



Tutorial on Using Android for Image Processing Projects

EE368 Digital Image Processing, Spring 2011



Windows Version

Introduction

In this tutorial, we will learn how to set up the Android software development environment and how to implement image processing operations on an Android mobile device (e.g., the DROID phone that you have received). Android is an open-source platform developed by Google and the Open Handset Alliance on which interesting and powerful new applications can be quickly developed and distributed to many mobile devices. There is a growing community of Android developers and a growing market for Android devices, which now include tablets and TV setup boxes in addition to smartphones. Android also comes with a vast library of useful functions, including functions for user interfaces, image/bitmap manipulation, and camera control that we will frequently use in EE368. We look forward to seeing your novel image processing algorithms and applications running on Android devices as the quarter progresses.

The tutorial is split into two parts. In the first part, we will explain how to download and install the Android software tools onto your computer. Then, in the second part, we will explain how to develop image processing programs that can run on an Android mobile device.

Estimated time to complete this tutorial: 2 hours

Part I: Creating the Software Development Environment ¹

We will use the Google Android SDK, the Java JDK, and the Eclipse IDE to design, implement, and debug Android-compatible programs in this class.

Downloading and Installing Java JDK

1. Download Java JDK 6.0 from this website:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Run the downloaded installer file.
3. Write down the path to where the JDK is installed to, for example:
`C:\Program Files\Java\jdk1.6.0_24`

Downloading and Installing Eclipse

1. Download “Eclipse IDE for Java Developers (Helios)” from this website:
<http://www.eclipse.org/downloads/>

¹ Parts of this tutorial borrow explanations from the official Android developers’ website (developer.android.com).

2. Unzip the downloaded file to a convenient location on your hard disk, for example:
C:\\eclipse

3. Open a command-line interface (e.g., Start > “cmd”) and change to the directory where you unzipped Eclipse. This directory should contain “eclipse.exe” or a comparable executable. Link Eclipse to the installed Java JDK using this command like the following:

```
eclipse.exe -vm "C:\\Program Files\\Java\\jdk1.6.0_24\\jre\\bin"
```

Note: Replace “C:\\Program Files\\Java\\jdk1.6.0_24” with the location where you installed the Java JDK, and verify there is a “jre\\bin” sub-directory present.

4. When asked to choose a default workspace, pick a folder that is easy to remember and access, for example:
C:\\eclipse\\workspace

5. Verify that Eclipse starts properly and an IDE window like in Figure 1 is shown.

6. Each time Eclipse is started in the future, it should use the full command in Step 3. If using Windows, you can create a Shortcut to “eclipse.exe” and then edit the “Target” field of Shortcut Properties to contain the full command in Step 3.

