ORNL/TM-2017/145

PopLib Unit Test Report For HPAC Version 6.4

Ronald W. Lee

March, 2017

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2017/145

Computational Sciences and Engineering Division

POPLIB UNIT TEST REPORT FOR HPAC VERSION 6.4

Ronald W. Lee

Date Published: March, 2017

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831-6285
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ACRONYMS

API	Application Programming Interface
DLL	Dynamic Link Library
DTRA	Defense Threat Reduction Agency
HPAC	Hazard Prediction and Assessment Capability
MGRS	Military Grid Reference System
SCIPUFF	Second-order Closure Integrated Puff Model

1. OVERVIEW

This document describes the methodology for and results of testing the PopLib component in Version 6.4 of the Hazard Predication and Assessment Capability (HPAC). PopLib's use in HPAC is solely as a dynamic link library (DLL) called from the Second-order Closure Integrated Puff Model (SCIPUFF).

PopLib is a cross-platform capability that is built using CMake on Windows, Mac OS X, and Linux platforms. In addition to the shared object (file with .so extension under Linux and .dylib extension under Mac OS X) or dynamic link library (DLL) (Windows file with .dll extension), a command-line utility, poplib_run is also created. The utility can be used to invoke any/all PopLib capabilities with files defining polygons passed as inputs. Most PopLib unit tests execute poplib_run with results compared against baseline files.

For testing, PopLib is built under Windows using a CMake out-of-source build in a simple four-step process, with one additional step to execute the tests. The steps below assume the source directory is current at the start.

```
> mkdir build
> cd build
> cmake -G "NMake Makefiles" ..
> make

(final step to run tests)
> make test
```

2. TEST DRIVER COMPONENTS

In addition to the poplib_run command-line utility, two components are needed for PopLib testing: a test driver implemented with Python-2, and a special test for Military Grid Reference System (MGRS) conversions mgrs_test.

2.1 PYTHON TEST DRIVER

The Python driver poplib_test_driver.py is listed in Appendix A. It takes care of executing poplib_run with the proper options and inputs and comparing test results to baselines.

2.2 CMAKE LIST FILE

As per CMake/CTest design, individual tests are defined in a CMake ”list” file, CMakeLists.txt, listed in Appendix B. Note the compile and link of mgrs_test is also specified in this file. Three macros aid in defining tests:

- add_exclusion_test()
- add_mgrs_test()
- add_simple_test()

Generally, test definitions include specification of:

- Population data directories to read,
- Input polygon files,
- Special options for poplib_run, and
- The allowed test timeout.

3. TEST APPROACH

All PopLib tests are pure unit tests that are automated and unattended, executed with CMake and CTest.[1] A total of 62 tests are run upon each build of PopLib on all three target platforms. Note the same baseline files are used for all platforms. Many tests represent pairs of the same test using tile and raster data, respectively. There are also many tests that exercise the exclusion polygon capability added for HPAC-6.3, some of which include the fixed population feature added for 6.3 Patch 2. Additional tests focus on exclusion polygons defined with Military Grid Reference System (MGRS) coordinates.

Population data against which PopLib is exercised are included in the source code repository. They are arranged in directories whose names are self-explanatory:

- rasters
  - levant
  - mosul
  - ne-usa
- rasters-only
- tiles
  - levant
  - ne-usa
- tiles-and-rasters

4. TEST DESCRIPTIONS

4.1 SIMPLE POPULATION COUNTING

Test0001_dc_around_tiles. This is a simple test of counting the population within a polygon using tile files. The population data and input polygon are depicted in Fig. 1.

Test0002_dc_around_rasters. Repeats Test0001 with raster population files.

Test0003_dc_inner_tiles. This is a simple test of counting the population within a polygon using tile files. The population data and input polygon are depicted in Fig. 2.

Test0004_dc_inner_rasters. Repeats Test0003 with raster population files.

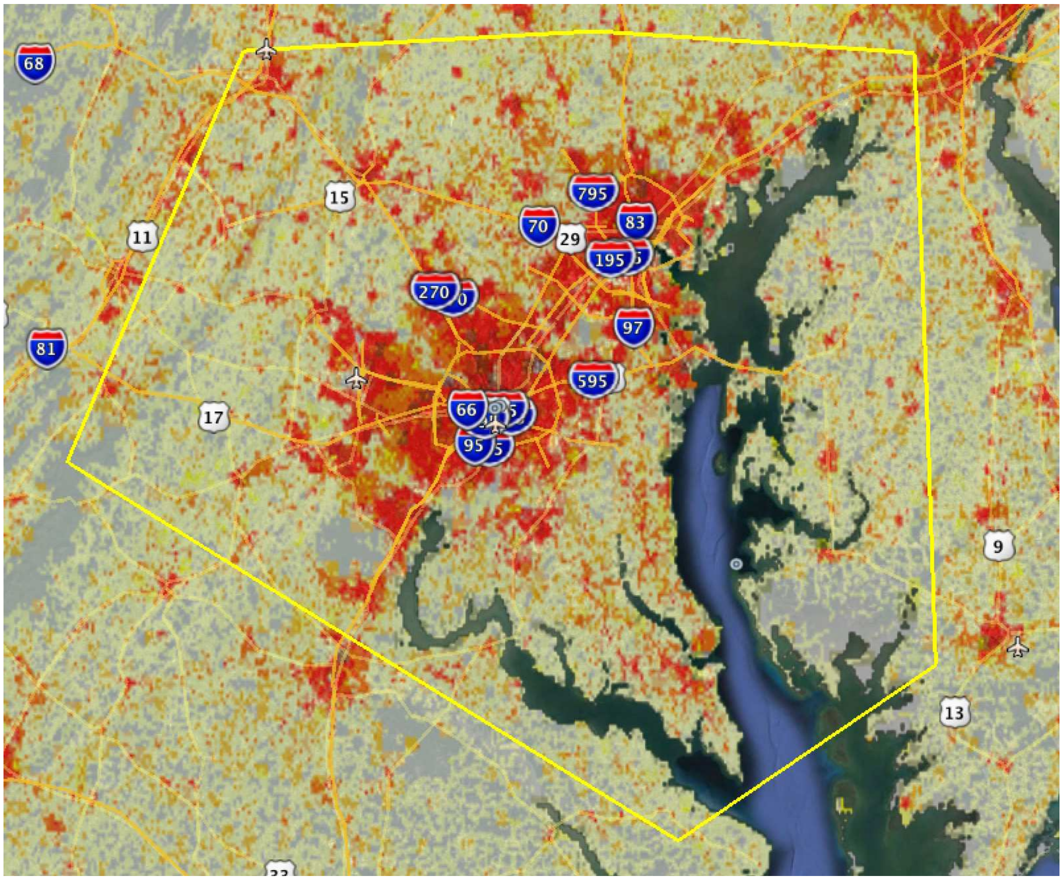

Fig. 1: Tests 0001 and 0002.

Test0005_dc_diamond_tiles. This is a simple test of counting the population within a polygon using tile files. The population data and input polygon are depicted in Fig. 3.

Test0006_dc_diamond_rasters. Repeats Test0005 with raster population files.

Test0007_ny_triangle_tiles. This is a simple test of counting the population within a polygon using tile files. The population data and input polygon are depicted in Fig. 4.

Test0008_ny_triangle_rasters. Repeats Test0007 with raster population files.

Test0033_i258_tiles. This is a simple test of counting the population within a complex polygon using tile files. The population data and input polygon (white) are depicted in Fig. 5.

Test0034_i258_rasters. Repeats Test0033 with raster population files.

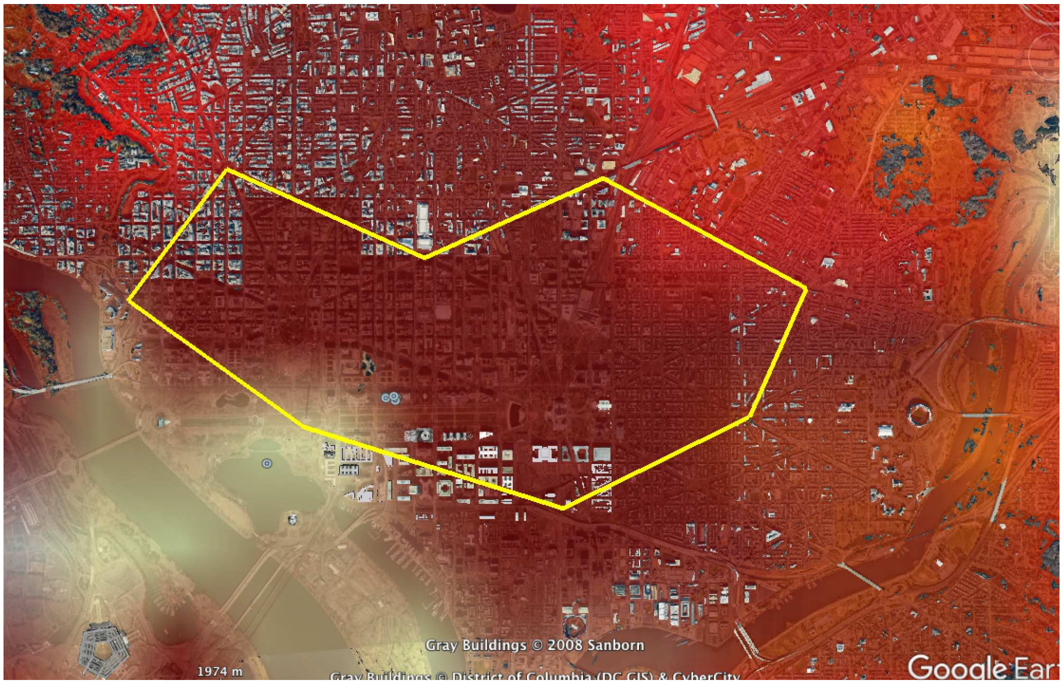
Gray Buildings © 2008 Sanborn
1974 m
Gray Buildings © District of Columbia (DC GIS) & CyberCity
Google Ear

Fig. 2: Tests 0003 and 0004.

4.2 POPULATION COUNTING WITH EXCLUSIONS

Test0009_va_exclude_1_tiles. Tests population counting with an exclusion polygon file specified via the PopLib_exclusionFile environment variable. Population data are in tile files. The population data, input polygon (yellow), and exclusion polygon (red) are depicted in Fig. 6.

Test0010_va_exclude_1_tiles_ini. Duplicates Test0010 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0011_va_exclude_1_rasters. Duplicates Test0009 with population data as rasters.

Test0012_va_exclude_1_rasters_ini. Duplicates Test0010 with population data as rasters.

