ARL-TR-8798 ● SEP 2019

Mobile App Development for the Three-Dimensional Wind Field (3DWF) Model on Android and Windows Platforms

by Giap Huynh, Kristyna Rehovicova, and Yansen Wang

NOTICES

Disclaimers

ARL-TR-8798 ● SEP 2019

Mobile App Development for the Three-Dimensional Wind Field (3DWF) Model on Android and Windows Platforms

Giap Huynh and Yansen Wang
Computational and Information Sciences Directorate, CCDC Army Research Laboratory

Kristyna Rehovicova
Institute for Human & Machine Cognition

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
September 2019	Technical Report	1 October 2018–1 October 2019

4. TITLE AND SUBTITLE	**5a. CONTRACT NUMBER**
Mobile App Development for the Three-Dimensional Wind Field (3DWF) Model on Android and Windows Platforms	
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER**
Giap Huynh, Kristyna Rehovicova, and Yansen Wang	
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
CCDC Army Research Laboratory ATTN: FCDD-RLC-EM Adelphi, MD 20783-1138	ARL-TR-8798
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

ORCID ID(s): Giap Huynh, https://orcid.org/0000-0001-6962-6034

14. ABSTRACT

The Three-Dimensional Wind Field (3DWF) model developed by the US Army Combat Capabilities Development Command Army Research Laboratory (ARL) is based on the mass conservation principle and produces a high-resolution wind field over complex terrain. Although its practical graphical user interface (GUI) has already been developed to simplify the use of 3DWF, it is only available on a Windows desktop computer. To expand the 3DWF model's usefulness on different operating systems and utilize the convenience of handheld mobile devices such as Samsung's Galaxy tablet and Microsoft's Surface tablet, two different Android and Windows apps have been created and deployed for those two tablets, respectively. This technical report documents collaborative efforts between ARL and the Institute of Human & Machine Cognition to design a 3DWF app for the Galaxy tablet. In addition, processes to migrate the previously designed 3DWF GUI from a Windows desktop to a Windows tablet to implement a 3DWF app on the Surface tablet are also described.

15. SUBJECT TERMS

Three-Dimensional Wind Field, 3DWF, Android, Windows tablet, app

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Giap Huynh
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified	UU	40	(301) 394-1156

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgments

We would like to thank the following individuals for their contributions to this effort:

- Dr Benjamin MacCall for his suggestions on how to improve the Three-Dimensional Wind Field (3DWF) mobile apps.
- Dr Brian Reen for his rigorous review and helpful inputs to the writing of this technical report.

1. Introduction

The Three-Dimensional Wind Field (3DWF) model was developed at the US Army Combat Capabilities Development Command (CCDC) Army Research Laboratory (ARL) and has been frequently improved over the past few years. The model is based on the mass conservation principle, wherein the divergence in a flow field is eliminated. A detailed description of the model has been provided in several papers (e.g., Wang et al. [2005, 2013]). In an effort to help the user operate the model properly on a Windows desktop computer, a user-friendly and Google GIS-based GUI was designed as a Hypertext Markup Language (HTML) webpage to utilize the online Google Maps plugin for displaying the 3DWF model's 2-D wind field result.

As handheld mobile devices such as tablets and smartphones become more popular, powerful, and convenient, designing and deploying mobile apps for the 3DWF model on such devices would expand the model's usefulness and its availability on different computer platforms. Since a smartphone's screen is relatively small for viewing a graphical representation of the wind field and its memory may be too limited to storing huge amounts of terrain data, it is more practical to view the wind field display on a tablet due to its larger screen size and greater capability for handling large terrain data files.

This technical report focuses on the creation of a 3DWF app for Samsung's Galaxy tablet and the migration of the existing 3DWF GUI for a Windows desktop to Microsoft's Surface tablet, as well as how to operate them. The mobile GUIs developed for the 3DWF model should also be very useful for viewing visualizations of output from CCDC ARL's next-generation microscale model, the Atmospheric Boundary Layer Environment – Lattice Boltzmann Model (ABLE-LBM) (Wang et al. 2018), since both the ABLE-LBM and 3DWF models intend to simulate the atmospheric boundary layer flows in the same microscale range.

2. Hardware

Since the 3DWF model involves a great deal of computations as well as reading from very large terrain data files and writing great amounts of model output to a local file, the chosen mobile devices should have enough memory and storage for the model to run without issues when a large 3-D terrain domain, such as one with 201 × 201 × 201 grid points, is applied. In addition, the mobile devices should also have fast processors in order to produce the model results in a very short time. For Android and Windows operating systems (OSs), the two most popular tablets on the market during this project were the Samsung Galaxy S3 tablet and the Microsoft Surface Pro tablet, respectively (Fig. 1), and as such they were selected for this effort.

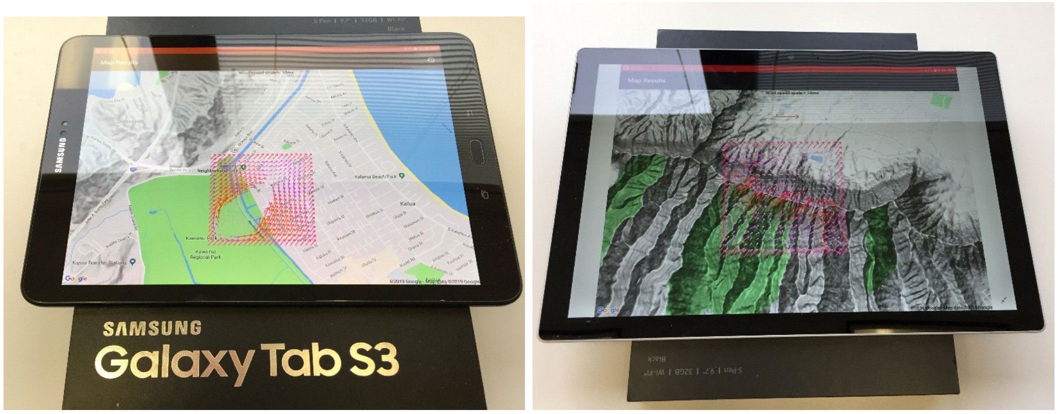


Fig. 1 Samsung Galaxy S3 tablet (left) and Microsoft Surface Pro tablet (right)

2.1 Android Platform Mobile Device

The Samsung Galaxy S3 tablet (Fig. 1, left) was chosen as it was the most popular Android platform tablet on the market during this project. It has a 4-GB memory (RAM) + 32-GB internal storage, up to 400-GB external storage, and is one of the best mobile central processing units (CPUs) on the market. The Galaxy is also equipped with standard Google Maps and Google Earth access, which is very helpful for displaying terrain. However, based on experiences during many test runs on this device, memory has been and still is a big issue when a large 3-D terrain domain (such as 201 × 201 × 201 grids) is applied. Therefore, it still has some limitations, such as slower processing time and domain size limitations, as compared to the Windows Surface Pro tablet.

2.2 Windows Platform Mobile Device

For a Windows platform mobile device, the Microsoft Surface Pro Model 1796 tablet (Fig. 1, right) was chosen due to its popularity and power. This tablet is equipped with an Intel Core i7 Processor and has a total 16 GB of RAM including 18.3 GB of virtual memory and 16 GB of installed physical storage. During the testing of the deployed 3DWF app on this tablet, it proved that it can handle the terrain of a much larger domain without memory issues. Although the Surface tablet is a little more bulky than the Galaxy tablet, it offers a larger screen display for the generated wind field and can act as a laptop computer with a connecting keyboard. Besides bulk, a disadvantage for this device is that the user must subscribe to and purchase Google Maps and Google Earth licenses in order to display the terrain map underneath. Otherwise, although the user can still view the background terrain map, it is purposely darkened, as shown in the wind field figures (Figs. 23 and 26 in Section 4).

3. Creating the Android App for the 3DWF Model

3.1 Android App Design Software: Android Studio IDE

To create an Android app, an Android app design software tool such as Android Studio Integrated Development Environment (IDE) must be utilized to design and to prepare code for the app. Android Studio IDE can be downloaded freely from https://developer.android.com/studio/index.html.

This IDE software does not have to be installed in an Android OS computer. It can be installed and operated in any Windows computer, such as the Windows Surface Pro tablet used for this project. Although the 3DWF's Java code programs can be composed from scratch in Android Studio, they can also be copied and pasted from the original 3DWF's Java programs designed for the Windows desktop. Android Studio IDE also allows unique and more convenient ways to read and store interactive inputs internally without having to store them in a file; therefore, some modifications were necessary to the original Java programs in order to improve the 3DWF app's performance on the Galaxy tablet.

Once installed and opened, Android Studio IDE prompts a general menu, as shown in Fig. 2.

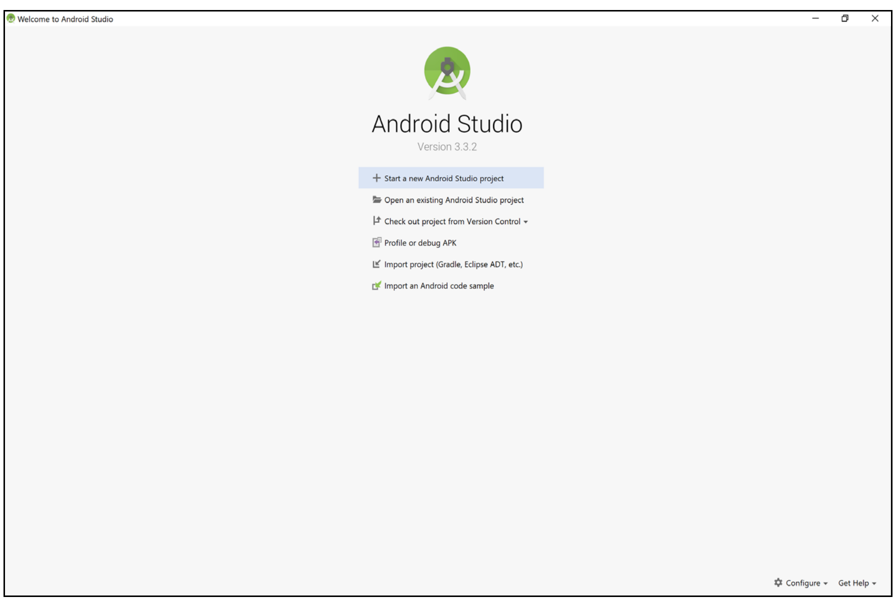


Fig. 2 Android Studio IDE general menu page

There are helpful built-in instructions as well as online tutorials to assist with operating Android Studio IDE and creating an Android app. This technical report only focuses on describing the basic structure of the 3DWF app and how to operate it.

3.2 3DWF App Architecture

For the Android OS, the JavaScript programs and HTML files designed for the 3DWF GUI from the Windows desktop cannot be migrated directly to an Android tablet since they contain some of Microsoft's ActiveX control commands to allow access to local files. Therefore, the 3DWF app for the Galaxy tablet had to be redesigned to conform to the Android OS while performing similar tasks to its Windows version. Figure 3 displays a flowchart of sequential pages to be constructed for the 3DWF app. Instead of operating from a single menu page as in the case of the Windows GUI version, the 3DWF app for Android tablet consists of sequential pages, and in each, it performs a particular task. First, the app quickly splashes the title page showing the CCDC ARL logo and then loads all the necessary resources. It then starts the MainActivity.java class (under the "app" subdirectory of the project) to extend Activity instances and grants the needed permissions before it starts.

