Imager Sensing Extraction and Easy Integration Tools (ISEEIT)

Michael Wilson
003508374

Abstract

The use of imagers as sensors is becoming more prevalent and more meaningful in studies involving potentially invasive sensor network deployments. Currently available software utilities lack the simple, intuitive software interfaces that could benefit researchers by easily supporting analysis and trend identification within these potentially massive image datasets. Researchers can benefit by using the Imager Sensing Extraction and Easy Integration Tools (ISEEIT) to interface their predefined image feature extraction algorithms with the rich and flexible API provided by the ISEEIT framework.

1. Introduction

The use of imagers as sensors has becoming increasingly useful as the cost and power consumption of imagers decreases through continued engineering process development. In circumstances where other means of measurement might be impossible or extremely invasive, deploying networks of imagers can still enable measurement in these scenarios. The sheer quantity of images collected through this process can be overwhelming, with only a few days worth of images generating thousands of impressions. ISEEIT provides tools designed specifically for use with data from imagers, laying a framework for the processing and analysis of this image data that generalizes to generic image feature extraction tools while providing added benefit over existing image processing packages.

2. Existing Image Processing Software Tools

2.1 General Software Tools

While software tools such as Adobe Photoshop and Mathworks Matlab are general enough to offer most of the features provided through ISEEIT, these software frameworks fail to offer a simple and intuitive API that allows researchers to quickly perform various types of analysis on their image data without extensive code modifications. If a code base is developed in one of these image processing environments, the cost of modification would make the addition of each additional image feature extraction algorithm infeasible. In addition to these additional costs, extra effort would be necessary for development of a rich and flexible viewing environment in which to examine large collections of image data and correlate these data with their extracted feature sets.

ISEEIT addresses these shortcomings by providing a simple and rich API through which researchers can utilize their already existing image feature extraction utilities. The ability to use extraction tools previously developed will benefit researchers by preventing unnecessary duplication of effort. In addition, the cost to include these extraction algorithms in the ISEEIT framework is minimized by the simplicity of the framework interface. The ISEEIT framework also defines access to online image management facilities that have been modified to easily display extracted features in the same frame as the original image data, thereby facilitating verification of the accuracy of the image feature

extraction algorithm and identification of trends within the image data set.

2.2 Specialized Software Tools

Most batch image processing tools currently available, such as ENVI and EyeBatch, limit the ability of the researchers by restricting users to a set of image feature extraction algorithms predefined in the software suite. While some researchers using imagers as sensors will be able to use the algorithms provided by these more specialized image processing tool sets, many researchers will find it useful to be able to define highly-specialized image extraction algorithms that cannot be synthesized using the tools provided by the specialized software suite. In addition, the added cost of moving to a new software suite for image processing would serve as a barrier for researchers using image extraction algorithms personalized to their line of research.

The ISEEIT framework overcomes the limitations of these specialized software tools by providing researchers with the ability to utilize their current image feature extraction algorithms without restricting the scope of the algorithms or requiring excessive overhead for integration of these algorithms. Without restrictions on the methods that they can use to extract the feature data from their image sets, researchers will benefit from the ease with which they can use the ISEEIT framework in conjunction with their processing tools.

3. The ISEEIT Framework

3.1 Framework overview

The basic framework of ISEEIT consists of a series of scripts written in Python that can be utilized directly or included in future automated image processing utilities. The framework exists in a careful balance between the specialization of image data feature set extraction tools and the generalization of arbitrary feature set extraction specification. Users will process their image data sets using ISEEIT in conjunction with their own defined image feature set extraction script. The script needs to adhere to the flexible API defined by ISEEIT, but that is the only restriction on the ability of the user-defined extraction tools. The processing script collects the output from the extraction tool and attaches that output to the metadata of the image. The metadata is uploaded to Picasa Web Albums, where the visualization script will process the metadata for the required feature set visualization.