Test0013_va_exclude_2_tiles. Tests population counting with an exclusion polygon file specified via the PopLib_exclusionFile environment variable. Population data are in tile files. The population data, input polygon (yellow), and exclusion polygon (red) are depicted in Fig. 7.

Test0014_va_exclude_2_tiles_ini. Duplicates Test0013 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0015_va_exclude_2_rasters. Duplicates Test0013 with population data as rasters.

Test0016_va_exclude_2_rasters_ini. Duplicates Test0014 with population data as rasters.

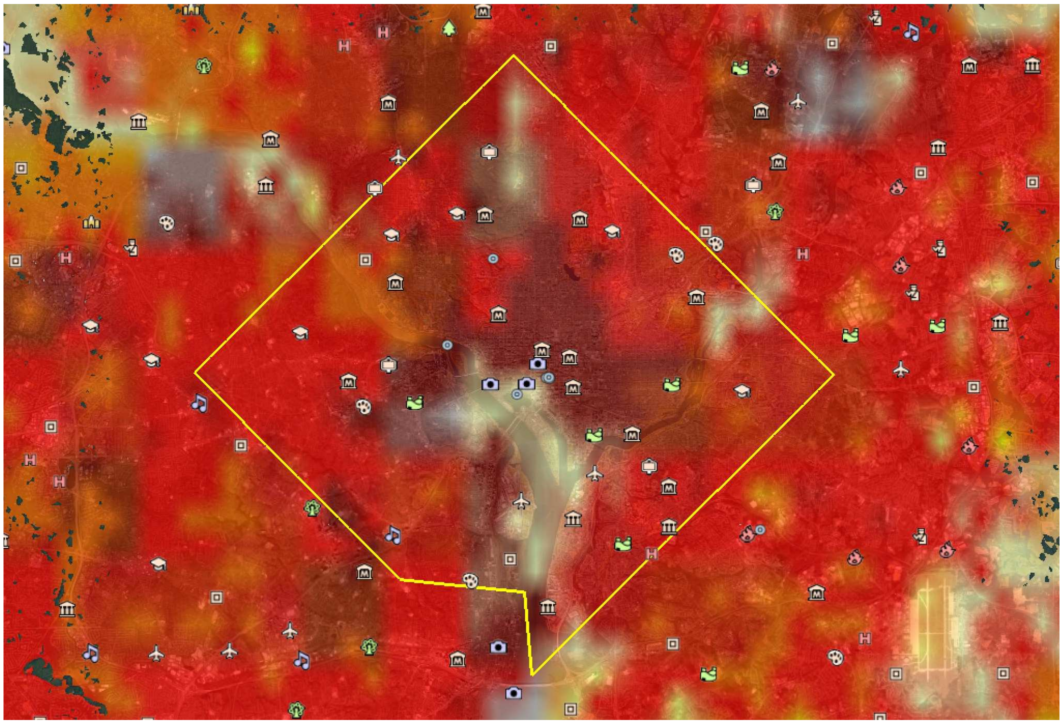

Fig. 3: Tests 0005 and 0006.

Test0017_va_exclude_12_tiles. Tests population counting with an exclusion polygon file specified via the PopLib_exclusionFile environment variable. Population data are in tile files, and the exclusion includes both polygons used in Test0009–Test0012 and Test0013-Test0016, respectively. Refer to Fig. 6 and 7.

Test0018_va_exclude_12_tiles_ini. Duplicates Test0017 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0019_va_exclude_12_rasters. Duplicates Test0017 with population data as rasters.

Test0020_va_exclude_12_rasters_ini. Duplicates Test0018 with population data as rasters.

Test0021_va_exclude_1n_tiles. Tests population counting with an exclusion polygon file specified via the PopLib_exclusionFile environment variable. Population data are in tile files, and the exclusion file specifies a population count of 50,000 inside the exclusion polygon. Both the exclusion and input polygons are identical as for Test0009.

Test0022_va_exclude_1n_tiles_ini. Duplicates Test0021 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0023_va_exclude_1n_rasters. Duplicates Test0021 with population data as rasters.

Test0024_va_exclude_1n_rasters_ini. Duplicates Test0022 with population data as rasters.

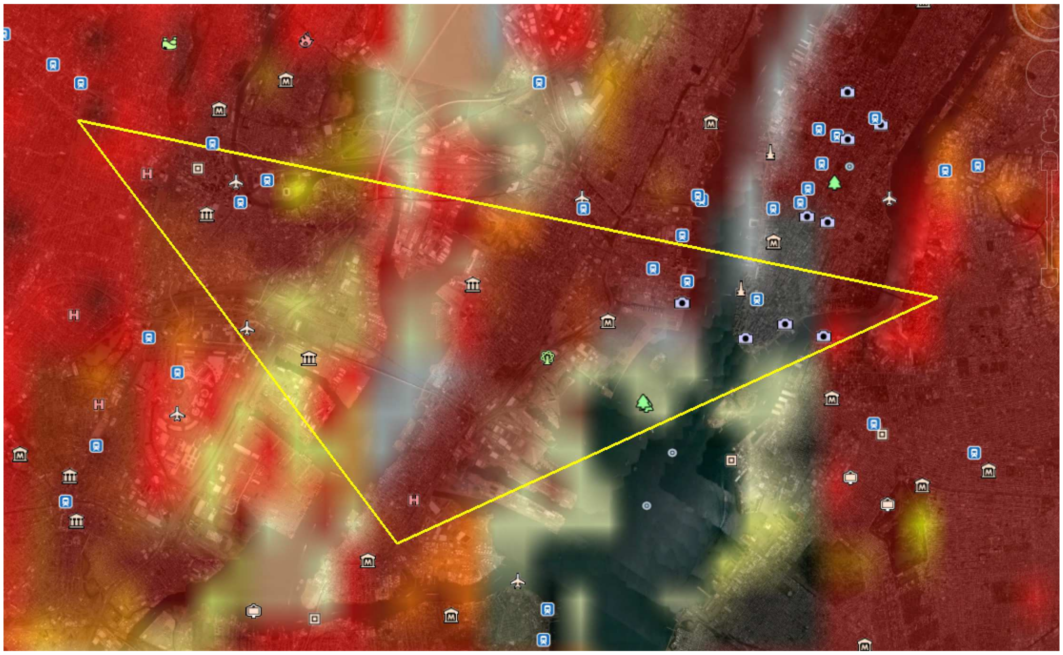

Fig. 4: Tests 0007 and 0008.

Test0025_va_exclude_2n_tiles. Tests population counting with an exclusion polygon file specified via the PopLib_exclusionFile environment variable. Population data are in tile files, and the exclusion file specifies a population count of 65,000 inside the exclusion polygon. Both the exclusion and input polygons are identical as for Test0013.

Test0026_va_exclude_2n_tiles_ini. Duplicates Test0025 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0027_va_exclude_2n_rasters. Duplicates Test0025 with population data as rasters.

Test0028_va_exclude_1n_rasters_ini. Duplicates Test0026 with population data as rasters.

Test0029_va_exclude_12n_tiles. Tests population counting with two exclusion polygons in a file specified via the PopLib_exclusionFile environment variable. Population data are in tile files, and the exclusion file specifies a population count of 100,000 inside the first exclusion polygon. Both the exclusion and input polygons are identical as for Test0017.

Test0030_va_exclude_12n_tiles_ini. Duplicates Test0029 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0031_va_exclude_12n_rasters. Duplicates Test0029 with population data as rasters.

Test0032_va_exclude_12n_rasters_ini. Duplicates Test0030 with population data as rasters.

Test0035_i258_exclude_tech_tiles. Tests population counting in a complex polygon with a

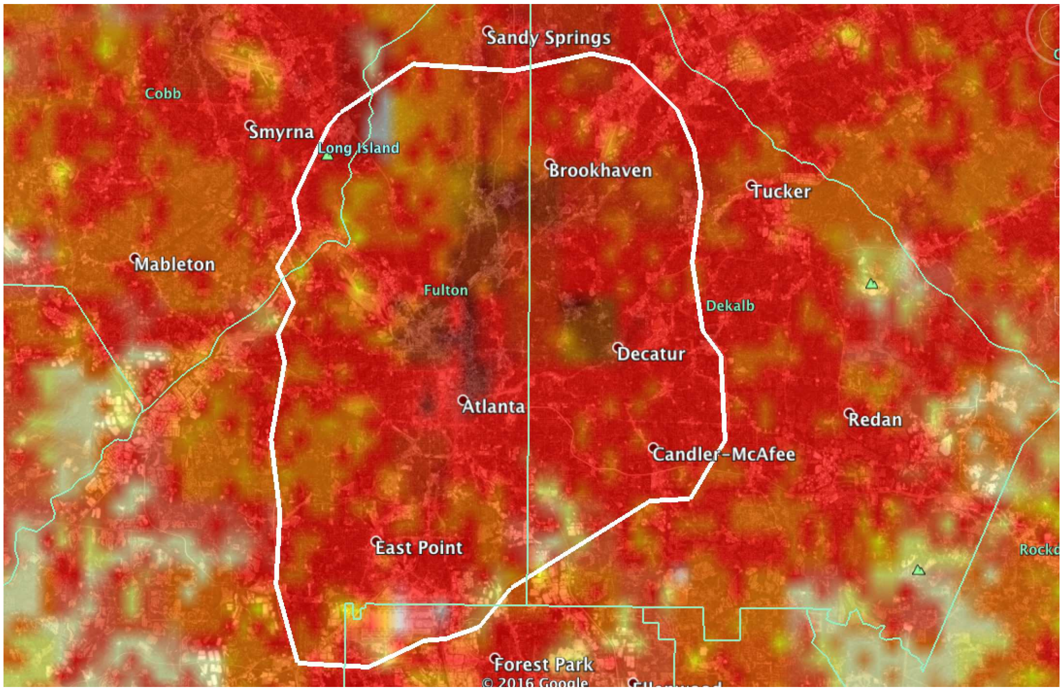

Fig. 5: Tests 0033 and 0034.

wholly contained exclusion polygon specified via the PopLib_exclusionFile environment variable. Population data are in tile files. The population data, input polygon (white), and exclusion polygon (red) are depicted in Fig. 8.

Test0036_i258_exclude_tech_tiles_ini. Duplicates Test0035 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0037_i258_exclude_tech_rasters. Duplicates Test0035 with population data as rasters.

Test0038_i258_exclude_tech_rasters_ini. Duplicates Test0036 with population data as rasters.

Test0039_i258_exclude_tech_fixed_tiles. Tests population counting in a complex polygon with a wholly contained exclusion polygon specified via the PopLib_exclusionFile environment variable. Population data are in tile files, and the exclusion file specifies a population of 1000 inside the exclusion polygon. The population data, input polygon, and exclusion polygon are the same as for Test0035.