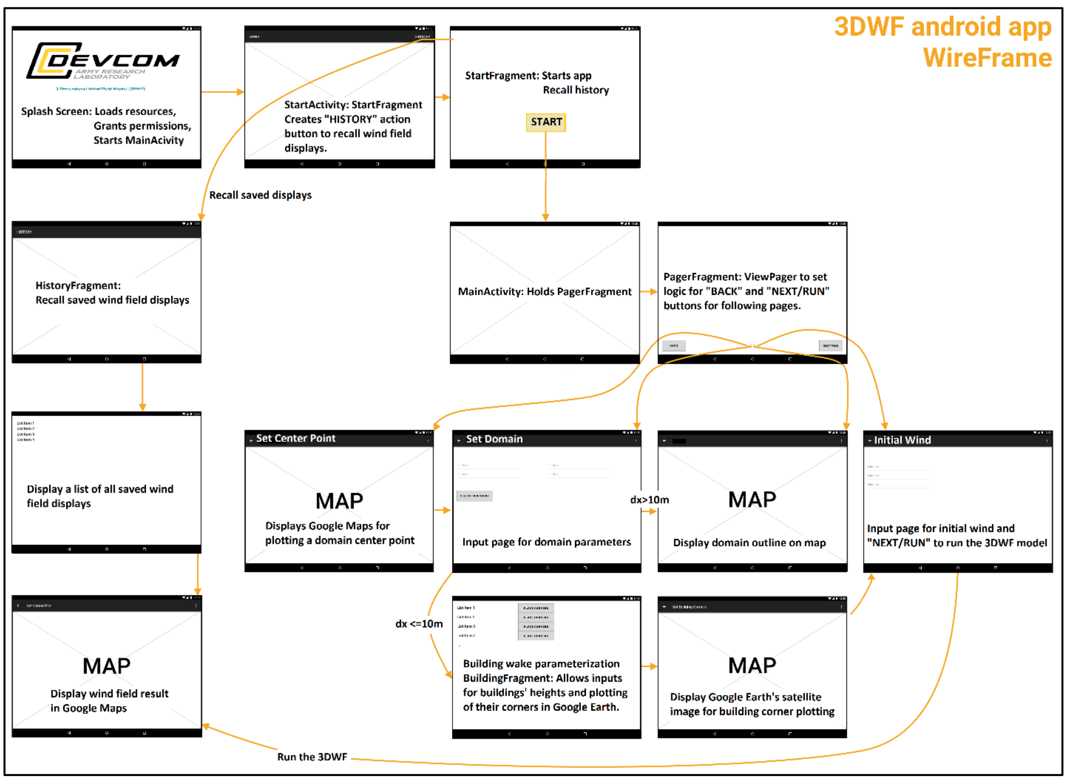


Fig. 3 Wireframe flowchart for the 3DWF app

Next, there are two tasks to be performed on the “START” page. The first task is to be able to save all generated wind field displays and then redisplay each when needed. The second task is to start the 3DWF app. For the first task, the app uses StartActivity, which contains StartFragment, to produce an action bar button named “HISTORY”, located at the top-right corner of the “START” page. This HISTORY button is controlled by the HistoryFragment.java class to help with recalling all generated wind field displays. For the second task, a START button is created that, when tapped by the user, goes on to the next page. Before going on to the next page, “Set Center Plot”, the app must create “BACK” and “NEXT/RUN” buttons to be used on the following pages. The logic of these two buttons is written inside the ViewPagerAdapter.java and PagerFragment.java files of the container MainActivity in order to allow the user to flip backward (one fragment to the left) to the previous page or forward (one fragment to the right) to the next page.

On the next page, “Set Center Plot”, the app displays a default Google Maps area for the user to plot a center point of a domain to be defined on the next page. The center-point coordinates are read in by the CenterPlotFragment.java file. Next, the app proceeds to the “Set Domain” page, which allows the user to enter parameters to define a 3-D domain for the 3DWF model. The domain parameters are read in by the DomainFragment.java file. It then runs the terrain processing program to locate the involved Shuttle Radar Topography Mission (SRTM) data files to create a domain data file for the 3DWF model and draw an outline of the domain (using OutlineFragment.java). If the app detects a grid spacing of $dx \leq 10$ m, then it will need to go through the rasterizing or digitizing building process to create the buildings’ grid information for a high-resolution case.

Finally, the app goes to the “Initial Wind” data input page (far-right block). The initial wind data entered are handled by the InitialWindFragment.java file. The “RUN” button on this page executes the 3DWF model program through the PagerFragment.java file, as mentioned earlier. Once finished, the generated wind field is displayed in Google Maps (bottom-left block). Constants and static methods for the 3DWF app are defined in the UTILS subdirectory of the project.

If the 3DWF app is successfully built (or imported) with the “Build project” option in the main window of Android Studio IDE, then the two main items in bold font, **3dwf** and **app**, are generated, as shown on the left of Fig. 4. All generated codes for this 3DWF app project are saved under the local subdirectory, C:\Users\arl\a3dwf\, of the Surface tablet. Double clicking on each file displays its content in the window to the right.

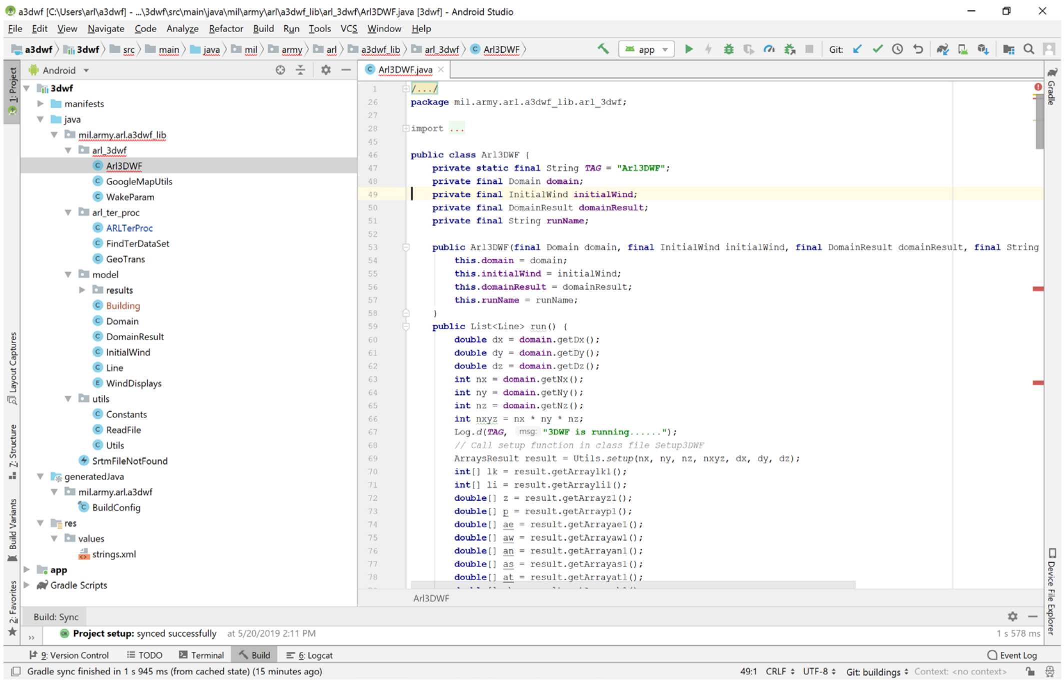


Fig. 4 General tree structure of a successfully built 3DWF app in Android Studio IDE

The branch "**3dwf**" contains Java programs of the 3DWF model and additional Android Studio-generated codes, as shown in Fig. 4. The sub-branches "**arl_3dwf**" and "**arl_ter_proc**" of the branch "**java**" contain codes for the 3DWF model and terrain data processing, respectively. In the Windows version of the app, the 3DWF model's Java programs were written separately from the JavaScript GUI, and therefore, interactive inputs from the GUI must be saved in a file before they can be read by the model's Java programs. However, Android Studio IDE allows the 3DWF app to read interactive inputs and execute the model's programs without reading saved files. As a result, some minor modifications had to be done to the original Java programs. The output data file created by the model is saved under the subdirectory 3DWF. The sub-branch "**model**" controls the behaviors on each page and the sub-branch "**utils**" controls the constants, reading the input file, and computations used to create data arrays for the 3DWF model. As a result of these required arrangements, some sections in the original Java programs of the 3DWF have been moved to the "**utils**" sub-branch.

Similarly, the branch "**app**" in Fig. 5 contains files for creating and deploying the 3DWF app to an Android device such as the Galaxy tablet. To deploy the 3DWF app to the Galaxy tablet, a USB cable must be connected between the Surface and the Galaxy tablets.

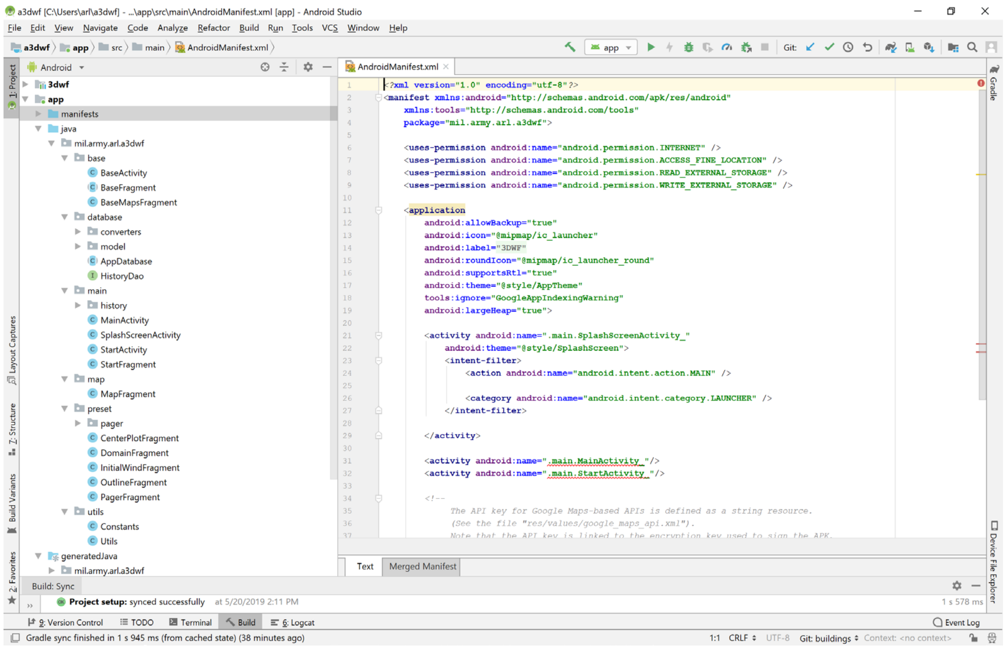


Fig. 5 Tree structure of files under the “app” branch

After the “**a3dwf**” project is created or modified, select “Build” from the option menu and double click on “Make Project” or “Rebuild Project” to build or rebuild the project. If successful, the message under the “Build” option at the bottom of the menu will indicate so. Next, select the “Run” option and double click on “Run app” to deploy the 3DWF app to the Galaxy tablet. If the USB cable is connected, the Galaxy tablet name shows up at the top of the connecting list. Once deployed, the 3DWF app must be added to the app list to be permanently installed on the Galaxy tablet. The user can add a deployed app to the app list by tapping on the “Google” app group and then tapping on “ADD APPS” at the bottom of its window. Next, the user simply places a check mark on the app’s icon and taps “ADD” at the top right to officially add it to the “Google” app group.

3.3 Customizing the 3DWF App’s Icon

After deployment, the 3DWF app’s Android default icon is added to the Galaxy tablet’s app list. The default icon is assigned and can be changed by selecting another one from the Android Studio’s white mouse face icon list (Fig. 6). However, it would appear more professional to design one’s own distinctive icon for the 3DWF app. The customized icon can be designed by following these two simple steps:

- First, use a Paint or similar app to create one’s own icon and then copy the icon file, named “ThreeDWF.jpg” for this example, to this specific subdirectory:

C:\Program Files\Android\Android Studio1\plugins\android\lib\templates \gradle-projects\NewAndroidModule\root\res\drawable-v24\

- Next, from the "**app**" branch of the project in Android Studio IDE, select New-Image Asset to open the Configure Image Asset window and then use Browse at the Source Asset's path to select the customized file "ThreeDWF.jpg". Click Next and Finish to complete. Figure 6 shows Android Studio app's default mouse face icons and the "Select Path" window to customize the 3DWF app's icon (shown with a wind field image).

