SLEUTH Master Class

Keith C. Clarke
Department of Geography
UC Santa Barbara

Summary

- Model theory and operation
- Data requirements
- Calibration
- Outputs

Urban Cellular Automata

- Cells are pixels
- States are land uses
- Time is “units”, e.g. years
- Rules determine growth and change
- Different models have different rule sets
- Many models now developed, few tested
- Requiem for large scale models (Lee)

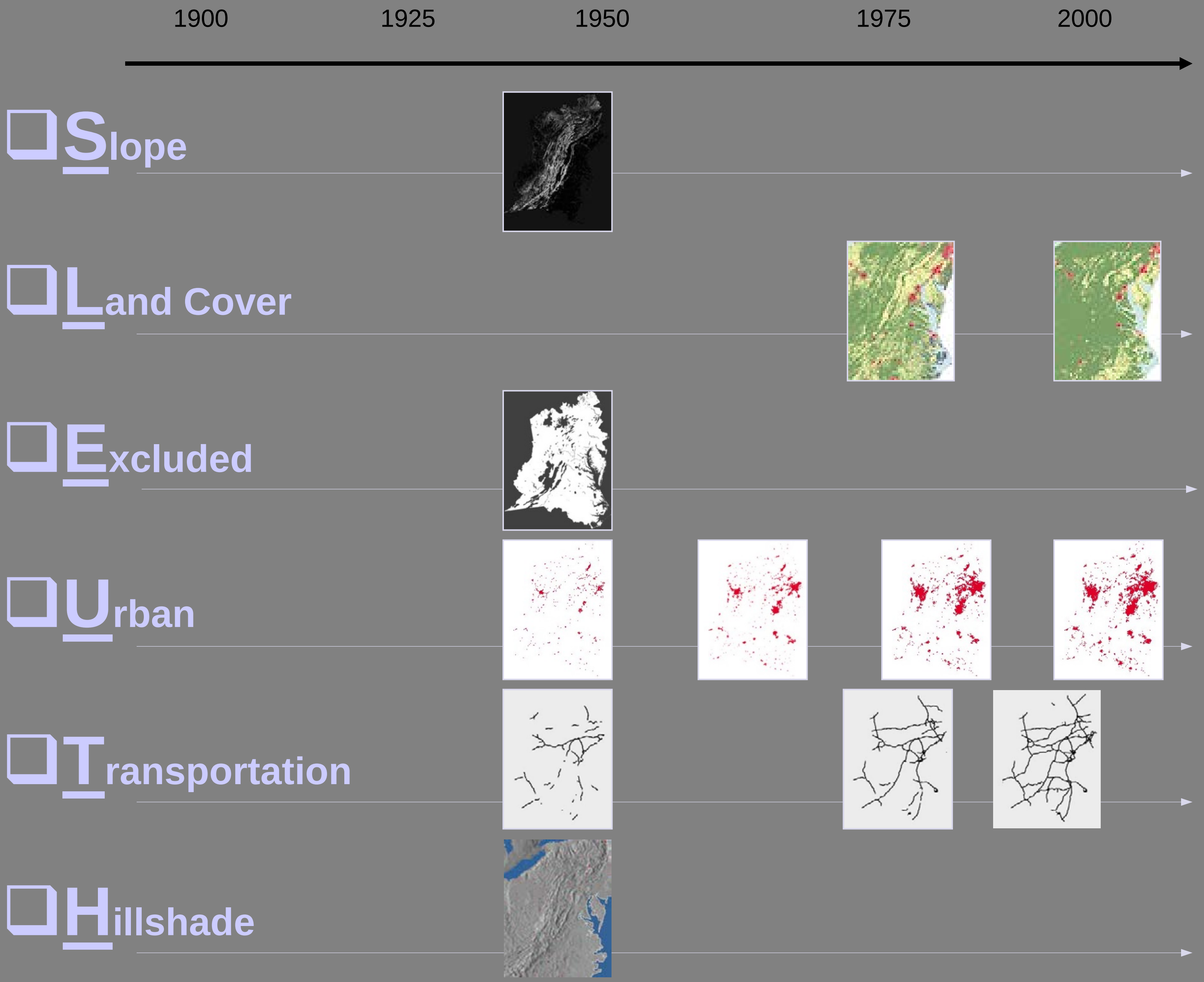
SLEUTH Model handles land use and urban growth

- Can use any level of consistent, space filling classification
- Needs two LULC layers to compute static Markov matrix
- Based on the concept of deltatrons
  - Generates synthetic LU change based on transition matrix and enforced spatial/temporal autocorrelation
  - Applies CA in change space
  - LU change = f(urban growth in last time period)

Project Web Site

- Set of background materials, e.g. publications
- Documentation as web pages in HTML
- Source Code for model in C
- Version 3.0 and SLEUTHGA available
- Updated version for Linux and cygwin
- Uses utilities and GD GIF libraries
- Parallel version requires MPI
- Set of sample calibration data demo_city
- **http://www.ncgia.ucsb.edu/projects/gig/ncgia.html**