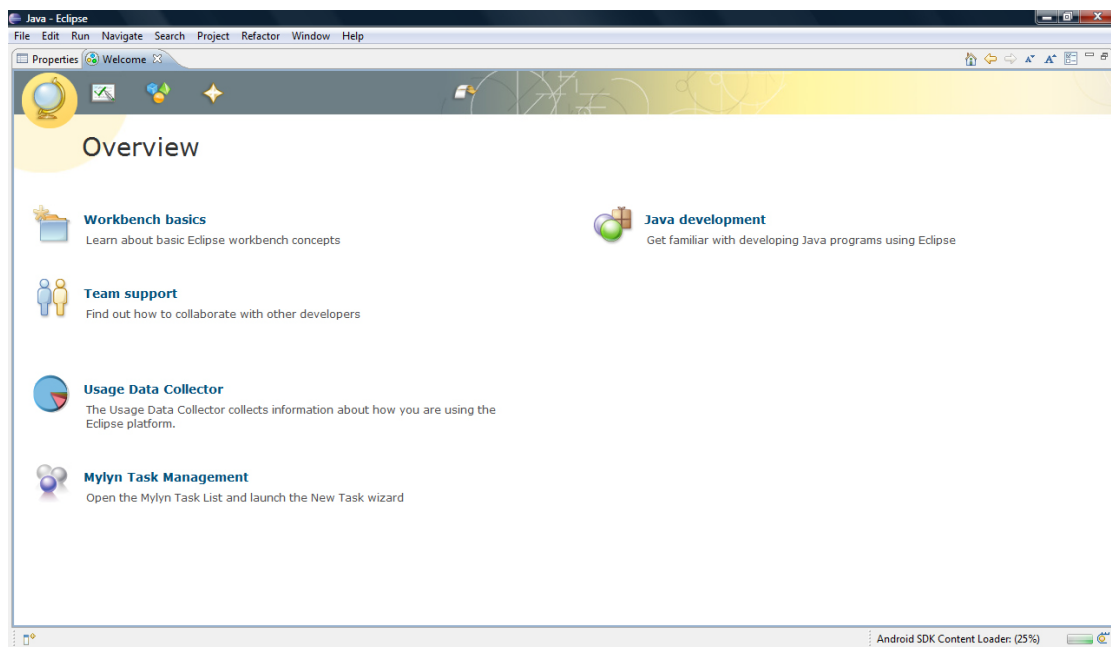


Figure 1. Start-up screen of the Eclipse IDE.

Downloading and Installing Android SDK

1. Download the Windows zip file for the Google Android SDK from this website:
<http://developer.android.com/sdk/index.html>
2. Unzip the downloaded file to a convenient location on your hard disk, for example:
`C:\android-sdk-windows`
3. Add the location of the “tools” and “platform-tools” subfolders for the Android SDK to your system PATH. For help on editing the PATH, please follow the tips here:
<http://www.windowsitpro.com/article/john-savills-windows-faqs/how-can-i-add-a-new-folder-to-my-system-path-.aspx>
4. Install the ADT plugin for Eclipse.
 - a. Open Eclipse.
 - b. From top menubar, choose Help > Install New Software.
 - c. In the Available Software dialog, click Add ...
 - d. In the Add Site dialog, enter “Android Plugin” in the “Name” field and enter the following URL in the “Address” field and click OK:
<http://dl-ssl.google.com/android/eclipse/>
 - e. In the Available Software dialog, you should see “Developer Tools” listed. Select the checkbox next to “Developer Tools” and click Next.
 - f. In the Install Details dialog, you should see “Android DDMS”, “Android Development Tools”, and possibly other tools listed. Click Next, accept all license agreements, and click Finish.
 - g. Restart Eclipse (be sure to link to the correct Java SDK using the “-vm” option mentioned above).
5. Link Eclipse to the installed Android SDK.
 - a. In Eclipse, select Window > Preferences. A window like that in Figure 2 should pop up.
 - b. Select Android from the left panel.
 - c. For the “SDK Location”, click Browse and find where you installed the Android SDK.
 - d. Click Apply and then click OK.
 - e. Restart Eclipse (be sure to link to the correct Java SDK using the “-vm” option mentioned above).

6. Download updates for the Android SDK.
 - a. In Eclipse, select Window > Android SDK and AVD Manager. A window like that in Figure 3 should pop up.
 - b. Select Available Packages from the left panel.
 - c. Click the checkboxes for all available packages and click Install Selected.
 - d. After installation, select Installed Packages from the left panel and verify that the packages are correctly installed.
7. If you encountered problems in this section, please take a look at the tips on these sites:
 - <http://developer.android.com/sdk/index.html>
 - <http://developer.android.com/sdk/installing.html>
 - <http://developer.android.com/sdk/eclipse-adt.html>

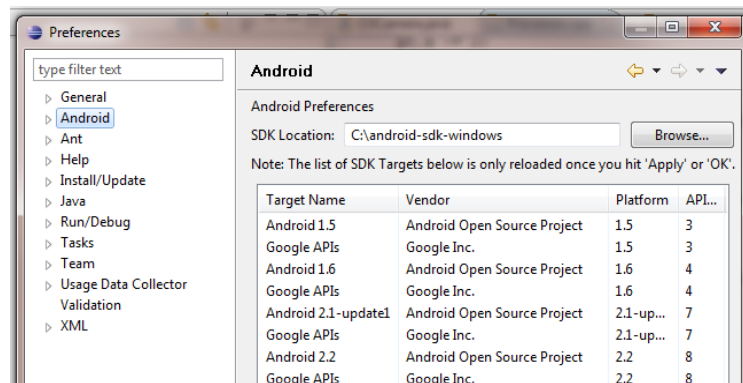


Figure 2. Android preferences panel in Eclipse.

