Image + EXIF Viewer

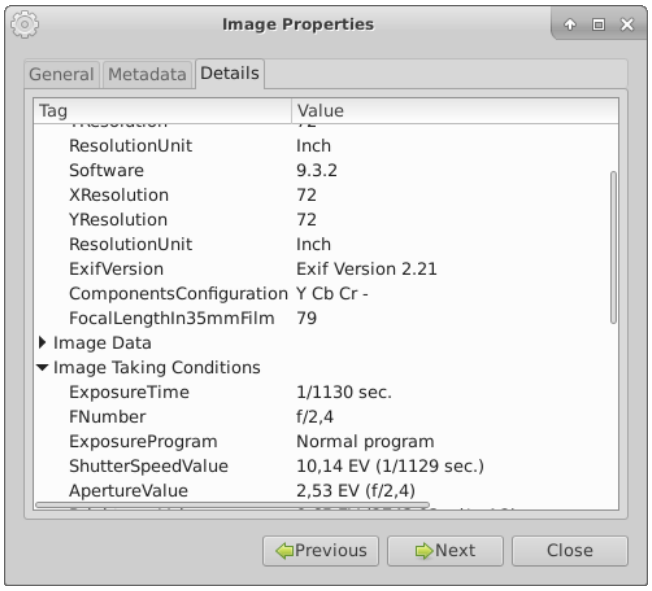
HCI 2017-2018 Programming Assignment

Prof. Andrew D. Bagdanov

November 20, 2017

1 Overview

Images are everywhere. With the proliferation of images and image formats (**primarily JPEG**) came the realization that there was a need for a **common standard** for exchanging **metadata** about digital images taken with digital cameras. That is precisely what the **Exchangeable Image File Format (EXIF)**. EXIF metadata is stored in JPEG images (in an **Application Segment**) and takes the form of (key, value) pairs. For example (this is the EXIF metadata viewer for the default image viewer in Ubuntu Gnome):



In this assignment you will implement an Image and EXIF Metadata Viewer as a polished **Graphical User Interface**. Your application must support a minimum of (scaled) image viewing and **EXIF tag visualization**.

2 Assignment

For this assignment you must implement a simple **JPEG Image and EXIF Viewer**. You may implement this in **any programming language** and using **any GUI framework** you desire. Almost anything goes, however the features your GUI must implement to receive full credit your GUI must implement the following features:

- **Visualization of images**: your GUI must support visualization of JPEG images (one at a time). Images should be **scaled** to have a maximum dimension (height or width) of 512 pixels.

- **Visualization of image EXIF data**: your GUI must list **all** EXIF tags encoded in the JPEG file. If the number of tags is too great to view in the application window, your GUI must provide a **scrolling widget** to view them all.
- **Rescaling**: the main window of your GUI must support **rescaling**. That is, when the user **resizes** the application, the user interface must **scale**, and most importantly the **image** should scale appropriately.
- **Image rotation**: your GUI must support **rotation** of the image in 90° increments. The GUI should support both a **button** (or menu) interface to access the rotation, and a **hotkey** interface.

Your GUI can also support the following features (which will be considered as **extra credit** in the final evaluation:

- **Geolocalization**: if an image has GPS Geolocation Tags in its EXIF tag set, implement a feature that allows users to **click** on the location and open a browser with Google Maps centered at the GPS location of the image. This link has useful information about using Google Maps URLs.
- **View multiple images**: allow the user to specify **more than one** image (from the **command line**, or from a **file selector**). Implement controls for switching/cycling to next/previous image in the list.

3 Evaluation

You **must** submit the source code for your complete implementation by the **deadline for registering for the exam**. Late submissions will **NOT** be accepted.

A note on **cheating**. I realize that there are **many** implementations of **Image and EXIF Viewers** out there. That is great, and you should look at them and learn from them. However, the code you submit **MUST BE YOUR OWN WORK**. Learn from others, but **write and submit your own implementation**.

Your implementation will be evaluated based on how you apply the programming models and constructs learned during the course. More specifically:

- **Functionality** (25%): your code must **work**. In your submission, include a minimal README that explains how to run your program (including any dependencies).
- **Code hygiene** (25%): keep things **clear**. This means writing concise, clear, and **reasonably documented** code. You should carefully identify **model**, **view**, and (maybe) **controller** components of your implementation.
- **Completeness** (50%): did you implement all of the **required** features? See the list above, but your GUI must implement all of the required features to earn full credit.
- **Extras** (max 10% bonus points): above and beyond. If you implement **extra features**, make sure you highlight them in the documentation for your submitted project.

4 Hints

Here I will collect some useful hints and advice for anyone choosing this programming assignment (this list might be updated, check back):

- The focus of this assignment is on **complete** and **correct** implementation using **best practices** (i.e. MVC). Your interface does not have to be **pretty**. Make it all **work**, make it work **correctly**, and **then** make it pretty if you have time.

- Look at hoe **other image viewers** visualize EXIF metadata. Learn from them and emulate their solutions.
- When it comes to reading and interpreting **EXIF** data, do **NOT** be tempted to roll your own – use an **existing library** for reading EXIF. Here are some good choices:
  - In Java: `https://github.com/drewnoakes/metadata-extractor/`
  - In C#: `https://github.com/drewnoakes/metadata-extractor-dotnet`
  - In Python: `http://pillow.readthedocs.io/en/3.4.x/index.html`
  - In C++ (yuck): `https://github.com/mayanklahiri/easyexif`

5 Useful Links

- A good basic overview of EXIF Tags.
- And another one.
- A list of common EXIF tags.