Researchers benefit from the choice of Python as the language for the integration of these extraction tools by being able to take advantage of the various libraries that have been developed to interface Python with many existing application engines. For example, if an extraction algorithm has already been defined in Matlab, there exists a simple interface solution library that allows Python to access the Matlab engine. In addition, Python provides powerful processing capabilities with a low cost to learn.

After uploading the images and the extracted data, researchers can analyze the image set and the extracted data from any computer using Firefox. The visualization script runs in the browser to modify the appearance of the Picasa Web pages that contain each image and the agreed data. The modification requires that the attached data be read and interpreted by the script, after which the script will import the visualization into the web document directly. This technique can be applied at the individual image level or at the album level, displaying a visualization measured across the entire dataset.

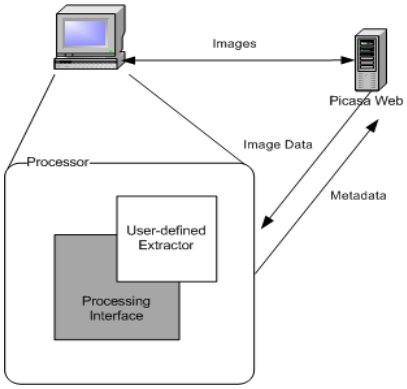

Figure 1: ISEEIT Processing Structure

3.2 Extractor Integration

Researchers benefit by the ability to define custom extraction algorithms and specify which algorithm to use for processing each batch of image data. The integration of these extractors defines a flexible API that provides the ability to perform preprocessing activities, then to output, for each input image, short tags and longer references. The short tags provide the ability to index over the image data set, labeling each with independent tags or combining them in meaningful groups. The longer references provide the information to be read by the visualization script, and can be used to define the parameters of visualization for each image.

3.3 Feature Set Visualization

The feature set visualization options are as flexible and rich as the extraction integration options. The visualizer is written in JavaScript and used directly by GreaseMonkey, a free add-on to Firefox. Although the required use of GreaseMonkey as part of ISEEIT reduces portability, this cost is repaid through the benefit of being able to take advantage of the intuitive interface offered by Picasa Web for image and metadata management.

The visualization script first reads the feature set data from the comments section of the Picasa Web page. This data is most likely a URL source for an image representation of the data, such as a chart displaying information about the characteristics of the image. The extraction tools can make use of a Python library that will encode the data in a URL to take advantage of the Google Chart API. After reading this data, the image referenced in the feature set data can be added to the document of the page in such a way that the image and the resulting data can be analyzed in the same frame.

The visualization script can also modify the image viewing platform in ways that might made the visualization more clear or more meaningful, such as resizing the image or moving aside some of the unnecessary metadata in the sidebar. All of these options can be provided by setting parameters passed from the extraction tools in the feature set data for each image. As a result, only minor modifications to the visualization script will be needed as options in the future are added to the abilities of the extraction tools. Also, the ability to include an endless number of comments that can be attached to each image provides the ability to include an endless number of visualizations for each image.

4. Conclusion

There are quite a few software frameworks available for the processing of digital imagery and the presentation of the resulting image feature sets. Unfortunately, these software suites are either too general or too specific, forcing researchers to spend too much time building extra code to take advantage of the features of the suite or modifying their extraction algorithms to fit the limitations of the software. ISEEIT benefits researchers by providing a framework that is specific to this application while remaining general to the type of extraction and visualization that the researcher envisions. The applicability of this framework is magnified through the availability of a rich and flexible API to the extraction and visualization elements. For the minor cost of integrating current extractors with the framework, researchers can benefit from the ease with which their image data and the extracted feature sets can be analyzed and trends within the dataset can be identified.

Acknowledgments

I would like to thank Deborah Estrin and Josh Hyman for providing me with the opportunity to pursue this research and guiding me through the entire process. I would also like to thank Amit Sahai for helping me develop these research ideas along the way.