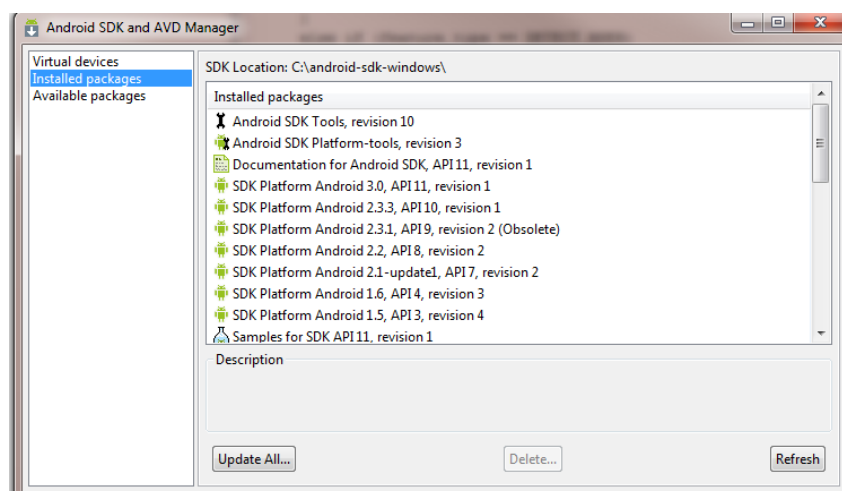


Figure 3. Android SDK manager panel in Eclipse.

Linking Your Phone to Your Computer

1. Turn on your phone.
2. Go to the home screen.
3. Select Menu > Settings > Applications > Development and then enable USB debugging.
4. After you have downloaded updates for the Android SDK in Eclipse above, the USB driver should have been included. Install the USB driver on your computer, following the tips on these pages:
<http://developer.android.com/guide/developing/device.html#setting-up>
<http://developer.android.com/sdk/win-usb.html>
http://www.motorola.com/Support/US-EN/Support-Homepage/Software_and_Drivers/USB-and-PC-Charging-Drivers

Note: You may need to restart your computer after installing the USB driver in order for the phone to show up in Eclipse.

Part II: Developing Image Processing Programs for Android

Now that the Google Android SDK, the Java Runtime, and the Eclipse IDE are all set up on your computer, we are ready to start writing image processing programs that can run on an Android-compatible mobile device.

Hello World Example

First, we will build a simple Android program in Eclipse. This simple example will also help you to become familiar with how to create an Android project, how to (auto) compile source code, and how to run the generated executable on the mobile device. Please follow the instructions on this page to develop the “Hello World” program:

<http://developer.android.com/resources/tutorials/hello-world.html>

Note: In case there are strange errors when the project is created about “R.java”, simply add a comment line (e.g., “// dummy comment”) at the top of “gen : com.example.helloandroid : R.java” and save the file. This will make the errors disappear.

In the external “Hello World” tutorial, they only run the “Hello World” program in an emulator. Additionally, we will now also run the program on the actual Android phone. Make sure your phone is properly linked to your computer.

1. In Eclipse, select Run > Run Configurations > Android Application > HelloWorld > Target. Choose Manual for Deployment Target Selection Mode.

2. Select Run, and in the Device Chooser dialog, select your Android phone. The “Hello World” program will be sent to and automatically started on your phone, and you should see the screen similar to Figure 4 on your phone.

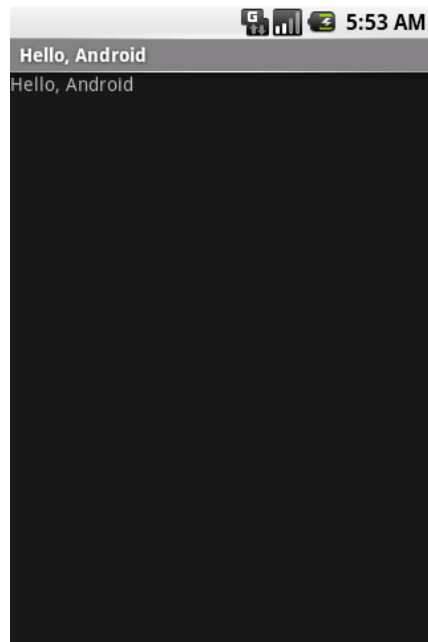


Figure 4. “Hello World” program running on an Android phone.