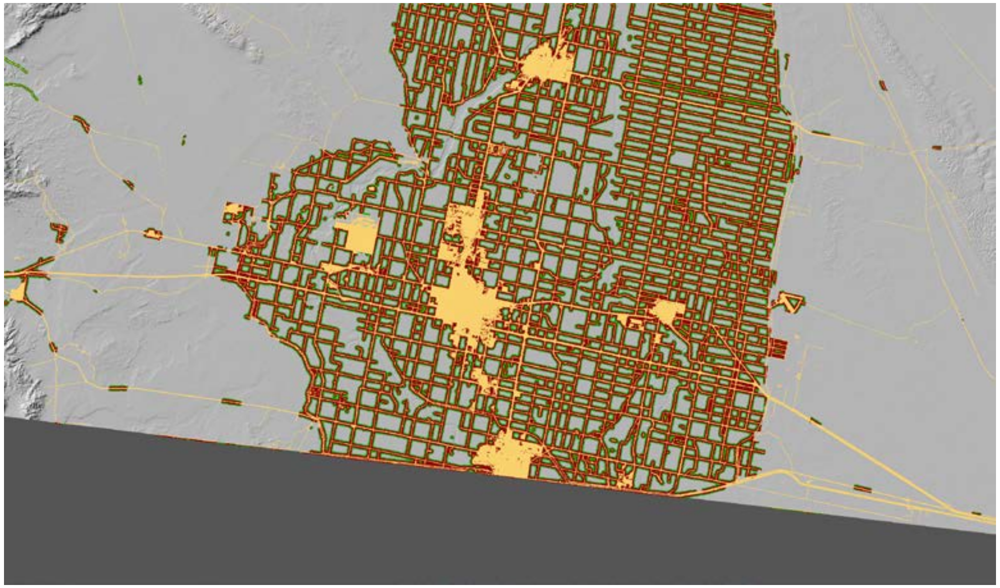
Data Requirements


1900
1925
1950
1975
2000
Slope
Land Cover
Excluded
Urban
Transportation
Hillshade