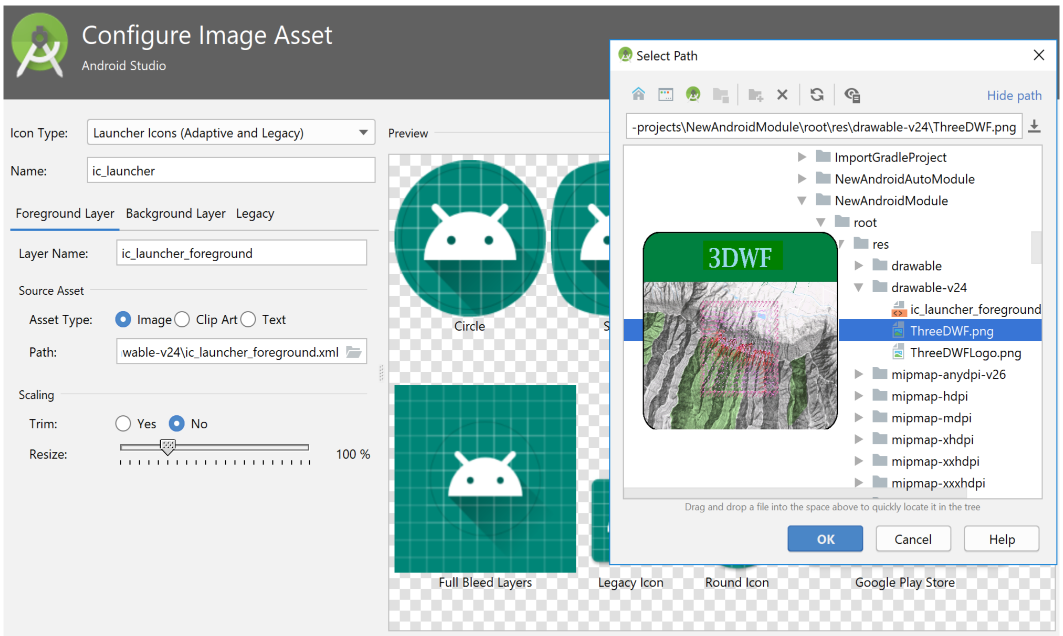


Fig. 6 Process to customize the 3DWF app's icon

Once an app is successfully added to the Galaxy tablet's app list, any future modifications to the app from Android Studio IDE or new redeployments to the Galaxy will automatically update the app. There is no need to perform "Add apps" in Android Studio IDE again unless it is removed from the Galaxy's app list.

3.4 Procedure to Operate the 3DWF App on the Galaxy Tablet

The Android app for the 3DWF was designed to run the 3DWF model in a user-friendly manner. Once deployed and added to the list of the Galaxy tablet's available apps (Fig. 7), the 3DWF app is ready for use. The user can proceed with the following steps to run two example cases, one for low resolution (grid spacing dx > 10 m) and another for high resolution.

1) From the Galaxy tablet's screen, tap on the Google app group and then tap on the 3DWF app's icon, as shown in Fig. 7, to open the 3DWF app.

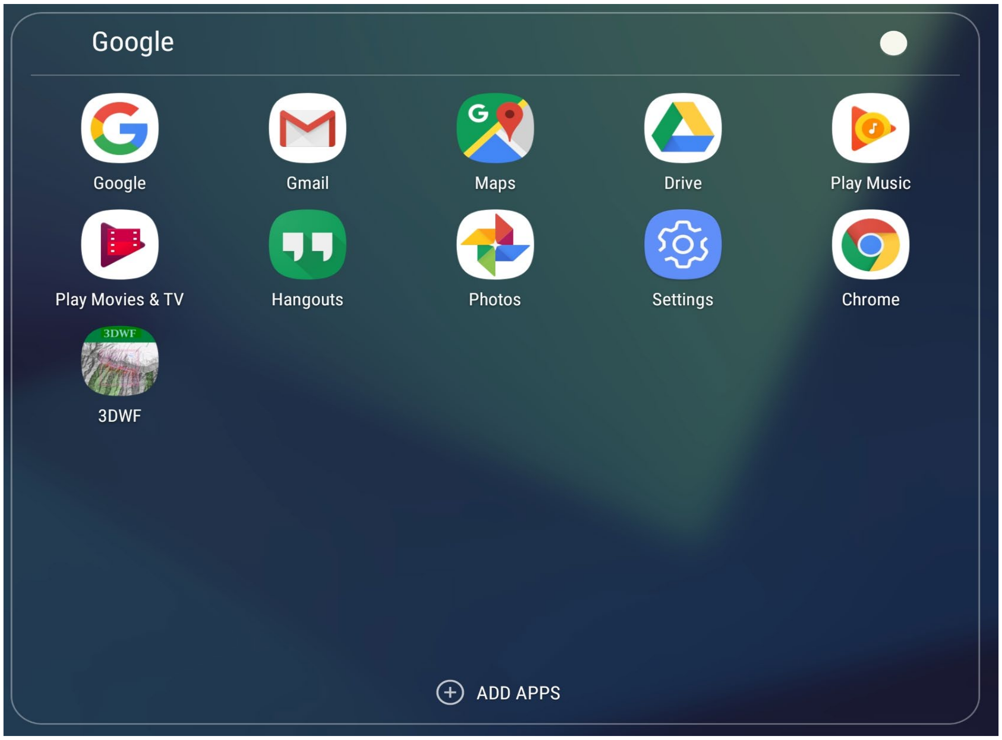


Fig. 7 The 3DWF app deployed on the Galaxy tablet's app list

A title page appears briefly showing the CCDC ARL logo and the 3DWF app's name (Fig. 8).

Fig. 8 Starting screen of the 3DWF app

Immediately after, a start page (Fig. 9) appears asking for a suffix to be added to the model's default output data filename "grads_file_3DWF.dat". This suffix allows the user to save all output files with distinctive names to avoid them being overwritten after each new run. For example, if the given suffix is "Nowake", then the output data file would be "grads_file_3DWF_Nowake.dat". The button "HISTORY" at the top right of the screen is for recalling all saved wind field result displays. Any of these wind field displays can be removed by quickly swiping a finger to the left on the wind field display's name in the list.

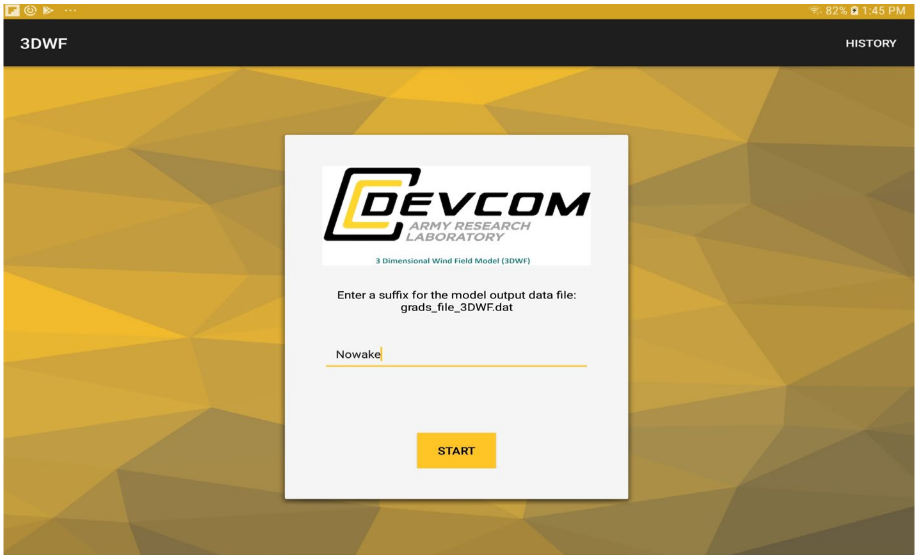


Fig. 9 Distinctive suffix "Nowake" to be added to the name of the output data file

Next, tap on the "Start" button to begin with the Set Center Plot page (Fig. 10). A help message appears with the suggestion: "Either use a finger to plot a domain center point on the map or enter its latitude and longitude directly into the 2 reserved text boxes at the top".

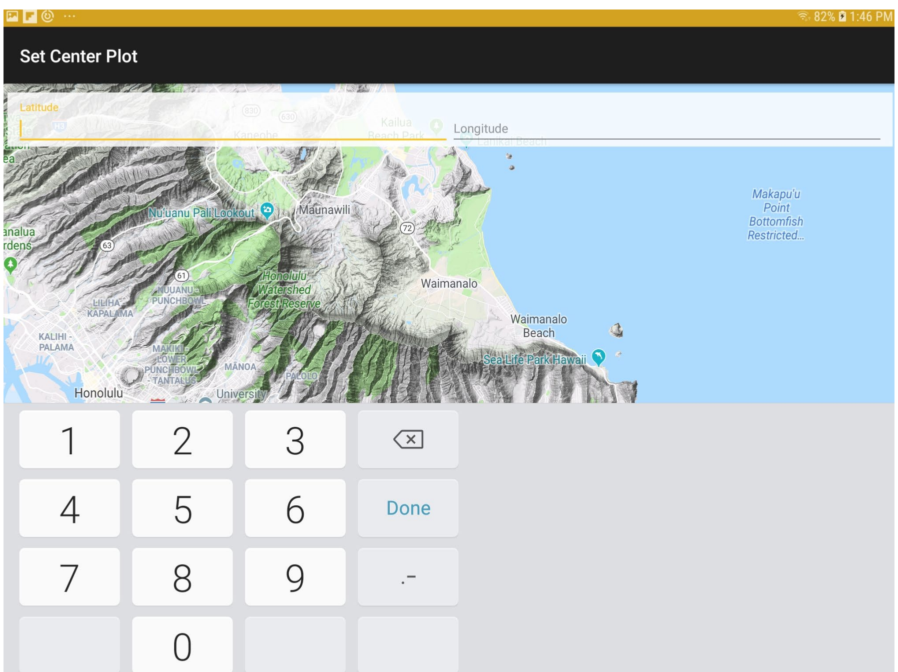


Fig. 10 Set domain center point manually by entering its latitude and longitude coordinates

2) On the Set Center Plot page, if the user prefers to enter the center point's coordinates manually in the latitude and longitude slots located at the top of the map, then they can use the pop-up number keyboard. If not using the number keyboard, tap the "Done" button to remove it from the map.

   On the other hand, if the user does not know nor care about the exact center point, then they can use a fingertip to plot the center point directly on the map. The default Google Maps is zoomed in at a hardcoded location in Hawaii. If the user prefers to run at a different location, then the user can use a finger to navigate around the globe to a preferred location. However, the selected location must lie inside the area defined by the downloaded 5° by 5° and approximately 30-m spatial resolution SRTM data files stored in the subdirectory "Internal storage/3DWF" of the tablet. For generating high-resolution ($dx \leq 10$ m) wind fields for which wake parameterization at high buildings or forest canopies must be performed, the center-point plot should be placed at an area that has some tall buildings or vegetation such as in an urban area or an area with large groups of tall trees. The user simply uses a fingertip to plot a center point at the location of interest. If the plotted point, marked by a red balloon as shown in Fig. 11, is not correct, the user can move it to another position.

