move the software to another system just follow the steps at the beginning of this document.

Image set sizes: The software can support more than five sets of images. We recommend only doing five image sets to keep runtimes low.

Good lighting: We have agreed upon and recommend using good lighting for the images to get the best dust analysis. Yet, we have also implemented different scripts that handle both lighter and darker regions in the images.

Analysis variables: The values we have placed in the entry boxes for the color analysis are what we recommend for running the software. You have the option to change these values and we have displayed the bounds in which you can enter. If the band size for circular detection shape is small(less than 10) then the software will take a significantly longer time to run. Also the values we have set for the position of the detection area are already position over most abrasion centers. If you feel the need to change these values you can view the changes yourself before running.

When to use circular detection shape: We have found that the circular detection shape has a lower level of variance between results but is slightly less accurate than the square detection shape.The circular method of detection is also slower than square since it is using discrete circles. We recommend that circular be used on the more clear abrasions that do not have odd shapes with in the abrasion.

When to use square detection shape: The square detection shape is faster in analysis and coloring depending on the buffer size you use. We recommend using a buffer size from 5 - 15 as these are the most accurate results. The square detection shape has a higher degree of variance between results, but is more accurate than circular detection. Square detection is faster than circular because it marks regions in the entire abrasion and not circle by circle. We recommend using this analysis for more densely covered abrasions with odd shapes in them.