Test0040_i258_exclude_tech_fixed_tiles_ini. Duplicates Test0039 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0041_i258_exclude_tech_fixed_rasters. Duplicates Test0039 with population data as rasters.

Test0042_i258_exclude_tech_fixed_rasters_ini. Duplicates Test0040 population data as rasters.

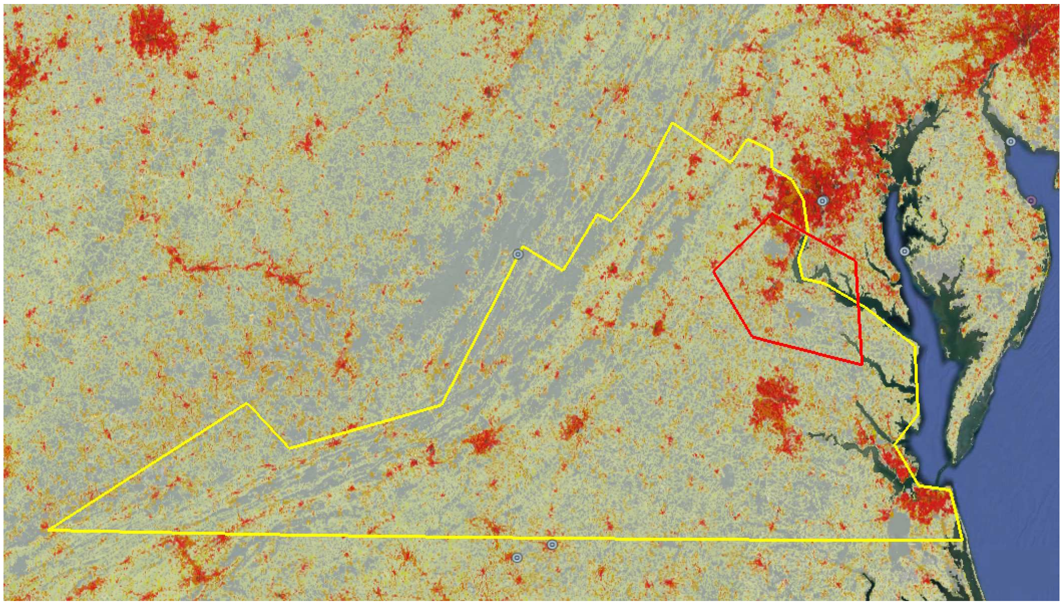

Fig. 6: Tests 0009–0012.

4.3 MGRS COORDINATES

Test0043_mgrs. Tests correct conversion between MGRS and geodetic coordinates.

Test0044_va_exclude_12_mgrs_12_tiles. This test is identical to Test0017 except the exclusion polygons are specified in MGRS coordinates. Population data are in tile files, and the exclusion file is specified via the PopLib_exclusionFile environment variable.

Test0045_va_exclude_12_mgrs_12_tiles_ini. Duplicates Test0044 with the exclusion polygon specified in the ini file instead of the environment variable.

Test0046_va_exclude_12_mgrs_12_rasters. Duplicates Test0044 with population data as rasters.

Test0047_va_exclude_12_mgrs_12_rasters_ini. Duplicates Test0045 with population data as rasters.

Test0048_va_exclude_12_mgrs_1_tiles. This test is identical to Test0044 except the first polygon in the exclusion file is specified in MGRS coordinates and the second polygon in geodetic coordinates. Population data are in tile files, and the exclusion file is specified via the PopLib_exclusionFile environment variable.

Test0049_va_exclude_12_mgrs_1_rasters. Duplicates Test0048 with population data as rasters.

Test0050_va_exclude_12_mgrs_2_tiles. This test is identical to Test0044 except the first polygon in the exclusion file is specified in geodetic coordinates and the second polygon in MGRS. Population data are in tile files, and the exclusion file is specified via the PopLib_exclusionFile

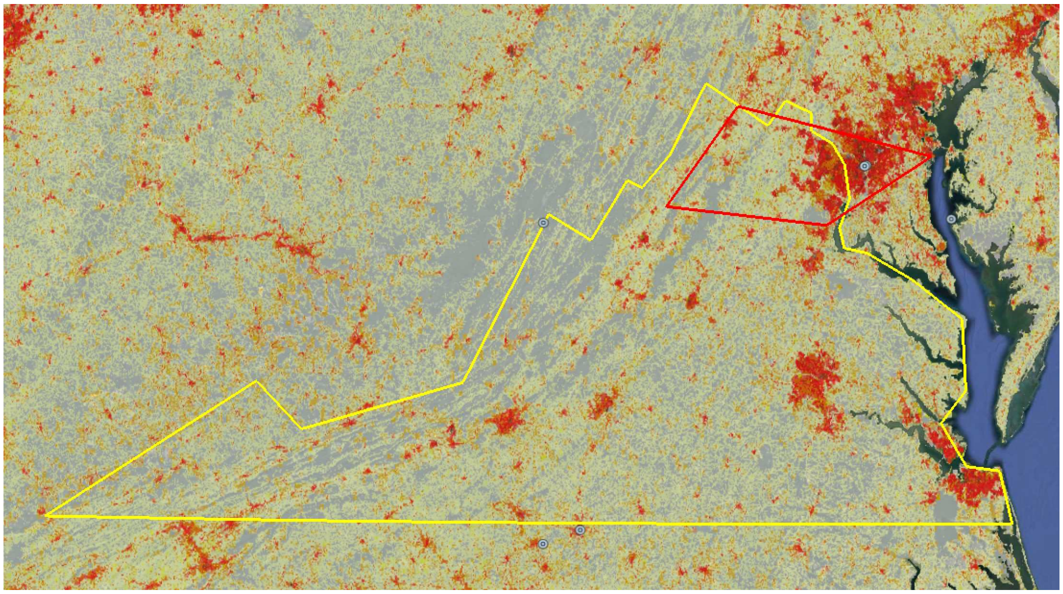

Fig. 7: Tests 0013–0016.

environment variable.

Test0051_va_exclude_12_mgrs_2_rasters. Duplicates Test0050 with population data as rasters.

4.4 HIGH RESOLUTION RASTER DATA

Test0052_mosul_raster_only_inner. Tests population counting with special high resolution raster data and the input polygon wholly contained within the raster. The population data and input polygon are depicted in Fig. 9.

Test0053_mosul_raster_only_cut. Tests population counting with special high resolution raster data, where the input polygon extends outside the raster but transects it. The population data and input polygon are depicted in Fig. 10.

Test0054_mosul_raster_only_outer. Tests population counting with special high resolution raster data, where raster is wholly contained within the input polygon. The population data and input polygon are depicted in Fig. 11.

Test0055_mosul_raster_levant_raster_inner. Tests population counting with special high resolution raster data, a background raster with lower resolution, and an input polygon wholly contained within the high resolution raster. The population data and input polygon are depicted in Fig. 12.

Test0056_mosul_raster_levant_raster_cut. Tests population counting with special high resolution raster data, a background raster with lower resolution, and an input polygon that

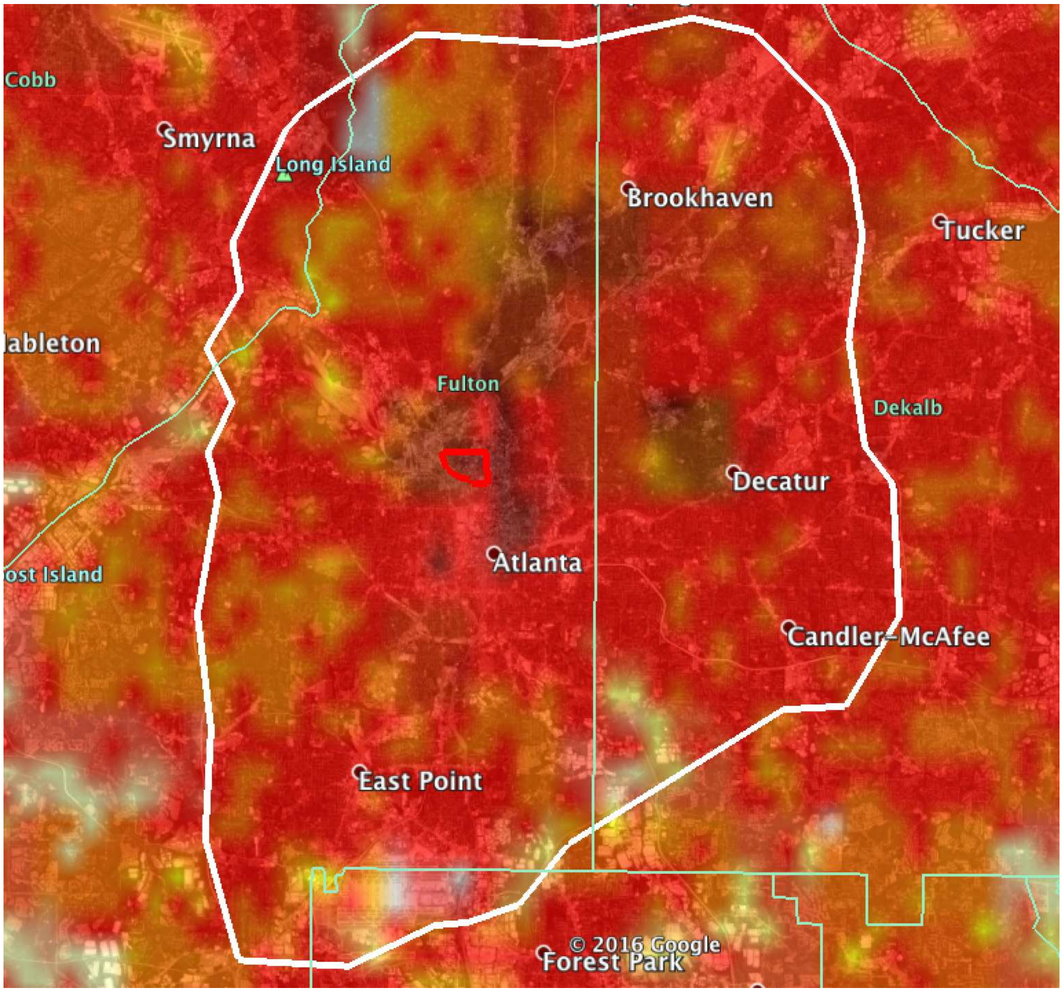

Fig. 8: Tests 0035–0038.

extends outside the high resolution raster but transects it. The population data and input polygon are depicted in Fig. 13.

Test0057_mosul_raster_levant_raster_outer. Tests population counting with special high resolution raster data, a background raster with lower resolution, and an input polygon that wholly contains the high resolution raster. The population data and input polygon are depicted in Fig. 14.