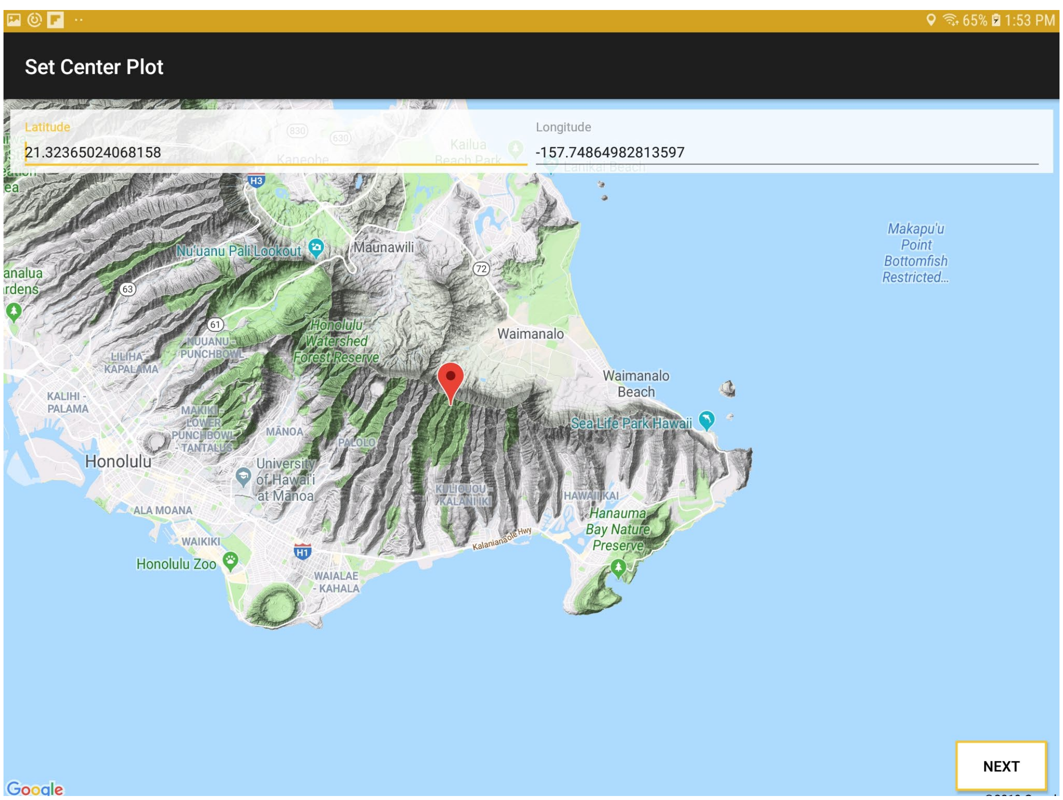


Fig. 11 Domain's center point marked by a red balloon

The marked center point is the center of the horizontal plane of the 3DWF model's 3-D domain with its parameters or sizes to be specified on the next page. The plotted latitude and longitude are also automatically displayed in the two slots at the top of the map. The 3DWF app captures the center point's coordinates and uses them to locate all adjacent SRTM data tile files involved in the chosen 3DWF domain.

3) Tap Next to go to the "Set Domain" page to enter the 3DWF domain's parameters (Fig. 12). Increments dx and dy are grid spacing numbers in the x- and y-directions of the horizontal plane of the 3-D 3DWF domain. dz is the level or layer spacing in the vertical direction. Integer parameters nx, ny, and nz are the number of grid points in the x-, y-, and z-coordinate directions, respectively. The hardcoded default values entered in this page are typical inputs for a domain in a low-resolution case indicated by dx = dy = 20 m. It is considered a high-resolution case when dx ≤ 10 m. All length inputs are entered in meters.

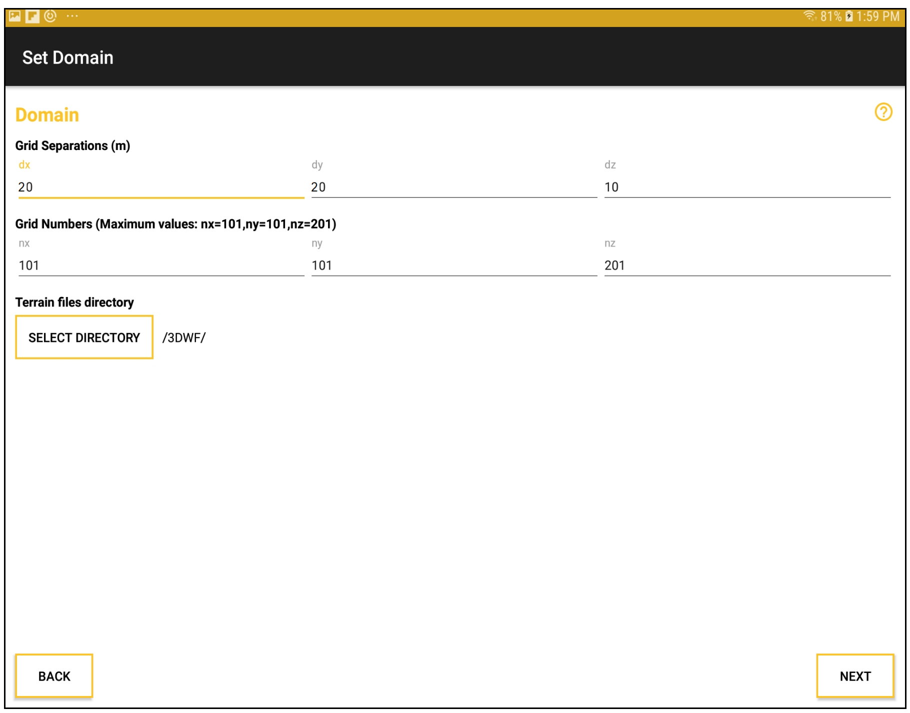


Fig. 12 Set domain input page for the low-resolution case

The user can replace these default values with a new set of values for dx, dy, dz, nx, ny, and nz. Although the 3DWF app is usually executed on a square in the horizontal plane with nx = ny, it also can be executed on a rectangle in the horizontal plane with nx different from ny. The last input is the default subdirectory containing the SRTM data tiles. Currently, these 5°

by 5° SRTM data tiles can be freely downloaded online from a website such as http://srtm.csi.cgiar.org/srtmdata/. The current SRTM data files are in ASCII format and contain latitude, longitude, and terrain elevation information.

Since the 3DWF model generates meaningful results only if the domain has sufficient height (i.e., extent in the z-direction) and this is the product of dz and nz, it requires the input values for dz and nz to satisfy the following condition with max_elev denoting the highest elevation of the terrain within the domain:

$$dz \times nz \geq max\_elev + 1000 \text{ m}$$

For example, if the 3DWF app detects the maximum terrain elevation for the chosen domain is max_elev = 525 m, then the product dz × nz must be equal or greater than (525 + 1000) or 1525 m. Therefore, combinations of entered values for dz and nz pairs such as dz = 20 m and nz = 101, or dz = 10 m and nz = 201, are valid since their products are 2020 and 2010 m, respectively. However, if the entered values are dz = 10 m and nz = 101, then the product of dz and nz is only 1010 m. Since the product dz × nz is less than the required height of 1525 m, these entered values are not considered valid for the 3DWF model. The 3DWF app was designed to automatically detect this problem and subsequently prompt a message showing the maximum elevation of the domain to remind the user to reenter a new set of data for dz and nz to satisfy that condition. There is also a Help button that appears as a question mark icon at the upper-right corner of the domain page to assist with other inputs.

Once done, tap the Next button at the lower right to execute all Java terrain process programs under the branch "**arl_ter_proc**" in Android Studio IDE. The terrain process programs will search for involved SRTM data tiles and create 3-D domain data for the 3DWF model. Due to limited memory and the speed of a tablet, this processing may take few minutes. The domain data derived from SRTM will include latitude, longitude, and terrain elevation. If successfully executed, the app displays the yellow domain outline in Google Earth, as shown in Fig. 13.

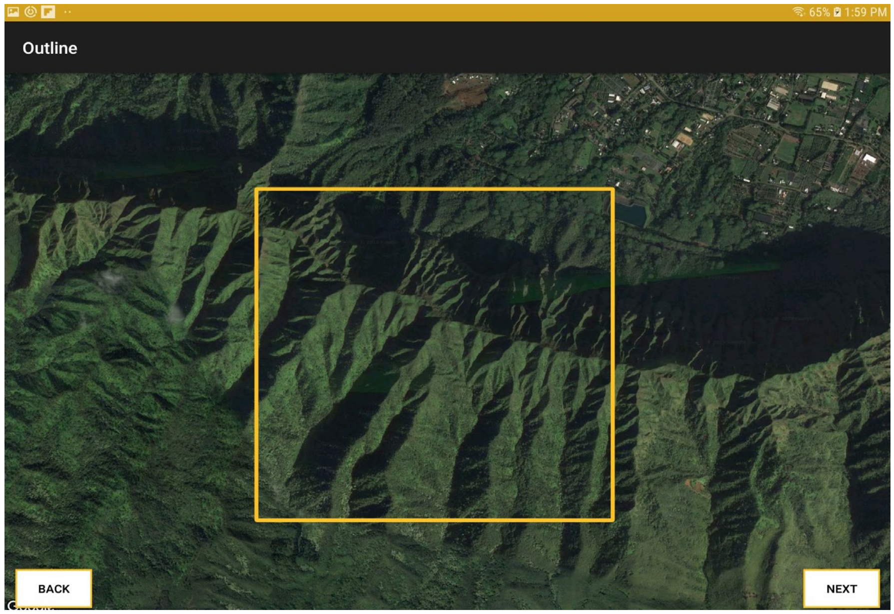


Fig. 13 Domain outline displayed in Google Earth

4) Tap Next to open the "Set Initial Wind" page for initial wind entries, as shown in Fig. 14. In this page, the user can enter the height above the ground (in meters) at which the wind is specified (e.g., the height where the anemometer sensor is placed), the initial wind speed in meters per second, and the initial wind direction in degrees. The wind direction follows the meteorological wind direction with respect to true north, which is 0° for wind flowing from north to south, 90° for wind flowing from east to west, 180° for wind flowing from south to north, and 270° for wind flowing from west to east. For example, enter 30 if the anemometer's height is at 30 m, 6 for a wind speed of 6 m/s, and 220 for a wind direction of 220°. The last item "Wind Display" allows the user to select to plot in arrow style or wind barbs style. It is important to decide which display option to select before running the 3DWF model. To display with the other option, the user must go back and start from the beginning of the app. Once finished adjusting the wind display type, tap the RUN button to execute the 3DWF model. Again, due to limitations of the Galaxy tablet, it may take another few minutes to run the 3DWF model, depending on the given size of the 3-D domain. If successfully executed, the 3DWF app produces an output data file named with a suffix as mentioned in the beginning and stored in the subdirectory /3DWF.