EE368 Viewfinder Example

Now, having grasped the fundamentals of building and running an Android application, we will create a more complicated project involving the onboard camera and real-time image processing.

1. Create a new Android project with the following parameters.
Project name: ViewfinderEE368
Check the box for Android 2.0.1
Application name: Viewfinder EE368
Package name: com.example.viewfinderee368
Check the box for Create Activity and enter: ViewfinderEE368
Min SDK Version: 6
2. Copy the text in the following document into `AndroidManifest.xml`. This defines the main activities and permissions for this program.
<http://ee368.stanford.edu/Android/ViewfinderEE368/AndroidManifest.xml>
3. Copy the text in the following document into `src : com.example.viewfinderee368 : ViewfinderEE368.java`. This defines the classes in this program.
<http://ee368.stanford.edu/Android/ViewfinderEE368/ViewfinderEE368.java>

4. Check to make sure everything is copied correctly into the project. If there are compilation errors, a red X will appear in the Package Explorer.
5. Select Run and in the Device Chooser dialog, select your phone. You should see something like Figure 5 on your phone. Point the camera at different objects around you to see how the mean, standard deviation, and histogram of each color channel changes dynamically. You are augmenting the viewfinder in real time!



Figure 5. “Viewfinder EE368” program running on an Android phone.
Demo video: <http://ee368.stanford.edu/Android>

Real-time Phone Debugging in Eclipse

It is actually possible to view real-time messages from the phone in Eclipse, which can be very helpful for debugging and code development.

1. Select Window > Open Perspective > DDMS.
2. A new tab entitled “DDMS” should appear next to the default “Java” tab. Click on the “DDMS” tab.
3. Select Window> Show View > LogCat. The LogCat view shows a sequential list of real-time messages from the phone. In particular, error messages in red can be very useful when trying to debug a problem.

Taking a Screenshot of the Phone

At some point, it may be useful to take a screenshot of the phone, e.g., to use as a figure in your project report.

1. Make sure Eclipse is closed.
2. Go to “android-sdk-windows/tools” and execute the “ddms.bat” file.
3. Select your Android device, as shown in Figure 6. The “Log” panel should then display real-time updates from the device.
4. Select Device > Screen Capture, and a panel like Figure 7 should pop up.

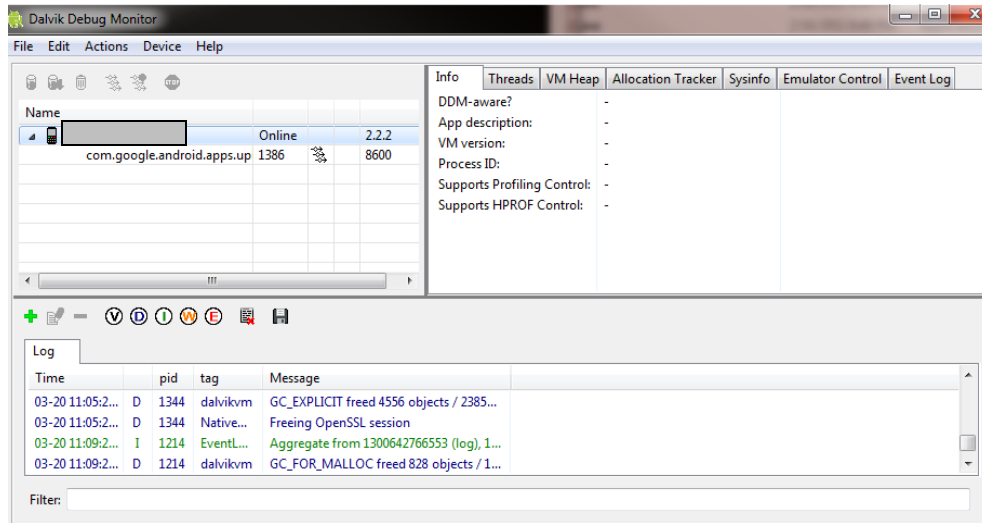


Figure 6. Dalvik Debug Monitor panel.



Figure 7. Device screen capture panel.