4.5 RASTERS AND TILES TOGETHER

Test0058_mosul_raster_levant_tile_inner. This test is identical to Test0055 except the background population data are in tiles.

Test0059_mosul_raster_levant_tile_cut. This test is identical to Test0056 except the background

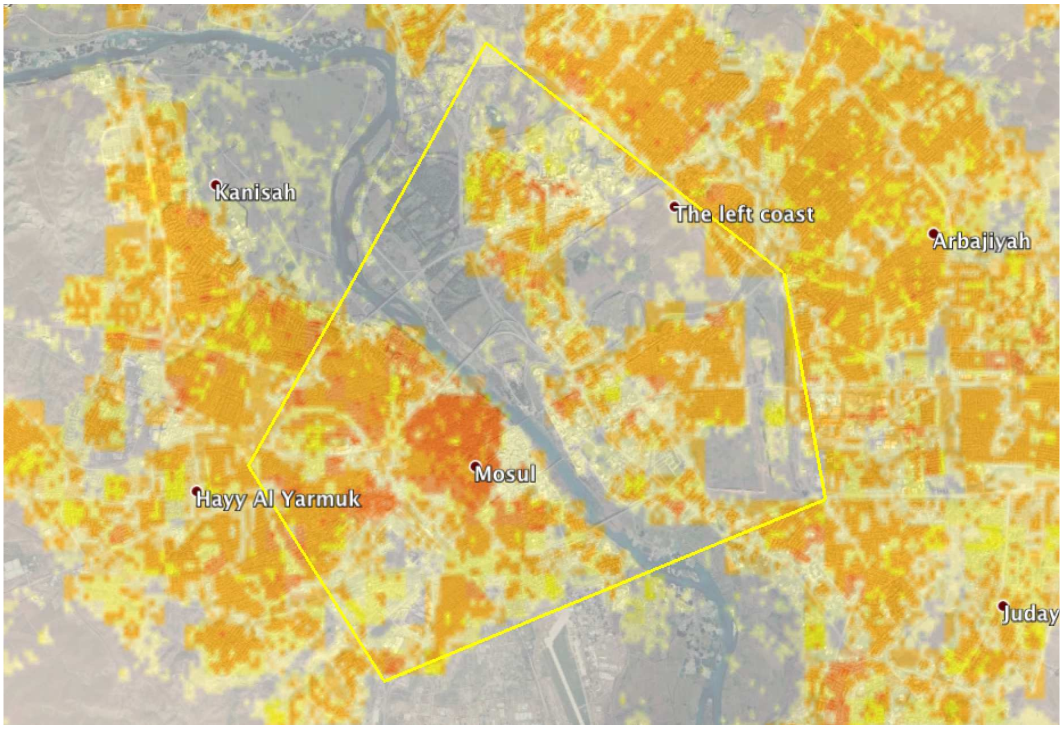

Fig. 9: Test 0052.

population data are in tiles.

Test0060_mosul_raster_levant_tile_outer. This test is identical to Test0057 except the background population data are in tiles.

Test0061_mosul_rasters_only. For comparison purposes, this test counts population in a directory containing raster files only. The population data and input polygon are identical to Test0057.

Test0062_mosul_rasters_and_tiles. Tests with the same input data and polygon as Test0061, but the background data are in tiles in the same directory as the high resolution raster.

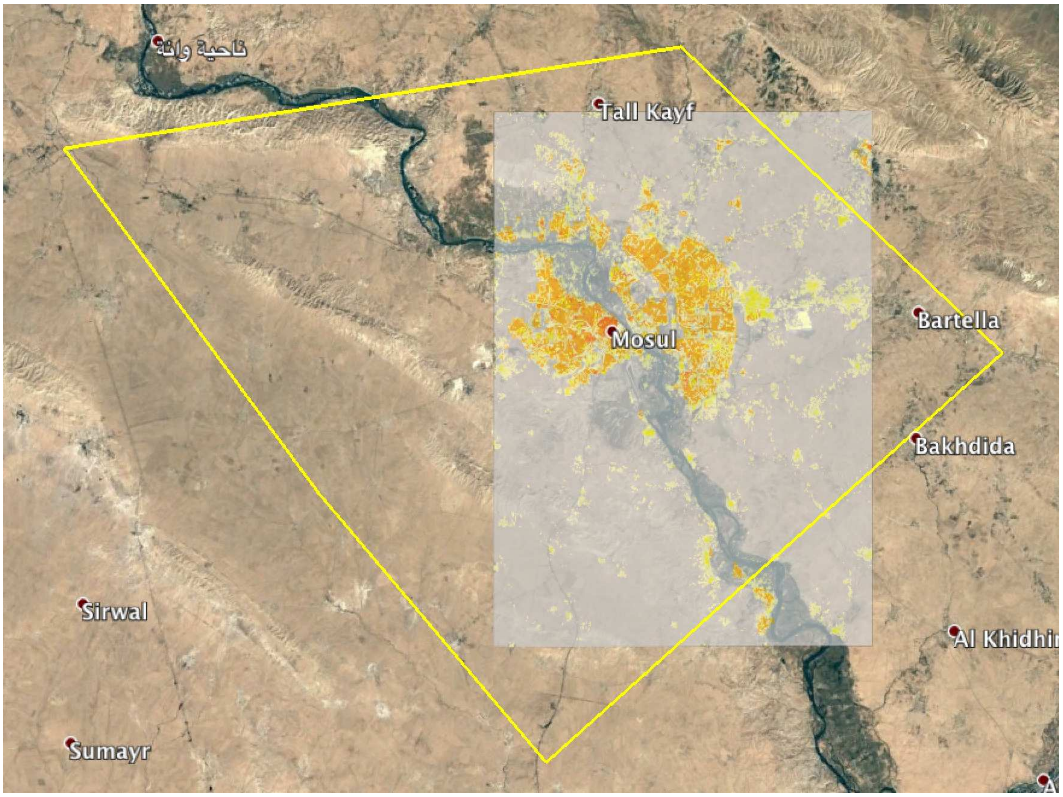

Fig. 10: Test 0053.

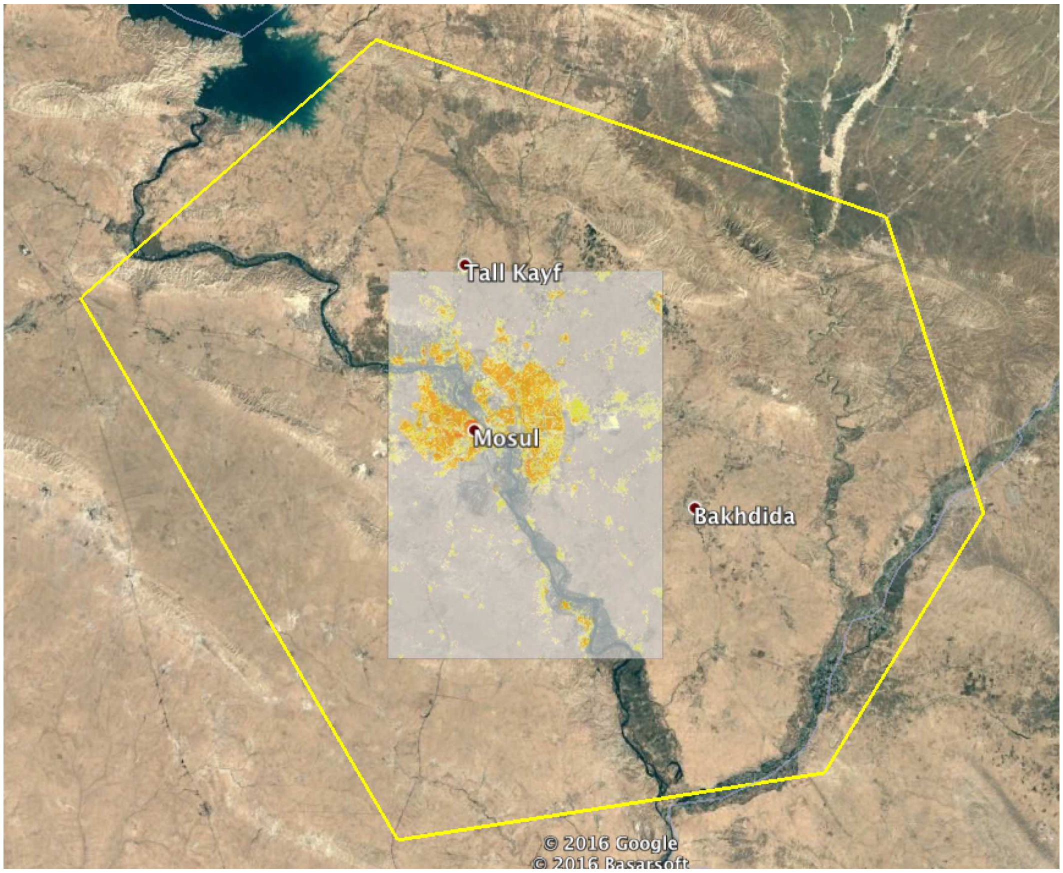

Fig. 11: Test 0054.

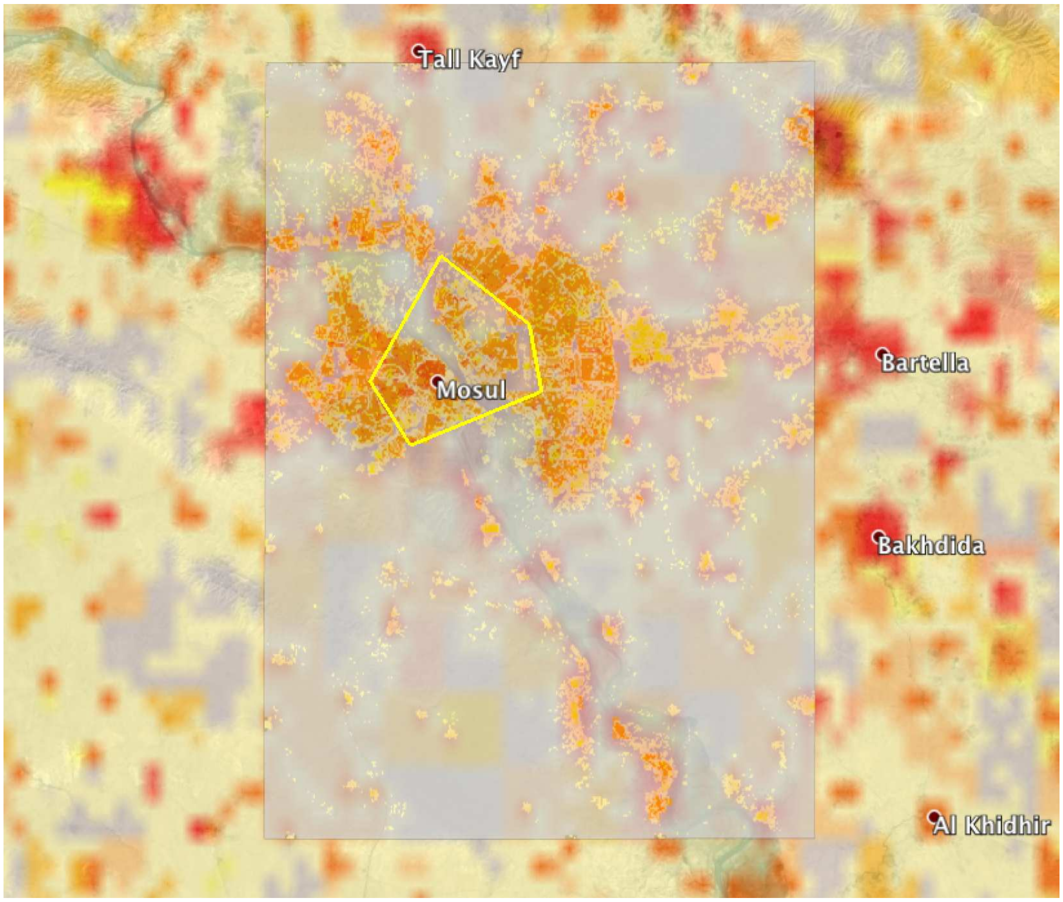

Fig. 12: Test 0055.