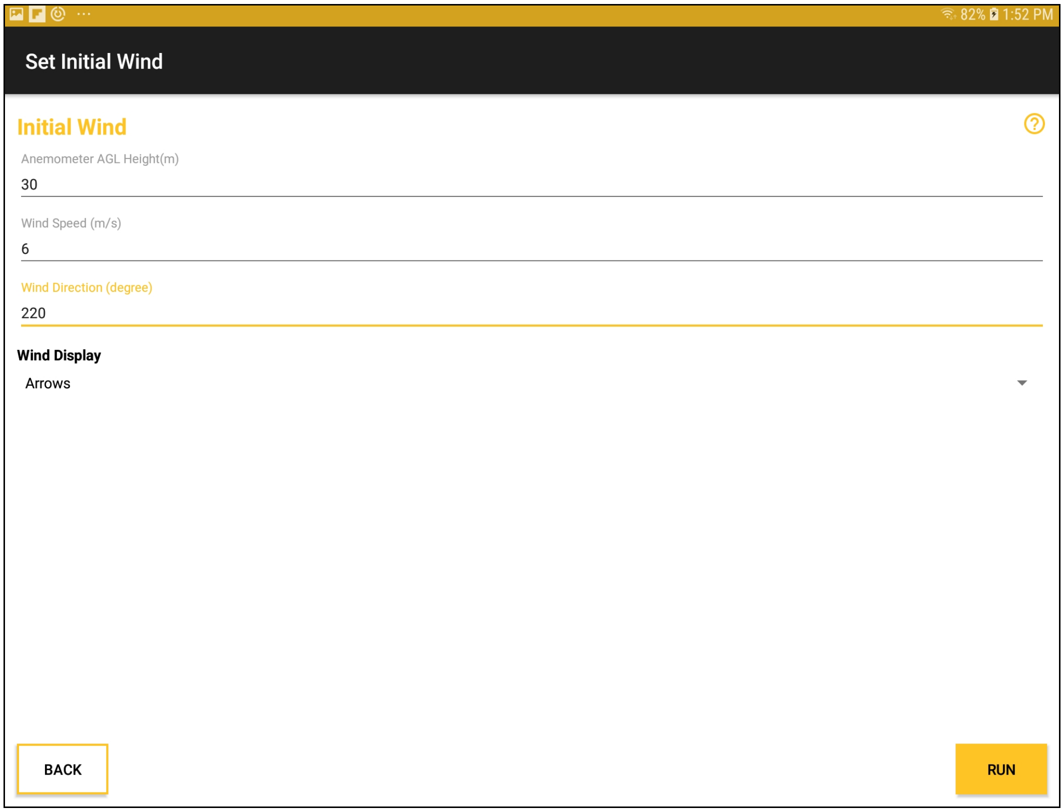
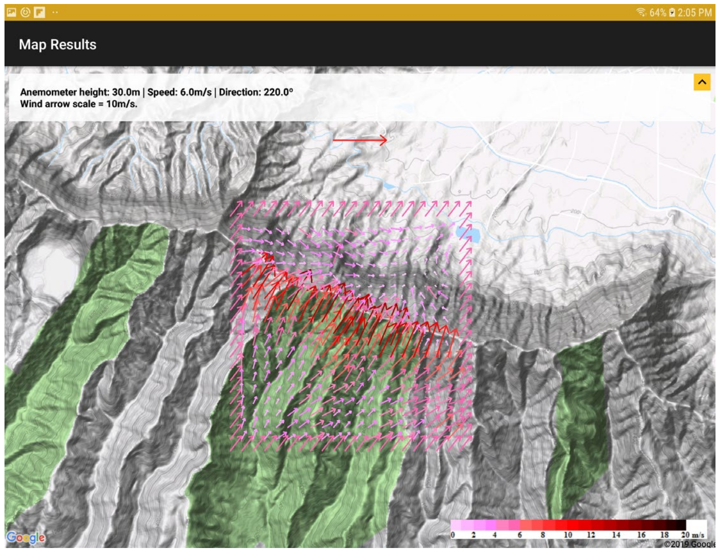


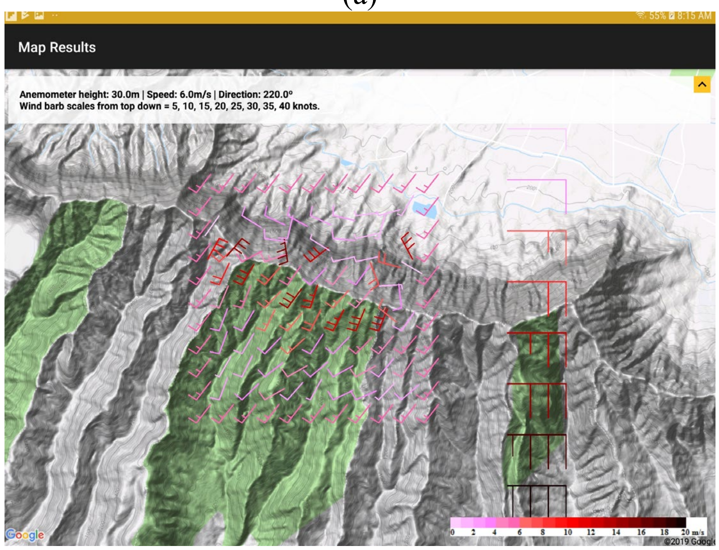
Fig. 14 Set Initial Wind input page

5) For this low-resolution example, it would take up to 8 min from the start to render the wind field in arrow style, as shown in Fig. 15a. Figure 15b shows the same wind field but in wind barbs style, as would be shown if the user chose to display wind barbs. As expected by the 3DWF model, the wind flow in Fig. 15a increases speed drastically as it climbs up the mountain and slows down after it passes the mountain ridge. An individual wind arrow's speed can be approximated either by the color bar or the 10-m/s arrow scale located just above the domain. Initial wind information displayed at the top can be hidden by toggling the yellow up or down arrow at the top-right corner.

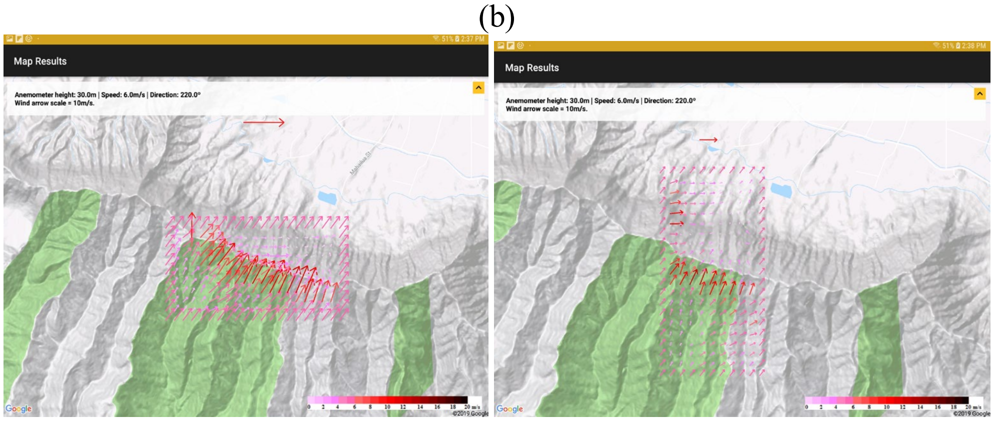
   As mentioned previously, the 3DWF app can also produce wind fields over rectangular domains. Figure 15c shows an example of the result wind fields over two different rectangular domains.



(a)



(b)



(c)

Fig. 15 a) 2-D wind field at 30 m AGL displayed in arrow style for a low-resolution case in a mountainous area with the given initial wind data shown at the top, b) wind field displayed in wind barbs style, and c) wind field displays for two different rectangular domains

6) For high-resolution cases in which the given horizontal grid separation dx and dy are equal or less than 10 m, the 3DWF model must apply wake parameterization at a given number of buildings or forest canopies, which potentially alter the wind flow in both speed and direction. Thus, after the outlined domain is displayed, the app must go through a building rasterizing process page before opening the “Set Initial Wind” input page. For this example, the chosen domain’s center is plotted at a location near downtown Honolulu in Hawaii, as shown in Fig. 16. The domain size is the same as in the low-resolution case except replacing both dx and dy with a value of 10 m for denser grids. The user should zoom-in as close as possible at the center point in order to locate and estimate the number of buildings to be rasterized.

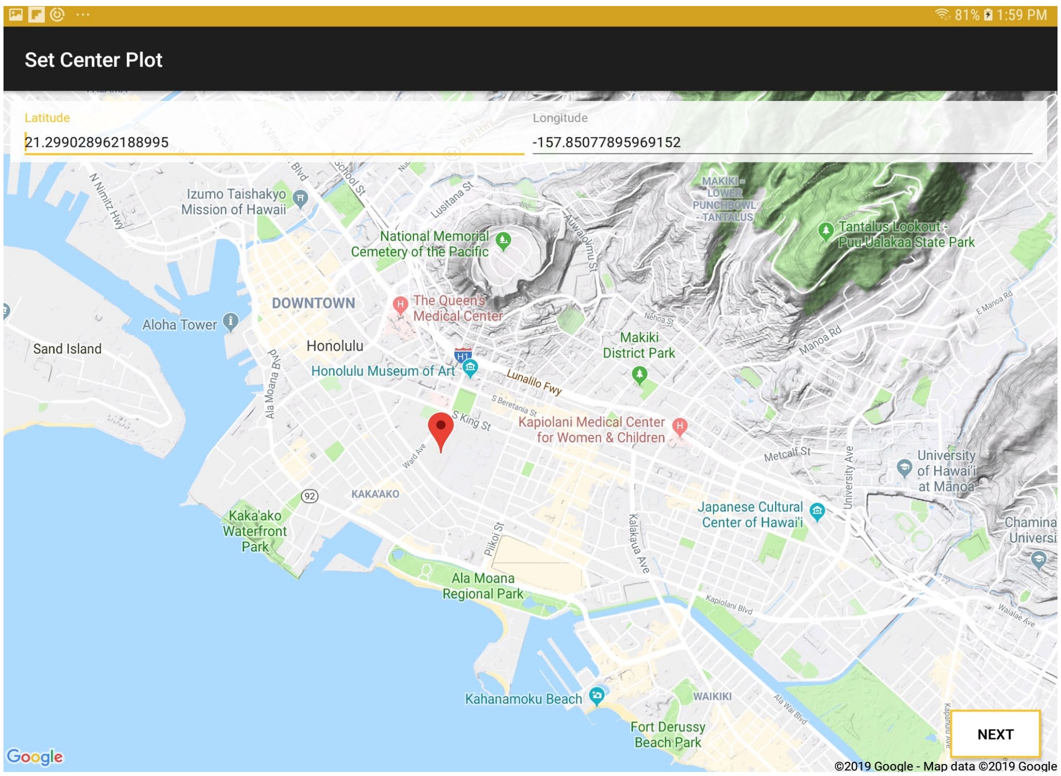


Fig. 16 Center-point plot near downtown Honolulu, Hawaii, for a high-resolution case

7) After the 3DWF app detects dx = dy = 10, it will open “Set Building Corners” page. For this example, only four tall buildings are considered, one pentagonal building with five sides and three other buildings with four sides in box shapes. First, the user needs to enter the total number of buildings, “4”, in the first slot of Fig. 17 to activate the drop-down list for the inputs of each building. From this building list, the user must complete each building’s information before going on to the next building down the

list. First, the user enters the first building's height in meters and then selects a building's type, either 4 sides or 5 sides. The building types in the model's code are defined as 1 or 2 (for four or five sides, respectively) as the order number they appeared in in the drop-down list of building type. Last, tap on "PLACE CORNERS" to display the zoom-in domain in Google Earth (similar to Fig. 18) for plotting the corners of each building.