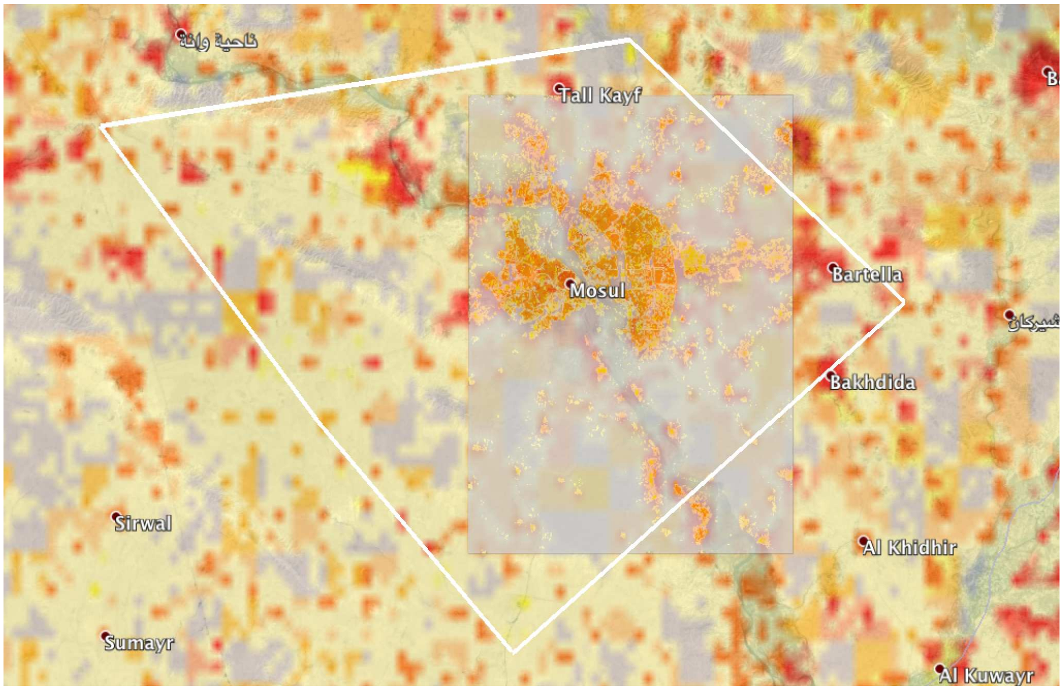

Fig. 13: Test 0056.

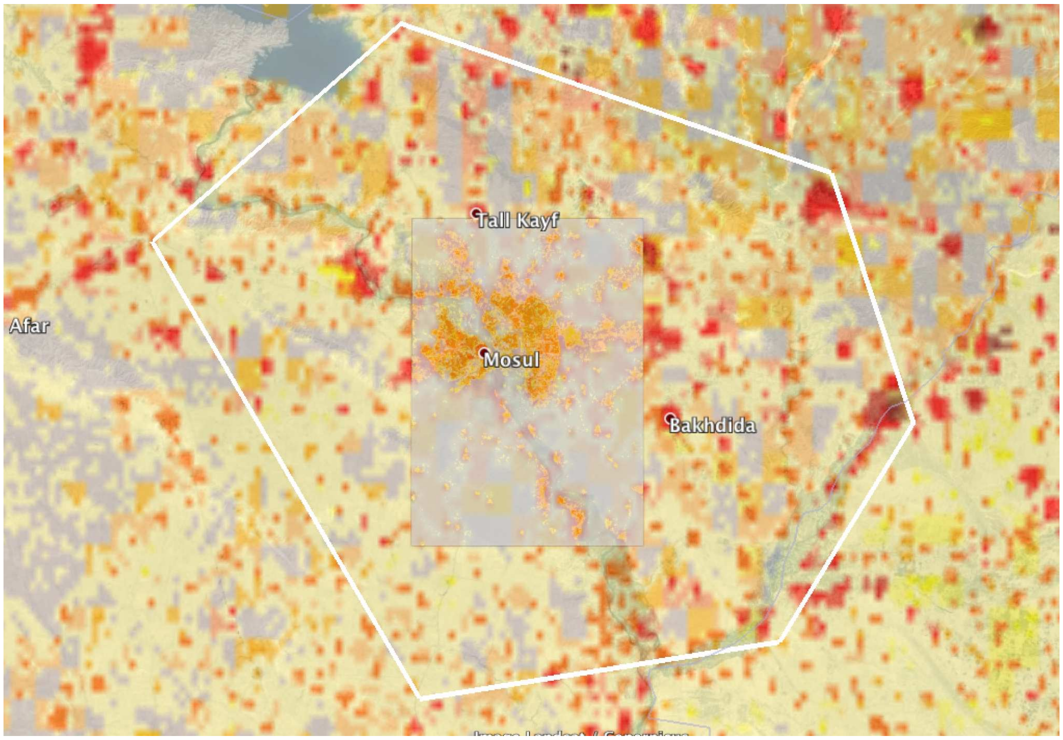

Fig. 14: Test 0057.

5. TEST RESULTS

CTest consolidates all test results into a single report, listed below.

```
Test project C:/Users/re7/src/Release
      Start  1: test0001_dc_around_tiles
 1/51 Test  #1: test0001_dc_around_tiles .......................   Passed    0.19 sec
      Start  2: test0002_dc_around_rasters
 2/51 Test  #2: test0002_dc_around_rasters .....................   Passed    0.20 sec
      Start  3: test0003_dc_inner_tiles
 3/51 Test  #3: test0003_dc_inner_tiles ........................   Passed    0.17 sec
      Start  4: test0004_dc_inner_rasters
 4/51 Test  #4: test0004_dc_inner_rasters ......................   Passed    0.16 sec
      Start  5: test0005_dc_diamond_tiles
 5/51 Test  #5: test0005_dc_diamond_tiles ......................   Passed    0.17 sec
      Start  6: test0006_dc_diamond_rasters
 6/51 Test  #6: test0006_dc_diamond_rasters ....................   Passed    0.17 sec
      Start  7: test0007_ny_triangle_tiles
 7/51 Test  #7: test0007_ny_triangle_tiles .....................   Passed    0.17 sec
      Start  8: test0008_ny_triangle_rasters
 8/51 Test  #8: test0008_ny_triangle_rasters ...................   Passed    0.17 sec
      Start  9: test0009_va_exclude_1_tiles
 9/51 Test  #9: test0009_va_exclude_1_tiles ....................   Passed    0.23 sec
      Start 10: test0010_va_exclude_1_tiles_ini
10/51 Test #10: test0010_va_exclude_1_tiles_ini ................   Passed    0.23 sec
      Start 11: test0011_va_exclude_1_rasters
11/51 Test #11: test0011_va_exclude_1_rasters ..................   Passed    0.30 sec
      Start 12: test0012_va_exclude_1_rasters_ini
12/51 Test #12: test0012_va_exclude_1_rasters_ini ..............   Passed    0.30 sec
      Start 13: test0013_va_exclude_2_tiles
13/51 Test #13: test0013_va_exclude_2_tiles ....................   Passed    0.27 sec
      Start 14: test0014_va_exclude_2_tiles_ini
14/51 Test #14: test0014_va_exclude_2_tiles_ini ................   Passed    0.22 sec
      Start 15: test0015_va_exclude_2_rasters
15/51 Test #15: test0015_va_exclude_2_rasters ..................   Passed    0.28 sec
      Start 16: test0016_va_exclude_2_rasters_ini
16/51 Test #16: test0016_va_exclude_2_rasters_ini ..............   Passed    0.27 sec
      Start 17: test0017_va_exclude_12_tiles
17/51 Test #17: test0017_va_exclude_12_tiles ...................   Passed    0.23 sec
      Start 18: test0018_va_exclude_12_tiles_ini
18/51 Test #18: test0018_va_exclude_12_tiles_ini ...............   Passed    0.23 sec
      Start 19: test0019_va_exclude_12_rasters
19/51 Test #19: test0019_va_exclude_12_rasters .................   Passed    0.30 sec
      Start 20: test0020_va_exclude_12_rasters_ini
20/51 Test #20: test0020_va_exclude_12_rasters_ini .............   Passed    0.28 sec
      Start 21: test0021_va_exclude_1n_tiles
21/51 Test #21: test0021_va_exclude_1n_tiles ...................   Passed    0.23 sec
      Start 22: test0022_va_exclude_1n_tiles_ini
22/51 Test #22: test0022_va_exclude_1n_tiles_ini ...............   Passed    0.23 sec
      Start 23: test0023_va_exclude_1n_rasters
23/51 Test #23: test0023_va_exclude_1n_rasters .................   Passed    0.30 sec
      Start 24: test0024_va_exclude_1n_rasters_ini
24/51 Test #24: test0024_va_exclude_1n_rasters_ini .............   Passed    0.30 sec
      Start 25: test0025_va_exclude_2n_tiles
25/51 Test #25: test0025_va_exclude_2n_tiles ...................   Passed    0.23 sec
      Start 26: test0026_va_exclude_2n_tiles_ini
26/51 Test #26: test0026_va_exclude_2n_tiles_ini ...............   Passed    0.22 sec
      Start 27: test0027_va_exclude_2n_rasters
27/51 Test #27: test0027_va_exclude_2n_rasters .................   Passed    0.27 sec
      Start 28: test0028_va_exclude_2n_rasters_ini
28/51 Test #28: test0028_va_exclude_2n_rasters_ini .............   Passed    0.28 sec
      Start 29: test0029_va_exclude_12n_tiles
29/51 Test #29: test0029_va_exclude_12n_tiles ..................   Passed    0.22 sec
      Start 30: test0030_va_exclude_12n_tiles_ini
```

```
30/51 Test #30: test0030_va_exclude_12n_tiles_ini ..............   Passed    0.23 sec
      Start 31: test0031_va_exclude_12n_rasters
31/51 Test #31: test0031_va_exclude_12n_rasters ................   Passed    0.27 sec
      Start 32: test0032_va_exclude_12n_rasters_ini
32/51 Test #32: test0032_va_exclude_12n_rasters_ini ............   Passed    0.28 sec
      Start 33: test0033_i285_tiles
33/51 Test #33: test0033_i285_tiles ............................   Passed    0.17 sec
      Start 34: test0034_i285_rasters
34/51 Test #34: test0034_i285_rasters ..........................   Passed    0.16 sec
      Start 35: test0035_i285_exclude_tech_tiles
35/51 Test #35: test0035_i285_exclude_tech_tiles ...............   Passed    0.19 sec
      Start 36: test0036_i285_exclude_tech_tiles_ini
36/51 Test #36: test0036_i285_exclude_tech_tiles_ini ...........   Passed    0.17 sec
      Start 37: test0037_i285_exclude_tech_rasters
37/51 Test #37: test0037_i285_exclude_tech_rasters .............   Passed    0.17 sec
      Start 38: test0038_i285_exclude_tech_rasters_ini
38/51 Test #38: test0038_i285_exclude_tech_rasters_ini .........   Passed    0.16 sec
      Start 39: test0039_i285_exclude_tech_fixed_tiles
39/51 Test #39: test0039_i285_exclude_tech_fixed_tiles .........   Passed    0.19 sec
      Start 40: test0040_i285_exclude_tech_fixed_tiles_ini
40/51 Test #40: test0040_i285_exclude_tech_fixed_tiles_ini .....   Passed    0.17 sec
      Start 41: test0041_i285_exclude_tech_fixed_rasters
41/51 Test #41: test0041_i285_exclude_tech_fixed_rasters .......   Passed    0.17 sec
      Start 42: test0042_i285_exclude_tech_fixed_rasters_ini
42/51 Test #42: test0042_i285_exclude_tech_fixed_rasters_ini ...   Passed    0.17 sec
      Start 43: test0043_mgrs
43/51 Test #43: test0043_mgrs ..................................   Passed    0.02 sec
      Start 44: test0044_va_exclude_12_mgrs_12_tiles
44/51 Test #44: test0044_va_exclude_12_mgrs_12_tiles ...........   Passed    0.22 sec
      Start 45: test0045_va_exclude_12_mgrs_12_tiles_ini
45/51 Test #45: test0045_va_exclude_12_mgrs_12_tiles_ini .......   Passed    0.23 sec
      Start 46: test0046_va_exclude_12_mgrs_12_rasters
46/51 Test #46: test0046_va_exclude_12_mgrs_12_rasters .........   Passed    0.27 sec
      Start 47: test0047_va_exclude_12_mgrs_12_rasters_ini
47/51 Test #47: test0047_va_exclude_12_mgrs_12_rasters_ini .....   Passed    0.28 sec
      Start 48: test0048_va_exclude_12_mgrs_1_tiles
48/51 Test #48: test0048_va_exclude_12_mgrs_1_tiles ............   Passed    0.23 sec
      Start 49: test0049_va_exclude_12_mgrs_1_rasters
49/51 Test #49: test0049_va_exclude_12_mgrs_1_rasters ..........   Passed    0.28 sec
      Start 50: test0050_va_exclude_12_mgrs_2_tiles
50/51 Test #50: test0050_va_exclude_12_mgrs_2_tiles ............   Passed    0.23 sec
      Start 51: test0051_va_exclude_12_mgrs_2_rasters
51/51 Test #51: test0051_va_exclude_12_mgrs_2_rasters ..........   Passed    0.27 sec

100% tests passed, 0 tests failed out of 51

Label Time Summary:
exclusion    =   9.61 sec (40 tests)
mgrs         =   0.02 sec (1 test)
rasters      =   6.03 sec (25 tests)
simple       =   1.73 sec (10 tests)
tiles        =   5.31 sec (25 tests)

Total Test time (real) =  11.44 sec
```

References

1. Kitware. *CMake*. https://cmake.org.

A POPLIB TEST DRIVER

```
#!/usr/bin/env python
#------------------------------------------------------------------------
#       NAME:           poplib_test_driver.py                           -
#       HISTORY:                                                        -
#               2016-05-26      leerw@ornl.gov                          -
#               2016-04-22      leerw@ornl.gov                          -
#               2016-04-21      leerw@ornl.gov                          -
#------------------------------------------------------------------------
import argparse, os, subprocess, sys, tempfile, traceback


#------------------------------------------------------------------------
#       CLASS:          PopLibTestDriver                                -
#------------------------------------------------------------------------
class PopLibTestDriver( object ):


#               -- Object Methods
#               --


  #----------------------------------------------------------------------
  #     METHOD:         __init__()                                      -
  #----------------------------------------------------------------------
  def __init__(
      self, test_name, poplib_run, test_dir,
      pop_dirs, poly_fname, excl_fname = None, excl_ini_flag = False
      ):
    """
@param  test_name       name of test
@param  poplib_run      path to poplib_run executable
@param  test_dir        test base directory containing baseline/, polys/
                        rasters/ and tiles/ subdirs
@param  pop_dirs        test_dir-relative paths to population directories to use
@param  excl_fname      optional exclusion file in polys/ subdir
@param  excl_ini_flag   if True, exclusion file is specified in the INI