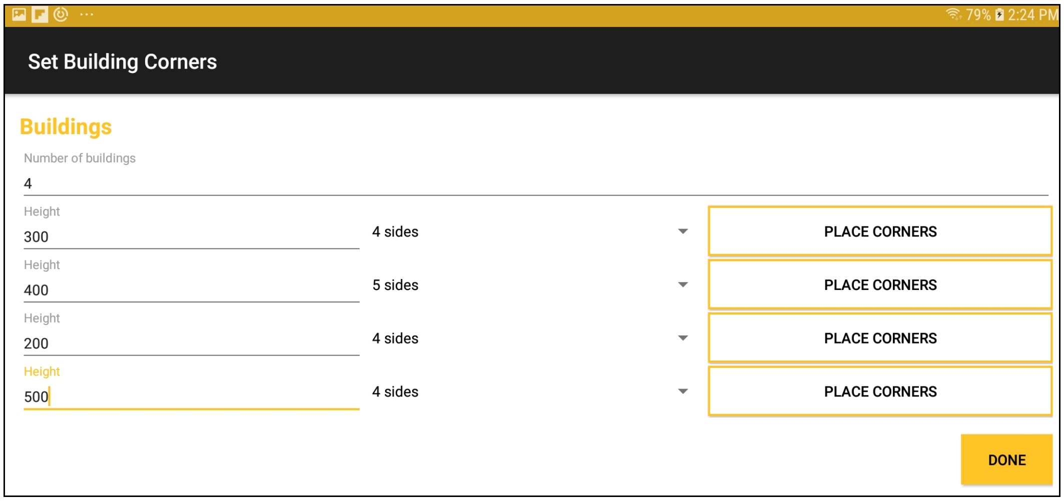


Fig. 17 Building information page

Fig. 18 Four plotted buildings displayed in Google Earth

Once finished for all buildings, tap DONE to display all plotted buildings, as shown in Fig. 18. For this Android app, the digitizing process of buildings is built in and no longer relies on a separate GUI like the one for the Windows version. The user can use fingers to zoom in as close as possible so that the building's corners can be easily plotted by a fingertip. As mentioned in the Java code, building corners for a box shape must be plotted clockwise starting from the leftmost corner. However, if box shape building has a north-south orientation, then plotting must begin at the upper-left (northwest) corner. For a pentagonal building, plotting must start at the topmost corner. There is a prompted message to help with plotting building corners. The app will translate the corner's latitude, longitude, and height into grid numbers and levels to be read by the 3DWF model.

The detailed grid information of these four plotted buildings is saved internally in the 3DWF app, but if the user wishes for future reference, it can be saved in a file, named building_suffix.txt. The content of this file is as follows:

```
4
1
9     79
15    83
20    75
14    72
2
32
2
43    69
49    67
49    61
43    59
38    63
2
42
1
16    18
23    23
31    12
25    8
1
21
1
67    55
73    52
71    47
65    50
2
52
```

Number "4" in the first row is the total number of buildings. After the first row is a sequence of plotted buildings. Each building starts with building type 1 or 2, as explained previously. The next four or five rows for four- or five-sided building have two columns identifying the building corners' x,y grid coordinates in the order in which they are plotted. The next two rows contain the vertical level numbers of the building's base and the building's top.

From the Google Earth display of four outlined buildings in Fig. 18, tap Next to open the "Set Initial Wind" page as described in the low-resolution case. If given the same input values as in the low-resolution case, the resultant wind field generated is as shown in Fig. 19.

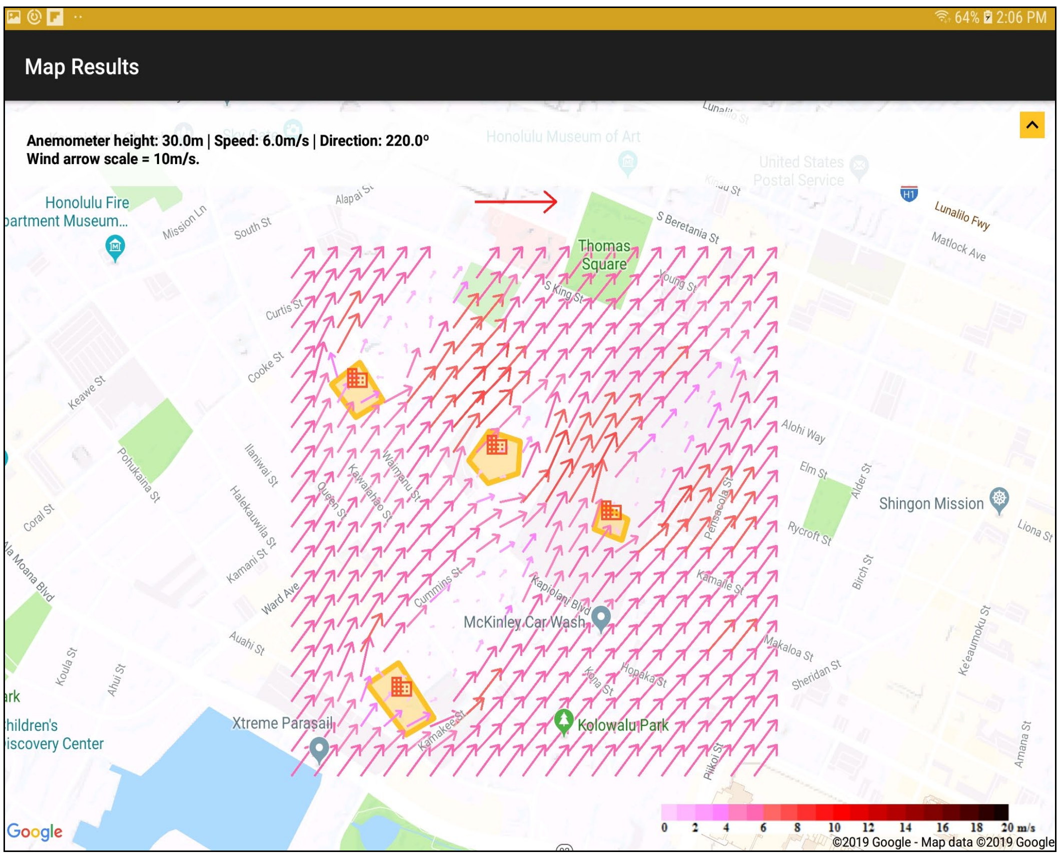


Fig. 19 Wind field for the high-resolution case with wake parameterization applied to the four outlined buildings

The 3DWF model predicts that wind flow patterns are altered as they approach the buildings based on the shape and size of the building and whether the wind is perpendicular to the leading wall or the leading corner. Due to the building wake effect at each building, wind speeds are reduced drastically behind the buildings in comparison to the relatively constant speeds throughout the unaffected areas of this relatively flat urban terrain.

4. Windows App for the 3DWF Model

Since the Microsoft Surface tablet shares the same Windows platform as the Dell Windows desktop, the 3DWF GUI and the 3DWF model originally created from this desktop can be migrated directly to the Surface tablet. Therefore, the Windows app builder software was not necessary for this project. The 3DWF app deployed on the Surface tablet is almost identical to the 3DWF GUI installed on the Dell Windows desktop. The 3DWF GUI was designed in JavaScript as an HTML file and interfaced with the 3DWF model's Fortran or Java executable code. Files for the 3DWF GUI and the executable code of the 3DWF model can be copied directly from the desktop to the Surface tablet via a USB cable. For the Fortran version of the 3DWF model, in order for its executable file to run successfully on the Surface tablet, the model's Fortran code must be recompiled using a Fortran compiler for Windows PC such as the Visual Studio 2010 IDE with the option "Debug Multithreaded (/libs:static /threads /dbglibs)" instead of "Debug Multithreaded DLL (...)" in the Project-Properties-Fortran-Libraries-Runtime library. However, since Java is the language of the Android app and in order to use the same language on both platforms, the Java version of the 3DWF was adopted for its Windows app. The 3DWF's Java programs were recompiled using NetBeans IDE software and its Java executable code "3DWF.jar" was executed by issuing the following commands:

set path="C:\Program Files\Java\jdk1.8.0_161\jre\bin"

java -Xmx8192000k -jar "C:\\3DWF\\3DWF.jar

The Windows 3DWF app also needs to execute a separate Java program named "Rasterize16.jar", written to rasterize building and forest canopies. This program provides a file named "building.dat", which contains the same information as the file "building_suffix.txt", as described in the procedures for the Galaxy tablet in Section 3. This building rasterization program and its GUI can be executed to via the option "Rasterize-Morph" in the 3DWF app's main menu page.

The Windows version of the 3DWF app consists of a series of HTML files written in JavaScript: 3DWF_GUI.html, domain_para.html, met_initial.html, and display_wind.html. There are also HTML files linked to by these files, namely, Bldg_GMaps_v3New.html and Cano_GMaps_v3New.html, for rasterization of buildings. All files for this app must reside in the hardcoded subdirectory C:\\3DWF of the Surface tablet.

The following are the procedures for operating the 3DWF App on the Surface tablet.

The user can start this app by double tapping or clicking on the file 3DWF_GUI.html or its shortcut globe icon, as shown in Fig. 20, to open its main menu page, as shown in Fig. 21.

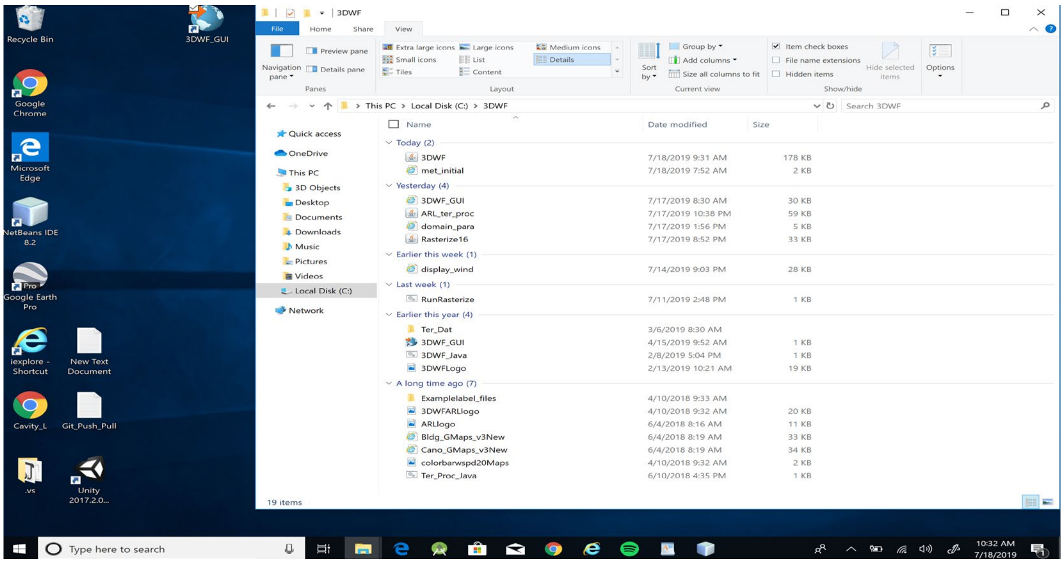


Fig. 20 Files and icon for the 3DWF app on the Surface tablet

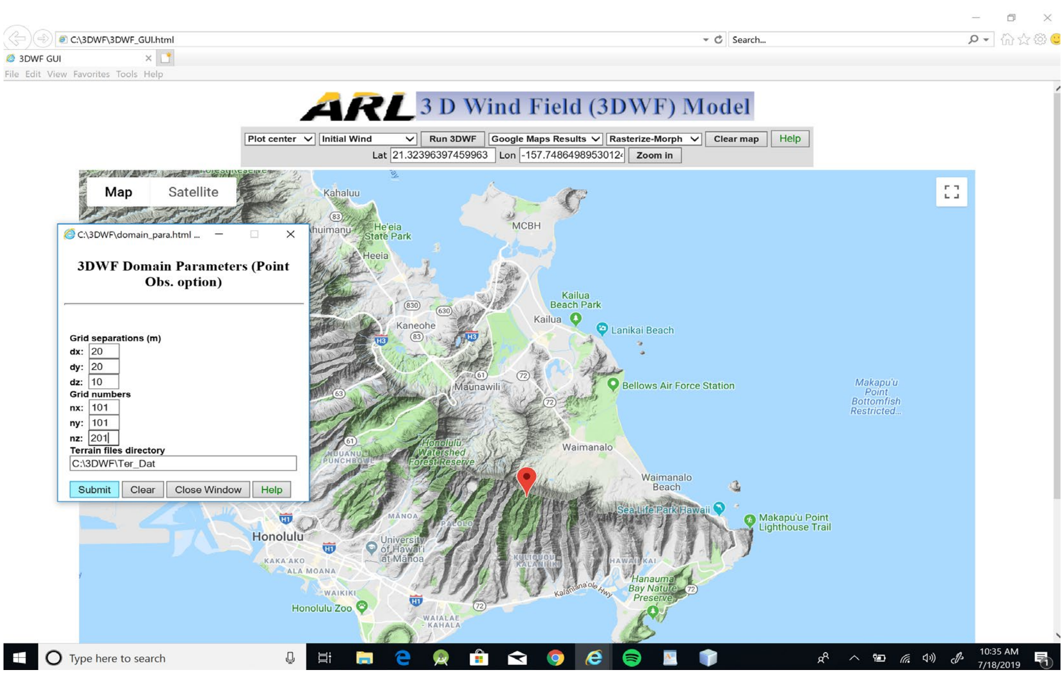


Fig. 21 Main page of the 3DWF app on the Surface tablet

The main menu page consists of a simple menu at the top and a Google Maps area showing the hardcoded default location in Hawaii, where the relevant SRTM data tiles have been downloaded and stored under C:\\3DWF\\Ter_Dat. During this run, there are pop-up messages asking "Allow blocked content" about allowing ActiveX controls to run. These messages are security measures from the Internet Explorer (IE) browser that occur every time the app accesses local files for reading or writing. The user must always allow by answering "Yes" to these messages in order to continue. Since direct access to local drive is not advisable while connecting to the Internet, this Windows app version must only be operated by trusted users and with online security measures in place. Some of the following operating steps for the Windows app differ from those described for the Android version but they accomplish similar tasks:

1) First, the user must decide which case, low or high resolution, to run in order to select the appropriate buttons on the menu. There are help messages during the inputs for domain parameters to guide the user with this. It is considered a high-resolution case when the given horizontal grid separations dx and dy are equal or less than 10 m, as explained previously in step 6 of Section 3.4. In both low- and high-resolution cases, the product of dz and nz also must satisfy the condition mentioned in step 3 of Section 3.4. Next, tap on the first option "Set Domain" on the menu and select "Plot center" to select the center point for the 3DWF domain. After that, use a finger to tap a location of interest, which will be marked with a red balloon as shown in Fig. 21. A pop-up window appears as shown to the left of Fig. 21 to allow the user to enter parameters for a domain of interest. To configure it to carry out the simulation shown in the Android app example, enter dx = dy = 10, nx = ny = 101, and nz = 201. The last slot "Terrain files directory" is the directory where 3DWF looks for SRTM data files and is prepopulated with the default location for these files.

   Once done, tap "Submit" to execute the file "ARL_ter_proc.jar". This executable code finds all involved SRTM data tiles and creates the domain data file "ht_lat_long_3DWF_domain.dat" for the 3DWF model. After the domain data file is created, the user can select the next option "Outline" under "Set Domain" to outline the domain.

2) For a low-resolution case (dx > 10 m), the user can select "Point Observation" under the option "Initial Wind" to pop up an entry page for the initial wind setup, as shown in Fig. 22.

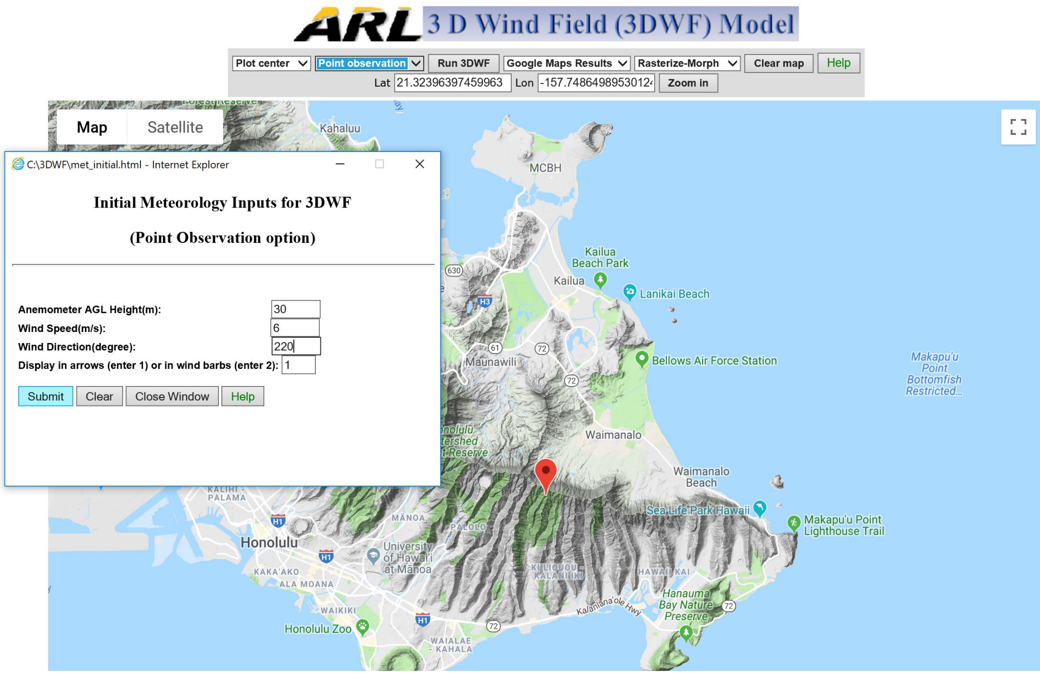


Fig. 22 Initial wind data input page

Once done filling out the initial wind input page, tap on the "Submit" button to save and then tap on the option "Run 3DWF" to run the model. Once the 3DWF model is finished running, select the display style option "in arrows" under "Google Maps Results" to display the resultant wind field in Google Maps, as shown in Fig. 23 (top). Similar to its Android version, the 3DWF Windows app also allows one to display in wind barbs and on rectangular domains. Figure 23 (bottom left) displays the resultant wind field in wind barbs, and Fig. 23 (bottom right) shows an example of the resultant wind field over a rectangular domain (nx = 101, ny = 51). The darkened background of this Google Maps display indicates that this Surface tablet needs to subscribe to an official copy of Google Maps application programming interface (API) software. According to Google's current policy (https://cloud.google.com/maps-platform/pricing), a Google Maps user must subscribe to an account using credit card information. Since Google Maps was preloaded as a standard feature on the Galaxy tablet, its user does not need to subscribe.

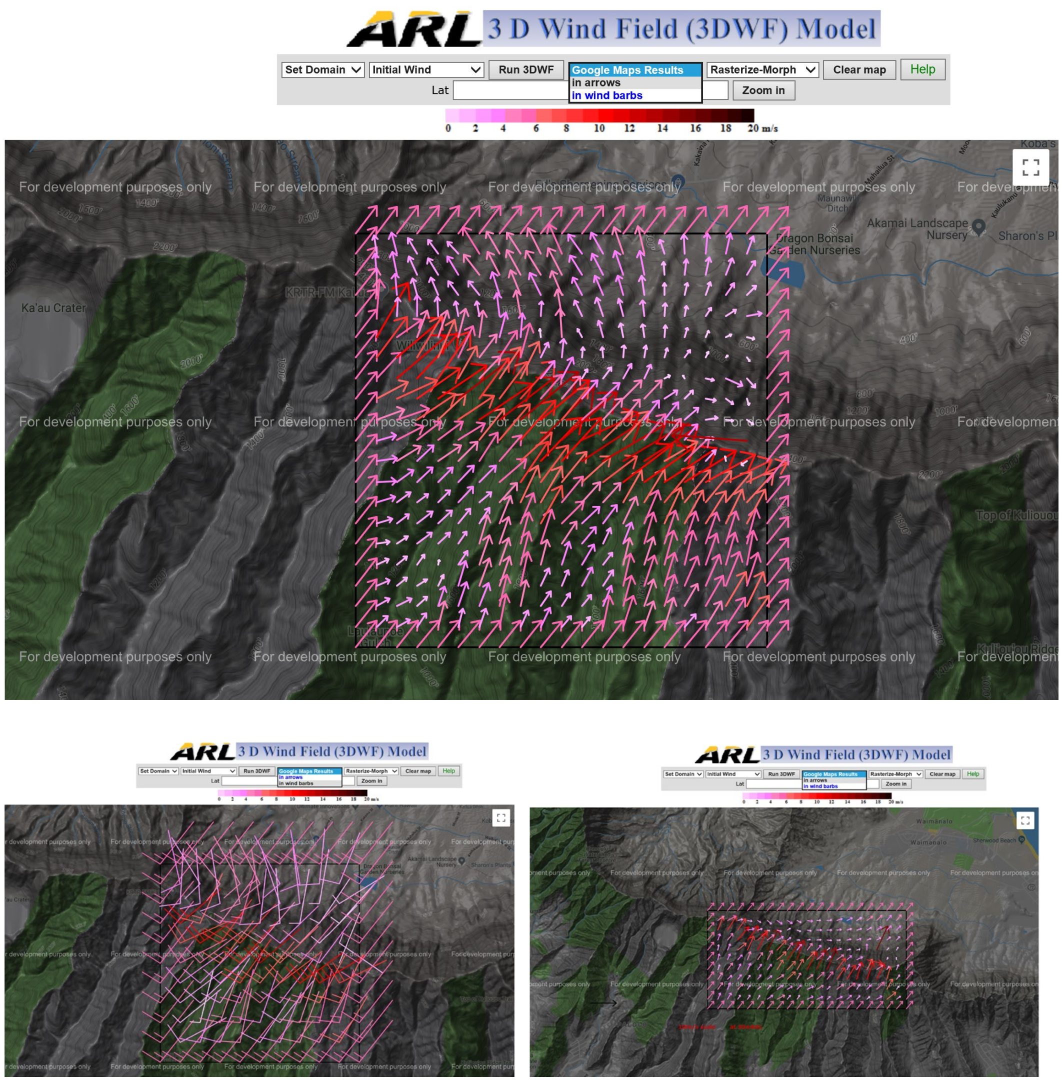


Fig. 23 Resultant wind fields in arrow style (top), wind barbs style (bottom left), and arrow style over rectangular domain (bottom right) at 30 m AGL

3) For a high-resolution case (dx ≤ 10 m), as mentioned previously, the user must identify building corners' grid coordinates by utilizing another GUI customized for this purpose (Huynh et al. 2010). Therefore, after finishing the "Initial Wind" step, the user must go to the option "Rasterize-Morph" to rasterize the buildings and create a file "building.dat" containing the buildings' grid numbers for the 3DWF program. Building wake parameterization is then performed and the resultant wind field displayed. This rasterize option also produces a morphological data file and Google Earth display of the 3-D polygons of the buildings. Under "Rasterize-Morph", there are two options, "Building polygon" and "Canopy polygon", to link to two similar rasterization GUIs designed for buildings and forest/vegetation canopies. For the following example, option "Building

polygon" was selected for an urban domain in Honolulu, Hawaii. The linked GUI, "Bldg_GMaps_v3New.html", is shown in Fig. 24.