                        file instead of via an env variable
"""
    self.fTestName = test_name
    self.fExe = poplib_run
    self.fTestDir = test_dir
    self.fPopDirs = \
        [ os.path.join( test_dir, p ) for p in pop_dirs.split( ’,’ ) ]
    self.fPolyPath = os.path.join( test_dir, ’polys’, poly_fname )

    self.fBaselinePaths = [
        os.path.join( test_dir, ’baselines’, ’%s.%d.output’ % ( test_name, i ) )
        for i in xrange( 4 )
        ]

    self.fExclPath = \
        os.path.join( test_dir, ’polys’, excl_fname )  \
        if excl_fname else  None
    self.fExclIniFlag = excl_ini_flag
  #end __init__


  #----------------------------------------------------------------------
  #     METHOD:         compareOutputs()                                -
  #----------------------------------------------------------------------
  def compareOutputs( self, expected, actual ):
    msg = ’passed’
    if len( expected ) != len( actual ):
      msg = ’output counts differ: expected=%d, actual=%d’ % \
          ( len( expected ), len( actual ) )

    else:
      for i in range( len( expected ) ):
        ex_item = expected[ i ]
        ac_item = actual[ i ]

        if len( ex_item ) != len( ac_item ):
          msg = ’output %d lengths differ: expected=%d, actual=%d’ % \
              ( i, len( ex_item ), len( ac_item ) )
          break

        else:
          for j in range( len( ex_item ) ):
            ex = ex_item[ j ]
            ac = ac_item[ j ]
            pop_diff = (ac[ 2 ] - ex[ 2 ]) / ((ac[ 2 ] + ex[ 2 ]) / 2.0)
            if ex[ 0 ] != ac[ 0 ] or \
                ex[ 1 ] != ac[ 1 ] or \
                abs( pop_diff ) >= 1.0e-5:
              msg = ’output %d/%d mismatch: expected=%d,%d,%.6g, actual=%d,%d,%.6g’ % \
                  ( i, j, ex[ 0 ], ex[ 1 ], ex[ 2 ], ac[ 0 ], ac[ 1 ], ac[ 2 ] )
              break
        #end if-else
      #end for i
    #end if-else

    return  msg
  #end compareOutputs


  #----------------------------------------------------------------------
  #     METHOD:         readOutput()                                    -
  #----------------------------------------------------------------------
  def readOutput( self, fparam, tee_path = None ):
    results = []

    if isinstance( fparam, file ):
      fp = fparam
    else:
      fp = file( fparam )

    tee_fp = file( tee_path, ’w’ ) if tee_path else None

    try:
      cur_list = None
      line = fp.readline()
      while line:
        line = line.rstrip( ’\r\n’ )
        if tee_fp:
          print >> sys.stderr, line
          print >> tee_fp, line

        if line.startswith( ’country’ ):
          cur_list = []

        elif line.startswith( ’total=’ ):
          ndx = line.find( ’=’ )
          total = float( line[ ndx + 1 : ] ) if ndx >= 0 else -1.0
          cur_list.append( [ -1, -1, total ] )
          results.append( cur_list )
          cur_list = None
        elif cur_list is not None:
          line = line.rstrip( ’\r\n’ )
          tokens = []
          ndx = line.find( ’:’ )
          if ndx > 0:
            tokens = line[ ndx + 1 : ].split( ’,’ )
          if len( tokens ) >= 3:
            cur_list.append(
                [ int( tokens[ 0 ] ), int( tokens[ 1 ] ), float( tokens[ 2 ] ) ]
                )
        #end elif

        line = fp.readline()
      #end while

      return  results

    finally:
      if tee_fp:
        tee_fp.close()
      if not isinstance( fparam, file ):
        fp.close()
  #end readOutput


  #----------------------------------------------------------------------
  #     METHOD:         run()                                           -
  #----------------------------------------------------------------------
  def run( self ):
    """
@return                 result string
"""
    result = ’baseline’
    mode_results = [ ’baseline’ for i in xrange( 4 ) ]

#               -- Create Logs Subdir if Necessary
#               --
    logs_dir = os.path.join( self.fTestDir, ’logs’ )
    if not os.path.exists( logs_dir ):
      os.mkdir( logs_dir )
      #os.makedirs( logs_dir )

#               -- Create INI File
#               --
    ini_fd, ini_path = tempfile.mkstemp( ’.ini’ )
    ini_fp = os.fdopen( ini_fd, ’w’ )
    ini_fp = file( ini_path, ’w’ )

    try:
#               -- Create INI File
#               --
      #print >> ini_fp, ’[Paths]%sPopDir=%s’ % ( os.linesep, self.fPopDir )
      print >> ini_fp, ’[Paths]%sPopDir=%s’ % ( os.linesep, ’,’.join( self.fPopDirs ) )
      if self.fExclPath is not None and self.fExclIniFlag:
        print >> ini_fp, ’PopExclusionFile=%s’ % self.fExclPath
      ini_fp.close()

#               -- Environment
#               --
      use_env = None
      if not (self.fExclPath is None or self.fExclIniFlag):
        use_env = dict( os.environ )
        use_env[ ’PopLib_exclusionFile’ ] = self.fExclPath
```

```
#               -- Execute
#               --
      for m in xrange( 4 ):
        run_args = [ self.fExe, ’-i’, ini_path, ’-m’, str( m ), self.fPolyPath  ]
        print >> sys.stderr, ’[poplib_test_driver]’
        print >> sys.stderr, ’,’.join( run_args )
        if use_env and ’PopLib_exclusionFile’ in use_env:
          print >> sys.stderr, \
              ’PopLib_exclusionFile=%s’ % use_env[ ’PopLib_exclusionFile’ ]
        proc = subprocess.Popen(
            run_args,
            env = use_env,
            stdout = subprocess.PIPE
#           stderr = subprocess.STDOUT
            )
        test_results = self.readOutput(
            proc.stdout,
            os.path.join( logs_dir, self.fTestName + ’.log’ )
            )
        proc.wait()

#                       -- Read Baseline
#                       --
        if os.path.exists( self.fBaselinePaths[ m ] ):
          base_results = self.readOutput( self.fBaselinePaths[ m ] )
          mode_results[ m ] = self.compareOutputs( base_results, test_results )

        else:
          base_fp = file( self.fBaselinePaths[ m ], ’w’ )
          self.writeOutput( base_fp, test_results )
          base_fp.close()
        #end if os.path.exists
      #end for m

      if mode_results == [ ’passed’ ] * 4:
        result = ’all passed’
      elif mode_results == [ ’baseline’ ] * 4:
        result = ’all baseline’
      else:
        pieces = [ ’%d: %s’ % ( i, mode_results[ i ] ) for i in xrange( 4 ) ]
        result = ’,’.join( pieces )

      return  result

    finally:
      os.remove( ini_path )
      pass
  #end run


  #----------------------------------------------------------------------
  #     METHOD:         writeOutput()                                   -
  #----------------------------------------------------------------------
  def writeOutput( self, fp, results ):
    for cur_list in results:
      print >> fp, ’country,posture,population’
      for i in range( len( cur_list ) - 1 ):
        tpl = cur_list[ i ]
        print >> fp, ’%d:%d,%d,%.6g’ % \
          ( i, tpl[ 0 ], tpl[ 1 ], tpl[ 2 ] )
      print >> fp, ’total=%.6g%s’ % ( cur_list[ -1 ][ 2 ], os.linesep )
    #end for
  #end writeOutput
```

```
#               -- Static Methods
#               --


  #----------------------------------------------------------------------
  #     METHOD:         main()                                          -
  #----------------------------------------------------------------------
  @staticmethod
  def main():
    try:
      parser = argparse.ArgumentParser()

      parser.add_argument( ’-n’, ’--name’, help = ’test name’ )
      parser.add_argument(
          ’-e’, ’--exe’,
          default = ’poplib_run’,
          help = ’poplib_run executable’
          )
      parser.add_argument(
          ’-d’, ’--test-dir’,
          help = ’test root directory’
          )
      parser.add_argument(
          ’-p’, ’--pop-dirs’,
          help = ’population data directories, comma-delimited’
          )
      parser.add_argument(
          ’--excl’,
          help = ’optional exclusion polygon file’
          )
      parser.add_argument(
          ’--excl-ini’,
          action = ’store_true’,
          help = ’specify exclusion file in INI file instead of environment’
          )
      parser.add_argument(
          ’-i’, ’--input’,
          help = ’polygon input file’
          )

      args = parser.parse_args()

      if not( args.name or args.pop_dir or args.input ):
        parser.print_help()

      else:
        test_dir = args.test_dir if args.test_dir else os.getcwd()
        driver = PopLibTestDriver(
            args.name, args.exe, test_dir,
            args.pop_dirs, args.input, args.excl, args.excl_ini
            )
        result = driver.run()
        print ’result: ’ + result

    except Exception, ex:
      et, ev, tb = sys.exc_info()
      while tb:
        print >> sys.stderr, \
            ’%s\nFile=%s\nLine=%s’ % \
            ( str( ex ), str( tb.tb_frame.f_code ),
              str( traceback.tb_lineno( tb ) ) )
        tb = tb.tb_next
  #end main

#end PopLibTestDriver
```

```
if __name__ == ’__main__’:
  PopLibTestDriver.main()
```