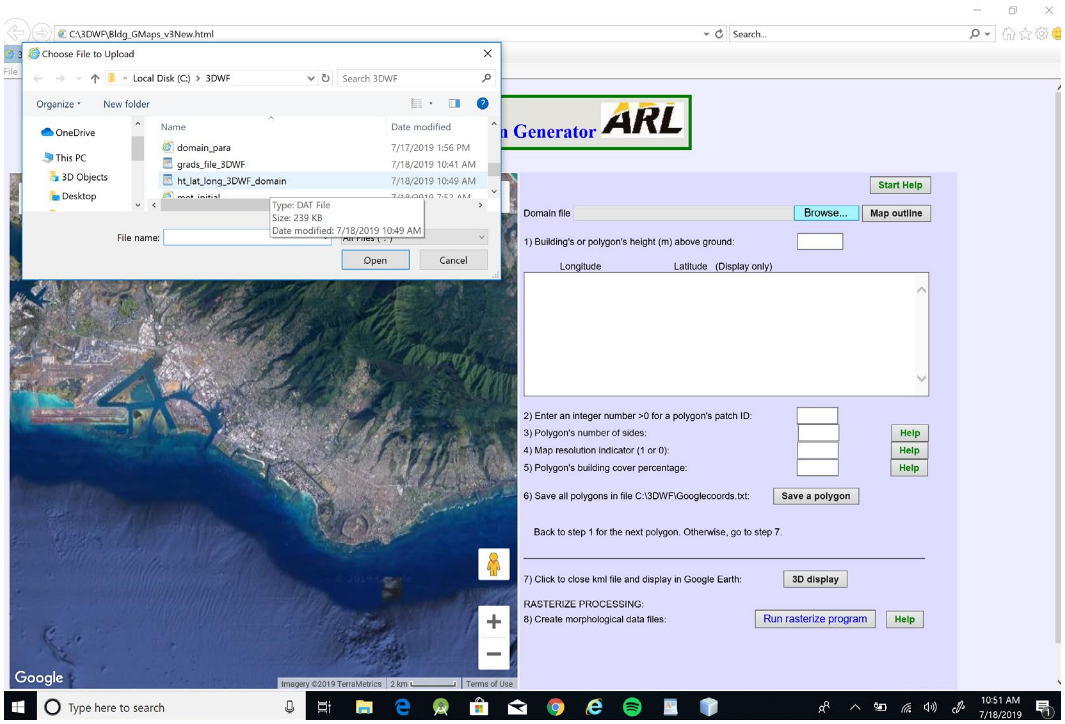


Fig. 24 Building Polygon Generator GUI

On this Building Polygon Generator GUI, click on "Browse..." to select the 3DWF domain data file "ht_lat_long_3DWF_domain.dat" just created from the "Set Domain" step. Next, click on the "Map outline" button to display the domain's outline in Google Earth so that large visible building roofs inside the outline can be observed for corner plotting. The user can zoom in as close as possible so that a building's corners are visible enough to be plotted by a fingertip or moving a cursor. All steps from 1 through 6 shown on this Building Polygon Generator GUI in Fig. 24 must be completed for each building. For example, enter "300" for the height of the first building. Next, plot the four corners of a building of interest, as shown in Fig. 25. The four corners' longitude and latitude coordinates appear in the large slot reserved for this information. Patch ID is any integer number to label a group of building. This building's number of sides is "4". The map resolution indicator is either "1" for building rasterization or "0" for forest canopies rasterization. The polygon's building cover percentage is the percentage of the overlaid polygon (created by the plotted points) that is covered by the roof of the building. It is usually 100% as this represents a polygon completely covered by the building's roof. Click on "Save a polygon" to finish one building. The user must go back to step 1 to enter a height for the second building and repeat all the steps again.

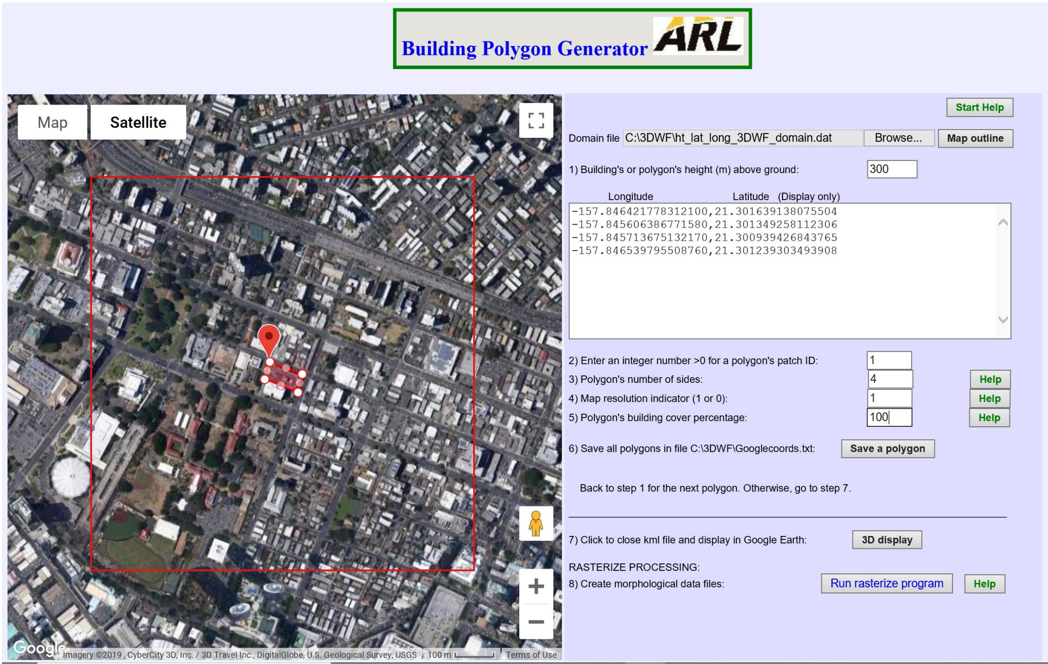


Fig. 25 Rasterization process of a four-sided building

After all building polygons are drawn, tap or click on the button "3D display" to see the 3-D display of all building polygons in Google Earth. The last step "Run rasterize program" is to execute the "Rasterize16.jar" file in order to produce the "building.dat" file for the 3DWF model and also create and save morphological data files for other purposes.

4) Back on the main 3DWF menu page, click "Run 3DWF" to execute the 3DWF model's program "3DWF.jar". Once done, select "in arrows" under the next option "Google Maps Results" to display the resultant wind field over the domain in wind arrow style. If displaying in wind barbs is preferred, the user can select the "in wind barbs" option and rerun the option "Run 3DWF" to generate a new display. Figure 26 displays the resultant wind field influenced by a hypothetical 300-m high-rise building polygon located near the center of the domain.

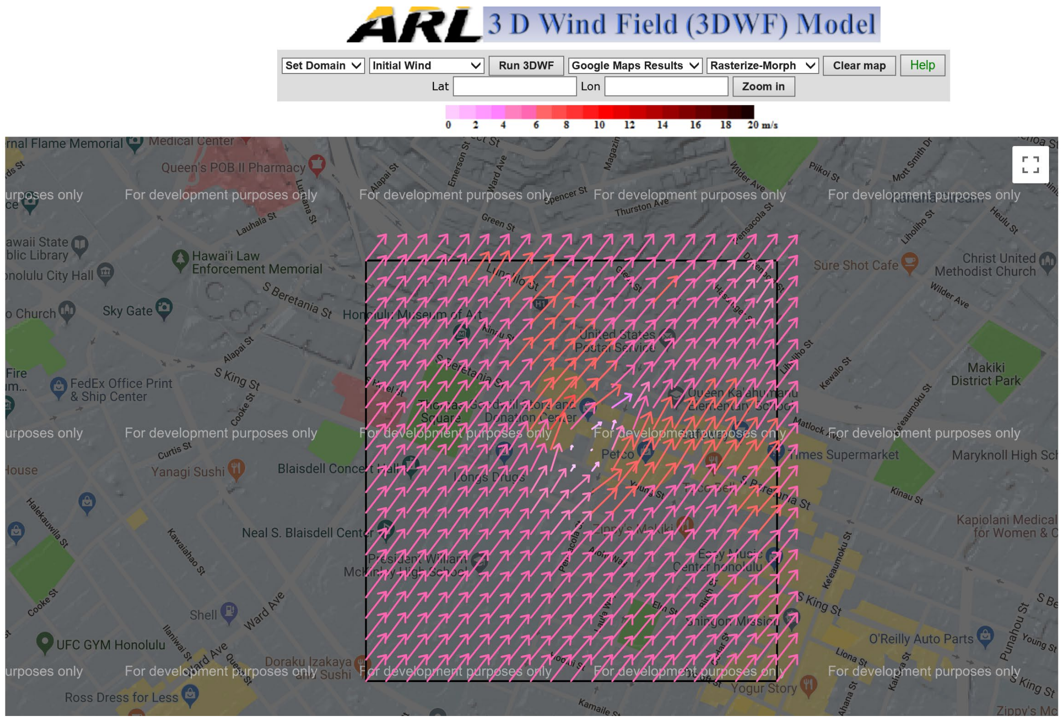


Fig. 26 Resultant wind field at 30 m AGL influenced by building wake at a single 300-m-high building

5. Conclusions

The 3DWF model and its Google GIS-based GUI developed by ARL has been a useful tool for researchers to predict and visualize the predicted wind flow over complex terrain and generate local geo-morphological data. However, it has been operated only on a Windows desktop computer. In recognizing the advantages and the convenience of the current popular handheld mobile devices, such as the Android platform Galaxy S3 tablet and the Windows platform Surface Pro tablet, the Atmospheric Modeling Branch at CCDC ARL has decided to develop mobile apps for the 3DWF on these two devices. Since the Windows Surface tablet shares the same platform as the Windows desktop, the migration of the 3DWF model and its GUI to the Surface tablet was straightforward and without many changes. However, since the Android app for the Galaxy tablet does not allow the use of the insecure ActiveX controls feature used in the 3DWF GUI for Windows, this GUI cannot be migrated to it and the 3DWF app for the Galaxy had to be redesigned and rebuilt by using Android Studio IDE software. CCDC ARL has been collaborating with the Institute for Human & Machine Cognition, and has successfully created and deployed a 3DWF app on the Galaxy S3 tablet. Based on many test runs to validate their performances, they have shown that although the Windows app ran the 3DWF model faster and the Surface Pro tablet can handle larger domains, the

3DWF app deployed on the Android Galaxy S3 tablet operated in a much simpler fashion due to its sequential page design. The Galaxy tablet is also more compact and lighter, and the user does not have to deal with frequent access to a local drive, with the exception of reading SRTM terrain data files and producing the model's output data file. Another advantage of an Android device is the included standard GIS software, such as Google Maps and Google Earth, which must be purchased to be installed on a Windows Surface Pro tablet. Nevertheless, both of these apps are big steps forward in an effort to modernize the use of the 3DWF model, enable multiplatform operations, and provide a learning experience in preparation for future implementation of CCDC ARL's next-generation microscale model, ABLE-LBM, on the mobile devices.

6. References

Huynh G, Wang Y, Williamson C. Building and vegetation rasterization for the three-dimensional wind field (3DWF) model. Aberdeen Proving Ground (MD): Army Research Laboratory (US). 2010. Report No.: ARL-TR-5419.

Wang Y, Huynh G, Williamson C. Integration of Google Maps/Earth with microscale meteorology models and data visualization. Computer and Geosciences. 2013;61:23–31.

Wang Y, MacCall B, Hocut C, Zeng X, Fernando HJS. Simulation of stratified flows over a ridge using a lattice Boltzmann model. Environmental Fluid Mechanics. 2018. doi: 10.1007/s10652-018-9599-3.

Wang Y, Williamson C, Garvey D, Chang S, Cogan J. Application of a multigrid method to a mass consistent diagnostic wind model. J Appl. Meteorology. 2005;44:1078–1089.

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
3DWF	Three-Dimensional Wind Field
AGL	above ground level
ARL	Army Research Laboratory
CCDC	Combat Capabilities Development Command
CPU	central processing unit
GIS	Geographic Information System
GUI	graphical user interface
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
IE	Internet Explorer
IHMC	Institute of Human & Machine Cognition
OS	operating system
PC	personal computer
RAM	Random Access Memory
SRTM	Shuttle Radar Topography Mission
USB	universal serial bus

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	CCDC ARL FCDD RLD CL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
4 (PDF)	CCDC ARL FCDD RLC E B MACCALL FCDD RLC EM G HUYNH Y WANG E CHIN
1 (PDF)	IHMC K REHOVICOVA