B CMAKE LIST FILE

```
#------------------------------------------------------------------------
#       NAME:           hpac5-poplib-tests::CMakeLists.txt              -
#       HISTORY:                                                        -
#               2016-06-13      leerw@ornl.gov                          -
#         Added mgrs_test.
#               2016-04-25      leerw@ornl.gov                          -
#               2016-04-22      leerw@ornl.gov                          -
#------------------------------------------------------------------------
cmake_minimum_required( VERSION 2.8 )
#project( hpac5-poplib-tests )


#       -- Find python
#       --
if ( NOT DEFINED PYTHON_EXECUTABLE )
  find_program( PYTHON_EXECUTABLE NAMES python )
endif()


#       -- Set POPLIB_BIN_DIR to directory containing poplib.dll and
#       -- poplib_run, default to parent bin
get_filename_component( BIN_DIR ${CMAKE_CURRENT_BINARY_DIR} DIRECTORY )
set(
    POPLIB_BIN_DIR ${BIN_DIR}
    CACHE PATH
    "Location of poplib shared object and poplib_run executable"
    )


#       -- Build mgrs_test
#       --

if ( ${CMAKE_CXX_COMPILER_ID} STREQUAL "Clang" )
  if ( CMAKE_CXX_COMPILER_VERSION VERSION_LESS 2.9.0 )
    add_definitions( -DUSE_OLD_REGEX )
  endif()

  if ( "${CMAKE_SYSTEM_NAME}" STREQUAL "Linux" )
    set( CMAKE_C_FLAGS "${CMAKE_C_FLAGS} -fPIC"  CACHE STRING "" FORCE )
    set( CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fPIC -std=c++0x" CACHE STRING "" FORCE )
  else()
    set( CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++0x" CACHE STRING "" FORCE )
  endif()

elseif ( ${CMAKE_CXX_COMPILER_ID} STREQUAL "GNU" )
  if ( CMAKE_CXX_COMPILER_VERSION VERSION_LESS 4.8.1 )
    add_definitions( -DUSE_OLD_REGEX )
  endif()

  if ( "${CMAKE_SYSTEM_NAME}" STREQUAL "Linux" )
    set( CMAKE_C_FLAGS "${CMAKE_C_FLAGS} -fPIC"  CACHE STRING "" FORCE )
    set( CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fPIC" CACHE STRING "" FORCE )
  endif()

  if ( NOT CMAKE_CXX_COMPILER_VERSION VERSION_LESS 4.3.0 )
    set( CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++0x" CACHE STRING "" FORCE )
  endif()
endif()

if ( ${MSVC} )
  add_definitions( -D_CRT_SECURE_NO_WARNINGS )
endif()
```

```
include_directories(
    ../gpc
    ../zlib-1.2.8
    ../LandScan
    ../geotrans_mgrs-3.5
    ../geotrans_mgrs-3.5/CoordinateSystem/mgrs
    ../geotrans_mgrs-3.5/CoordinateSystem/misc
    ../geotrans_mgrs-3.5/CoordinateSystem/polarst
    ../geotrans_mgrs-3.5/CoordinateSystem/tranmerc
    ../geotrans_mgrs-3.5/CoordinateSystem/ups
    ../geotrans_mgrs-3.5/CoordinateSystem/utm
    ../geotrans_mgrs-3.5/CoordinateSystemParameters
    ../geotrans_mgrs-3.5/CoordinateTuples
    ../geotrans_mgrs-3.5/Enumerations
    ../geotrans_mgrs-3.5/Exception
    )


add_executable( mgrs_test mgrs_test.cpp )
target_link_libraries( mgrs_test poplib )



#------------------------------------------------------------------------
#       MACRO:          add_exclusion_test()                            -
#------------------------------------------------------------------------
macro( add_exclusion_test name labels pop_dir poly_fname excl_fname excl_mode tout )
  add_test(
      NAME ${name}
      WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR}
      COMMAND ${PYTHON_EXECUTABLE} poplib_test_driver.py
          -n ${name}
          -e ${POPLIB_BIN_DIR}/poplib_run
          -d ${CMAKE_CURRENT_SOURCE_DIR}
          -p ${pop_dir}
          -i ${poly_fname}
          --excl ${excl_fname} ${excl_mode}
      )

  if ( ${CMAKE_SYSTEM_NAME} STREQUAL "Darwin" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "DYLD_LIBRARY_PATH=${POPLIB_BIN_DIR}:$ENV{DYLD_LIBRARY_PATH}"
        LABELS "exclusion;${labels}"
        PASS_REGULAR_EXPRESSION "all passed"
        TIMEOUT ${tout}
        )
  elseif ( ${CMAKE_SYSTEM_NAME} STREQUAL "Windows" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "PATH=${POPLIB_BIN_DIR}\\;$ENV{PATH}"
        LABELS "exclusion;${labels}"
        PASS_REGULAR_EXPRESSION "all passed"
        TIMEOUT ${tout}
        )
  else() #elseif ( ${CMAKE_SYSTEM_NAME} STREQUAL "Linux" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "LD_LIBRARY_PATH=${POPLIB_BIN_DIR}:$ENV{LD_LIBRARY_PATH}"
        LABELS "exclusion;${labels}"
        PASS_REGULAR_EXPRESSION "all passed"
        TIMEOUT ${tout}
        )
  endif()
```

```
endmacro( add_exclusion_test )


#------------------------------------------------------------------------
#       MACRO:          add_mgrs_test()                                 -
#------------------------------------------------------------------------
macro( add_mgrs_test name input_fname tout )
  add_test(
      NAME ${name}
      WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR}
      COMMAND mgrs_test polys/${input_fname}
      )

  if ( ${CMAKE_SYSTEM_NAME} STREQUAL "Darwin" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "DYLD_LIBRARY_PATH=${POPLIB_BIN_DIR}:$ENV{DYLD_LIBRARY_PATH}"
        LABELS "mgrs"
        PASS_REGULAR_EXPRESSION "passed"
        TIMEOUT ${tout}
        )
  elseif ( ${CMAKE_SYSTEM_NAME} STREQUAL "Windows" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "PATH=${POPLIB_BIN_DIR}\\;$ENV{PATH}"
        LABELS "mgrs"
        PASS_REGULAR_EXPRESSION "passed"
        TIMEOUT ${tout}
        )
  else() #elseif ( ${CMAKE_SYSTEM_NAME} STREQUAL "Linux" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "LD_LIBRARY_PATH=${POPLIB_BIN_DIR}:$ENV{LD_LIBRARY_PATH}"
        LABELS "mgrs"
        PASS_REGULAR_EXPRESSION "passed"
        TIMEOUT ${tout}
        )
  endif()
endmacro( add_mgrs_test )


#------------------------------------------------------------------------
#       MACRO:          add_simple_test()                               -
#------------------------------------------------------------------------
macro( add_simple_test name labels pop_dirs poly_fname tout )
  add_test(
      NAME ${name}
      WORKING_DIRECTORY ${CMAKE_CURRENT_SOURCE_DIR}
      COMMAND ${PYTHON_EXECUTABLE} poplib_test_driver.py
          -n ${name}
          -e ${POPLIB_BIN_DIR}/poplib_run
          -d ${CMAKE_CURRENT_SOURCE_DIR}
          -p ${pop_dirs}
          -i ${poly_fname}
      )

  if ( ${CMAKE_SYSTEM_NAME} STREQUAL "Darwin" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "DYLD_LIBRARY_PATH=${POPLIB_BIN_DIR}:$ENV{DYLD_LIBRARY_PATH}"
        LABELS "simple;${labels}"
        PASS_REGULAR_EXPRESSION "all passed"
        TIMEOUT ${tout}
        )
  elseif ( ${CMAKE_SYSTEM_NAME} STREQUAL "Windows" )
```

```
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "PATH=${POPLIB_BIN_DIR}\\;$ENV{PATH}"
        LABELS "simple;${labels}"
        PASS_REGULAR_EXPRESSION "all passed"
        TIMEOUT ${tout}
        )
  else() #elseif ( ${CMAKE_SYSTEM_NAME} STREQUAL "Linux" )
    set_tests_properties(
        ${name} PROPERTIES
        ENVIRONMENT "LD_LIBRARY_PATH=${POPLIB_BIN_DIR}:$ENV{LD_LIBRARY_PATH}"
        LABELS "simple;${labels}"
        PASS_REGULAR_EXPRESSION "all passed"
        TIMEOUT ${tout}
        )
  endif()
endmacro( add_simple_test )


#       -- Define tests
#       --
add_simple_test( test0001_dc_around_tiles "tiles" tiles/ne-usa dc-around.in 10 )
add_simple_test( test0002_dc_around_rasters "rasters" rasters/ne-usa dc-around.in 10 )
add_simple_test( test0003_dc_inner_tiles "tiles" tiles/ne-usa dc-inner.in 10 )
add_simple_test( test0004_dc_inner_rasters "rasters" rasters/ne-usa dc-inner.in 10 )
add_simple_test( test0005_dc_diamond_tiles "tiles" tiles/ne-usa dc.in 10 )
add_simple_test( test0006_dc_diamond_rasters "rasters" rasters/ne-usa dc.in 10 )
add_simple_test( test0007_ny_triangle_tiles "tiles" tiles/ne-usa ny.in 10 )
add_simple_test( test0008_ny_triangle_rasters "rasters" rasters/ne-usa ny.in 10 )

add_exclusion_test(
    test0009_va_exclude_1_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-1.in "" 10
    )
add_exclusion_test(
    test0010_va_exclude_1_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-1.in "--excl-ini" 10
    )
add_exclusion_test(
    test0011_va_exclude_1_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-1.in "" 10
    )
add_exclusion_test(
    test0012_va_exclude_1_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-1.in "--excl-ini" 10
    )
add_exclusion_test(
    test0013_va_exclude_2_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-2.in "" 10
    )
add_exclusion_test(
    test0014_va_exclude_2_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-2.in "--excl-ini" 10
    )
add_exclusion_test(
    test0015_va_exclude_2_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-2.in "" 10
    )
add_exclusion_test(
    test0016_va_exclude_2_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-2.in "--excl-ini" 10
    )
add_exclusion_test(
    test0017_va_exclude_12_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-12.in "" 10
    )
add_exclusion_test(
    test0018_va_exclude_12_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-12.in "--excl-ini" 10
    )
add_exclusion_test(
    test0019_va_exclude_12_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-12.in "" 10
    )
add_exclusion_test(
    test0020_va_exclude_12_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-12.in "--excl-ini" 10
    )

add_exclusion_test(
    test0021_va_exclude_1n_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-1n.in "" 10
    )
add_exclusion_test(
    test0022_va_exclude_1n_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-1n.in "--excl-ini" 10
    )
add_exclusion_test(
    test0023_va_exclude_1n_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-1n.in "" 10
    )
add_exclusion_test(
    test0024_va_exclude_1n_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-1n.in "--excl-ini" 10
    )
add_exclusion_test(
    test0025_va_exclude_2n_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-2n.in "" 10
    )
add_exclusion_test(
    test0026_va_exclude_2n_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-2n.in "--excl-ini" 10
    )
add_exclusion_test(
    test0027_va_exclude_2n_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-2n.in "" 10
    )
add_exclusion_test(
    test0028_va_exclude_2n_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-2n.in "--excl-ini" 10
    )
add_exclusion_test(
    test0029_va_exclude_12n_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-12n.in "" 10
    )
add_exclusion_test(
    test0030_va_exclude_12n_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-12n.in "--excl-ini" 10
    )
add_exclusion_test(
    test0031_va_exclude_12n_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-12n.in "" 10
    )
add_exclusion_test(
    test0032_va_exclude_12n_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-12n.in "--excl-ini" 10
    )

add_simple_test( test0033_i285_tiles "tiles" tiles/ne-usa i-285.in 10 )
add_simple_test( test0034_i285_rasters "rasters" rasters/ne-usa i-285.in 10 )
```

```
add_exclusion_test(
    test0035_i285_exclude_tech_tiles "tiles"
    tiles/ne-usa i-285.in gatech-campus.in "" 10
    )
add_exclusion_test(
    test0036_i285_exclude_tech_tiles_ini "tiles"
    tiles/ne-usa i-285.in gatech-campus.in "--excl-ini" 10
    )
add_exclusion_test(
    test0037_i285_exclude_tech_rasters "rasters"
    rasters/ne-usa i-285.in gatech-campus.in "" 10
    )
add_exclusion_test(
    test0038_i285_exclude_tech_rasters_ini "rasters"
    rasters/ne-usa i-285.in gatech-campus.in "--excl-ini" 10
    )
add_exclusion_test(
    test0039_i285_exclude_tech_fixed_tiles "tiles"
    tiles/ne-usa i-285.in gatech-campus-1k.in "" 10
    )
add_exclusion_test(
    test0040_i285_exclude_tech_fixed_tiles_ini "tiles"
    tiles/ne-usa i-285.in gatech-campus-1k.in "--excl-ini" 10
    )
add_exclusion_test(
    test0041_i285_exclude_tech_fixed_rasters "rasters"
    rasters/ne-usa i-285.in gatech-campus-1k.in "" 10
    )
add_exclusion_test(
    test0042_i285_exclude_tech_fixed_rasters_ini "rasters"
    rasters/ne-usa i-285.in gatech-campus-1k.in "--excl-ini" 10
    )

add_mgrs_test( test0043_mgrs mgrs.data 5 )

add_exclusion_test(
    test0044_va_exclude_12_mgrs_12_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-12-mgrs-12.in "" 10
    )
add_exclusion_test(
    test0045_va_exclude_12_mgrs_12_tiles_ini "tiles"
    tiles/ne-usa virginia.in excl-nova-12-mgrs-12.in "--excl-ini" 10
    )
add_exclusion_test(
    test0046_va_exclude_12_mgrs_12_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-12-mgrs-12.in "" 10
    )
add_exclusion_test(
    test0047_va_exclude_12_mgrs_12_rasters_ini "rasters"
    rasters/ne-usa virginia.in excl-nova-12-mgrs-12.in "--excl-ini" 10
    )

add_exclusion_test(
    test0048_va_exclude_12_mgrs_1_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-12-mgrs-1.in "" 10
    )
add_exclusion_test(
    test0049_va_exclude_12_mgrs_1_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-12-mgrs-1.in "" 10
    )
add_exclusion_test(
    test0050_va_exclude_12_mgrs_2_tiles "tiles"
    tiles/ne-usa virginia.in excl-nova-12-mgrs-2.in "" 10
    )
add_exclusion_test(
    test0051_va_exclude_12_mgrs_2_rasters "rasters"
    rasters/ne-usa virginia.in excl-nova-12-mgrs-2.in "" 10
    )


add_simple_test(
    test0052_mosul_raster_only_inner "rasters" rasters/mosul
    mosul-inner.in 10
    )
add_simple_test(
    test0053_mosul_raster_only_cut "rasters" rasters/mosul
    mosul-cut.in 10
    )
add_simple_test(
    test0054_mosul_raster_only_outer "rasters" rasters/mosul
    mosul-outer.in 10
    )

add_simple_test(
    test0055_mosul_raster_levant_raster_inner "rasters"
    "rasters/mosul,rasters/levant"
    mosul-inner.in 10
    )
add_simple_test(
    test0056_mosul_raster_levant_raster_cut "rasters"
    "rasters/mosul,rasters/levant"
    mosul-cut.in 10
    )
add_simple_test(
    test0057_mosul_raster_levant_raster_outer "rasters"
    "rasters/mosul,rasters/levant"
    mosul-outer.in 10
    )

add_simple_test(
    test0058_mosul_raster_levant_tile_inner "rasters"
    "rasters/mosul,tiles/levant"
    mosul-inner.in 10
    )
add_simple_test(
    test0059_mosul_raster_levant_tile_cut "rasters"
    "rasters/mosul,tiles/levant"
    mosul-cut.in 10
    )
add_simple_test(
    test0060_mosul_raster_levant_tile_outer "rasters"
    "rasters/mosul,tiles/levant"
    mosul-outer.in 10
    )

add_simple_test(
    test0061_mosul_rasters_only "rasters"
    "rasters-only" mosul-outer.in 10
    )
add_simple_test(
    test0062_mosul_rasters_and_tiles "rasters;tiles"
    "tiles-and-rasters" mosul-outer.in 10
    )
```

ORNL/TM-2017/145

INTERNAL DISTRIBUTION

1. R. W. Lee
2. C. D. Sulfredge
3. ORNL Office of Technical Information and Classification

EXTERNAL DISTRIBUTION

4. Todd Hann, Chief, Hazards & Effects Modeling Division, Information Sciences & Applications Department, Defense Threat Reduction Agency, 8725 John J. Kingman Rd., Stop 6201, Ft Belvoir, VA 22060-6201