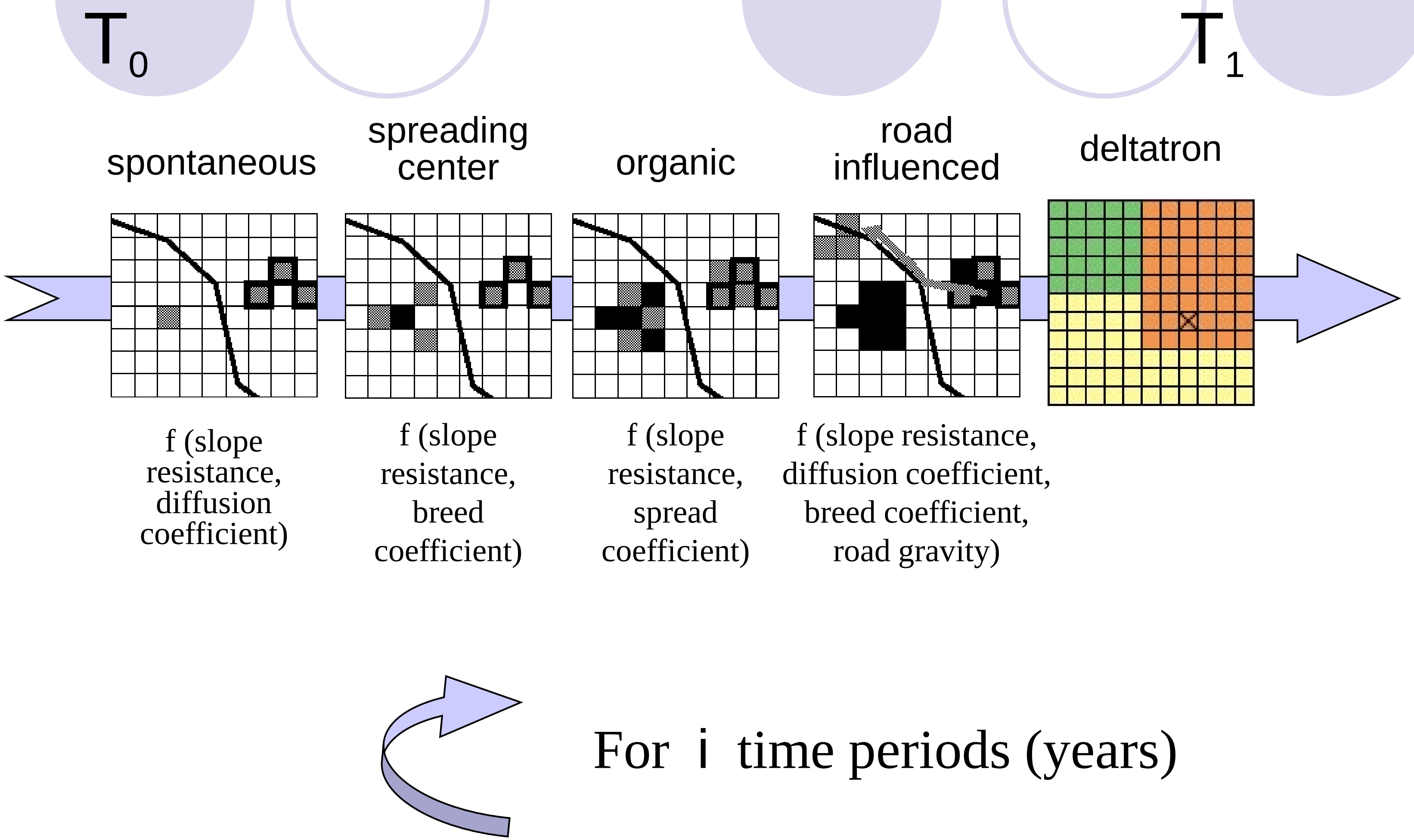
Thematic Data Input: Issues

- Vertical Integration of Temporal Data Layers
- Misregistration produces artificial change
- Deurbanization particularly upsetting to model
- Road breaks should be avoided, scale effect
- Avoid areas with zero growth

How SLEUTH works

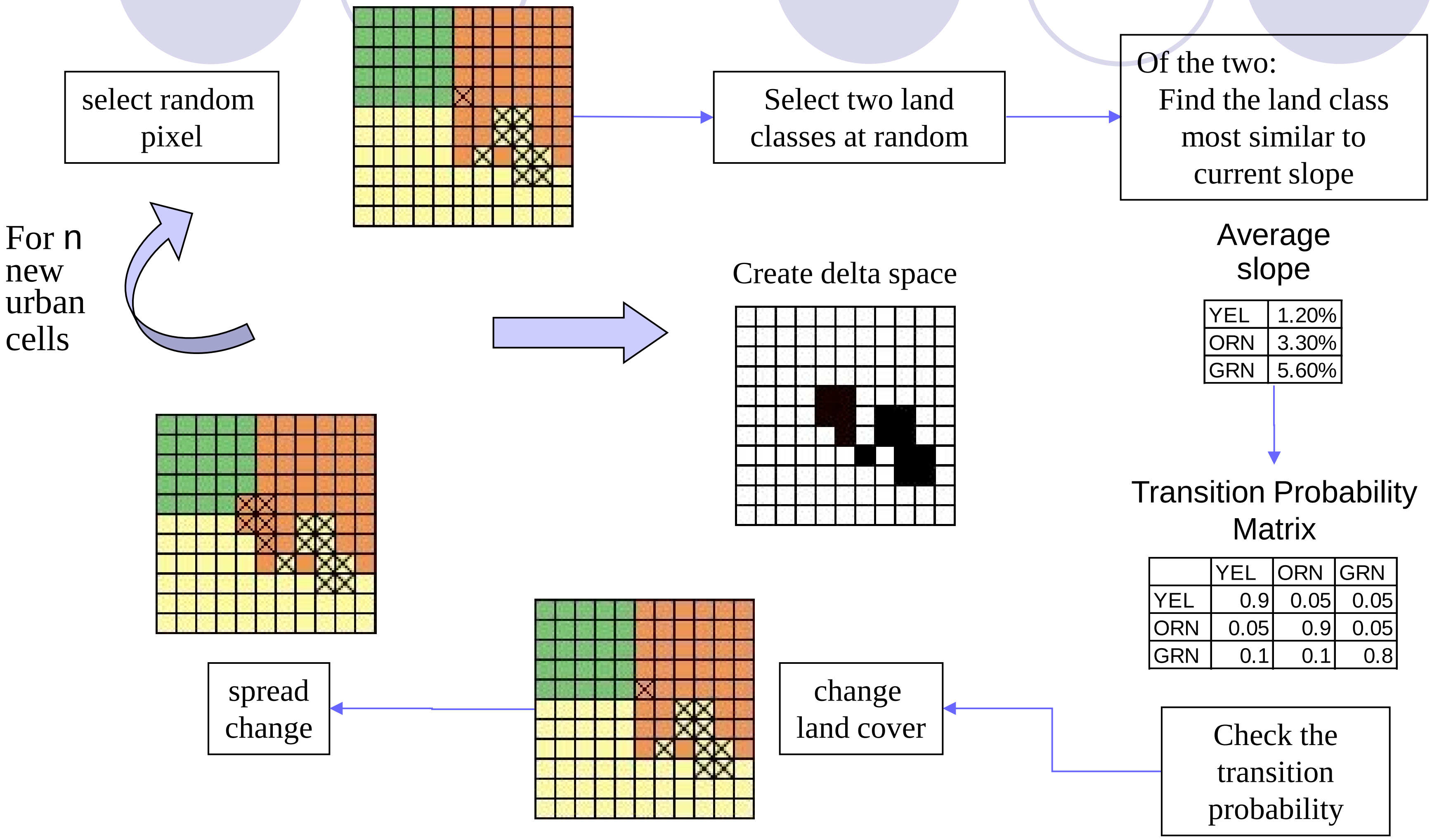


Behavior Rules



Deltatron Land Cover Model

Phase 1: Create change



Average slope

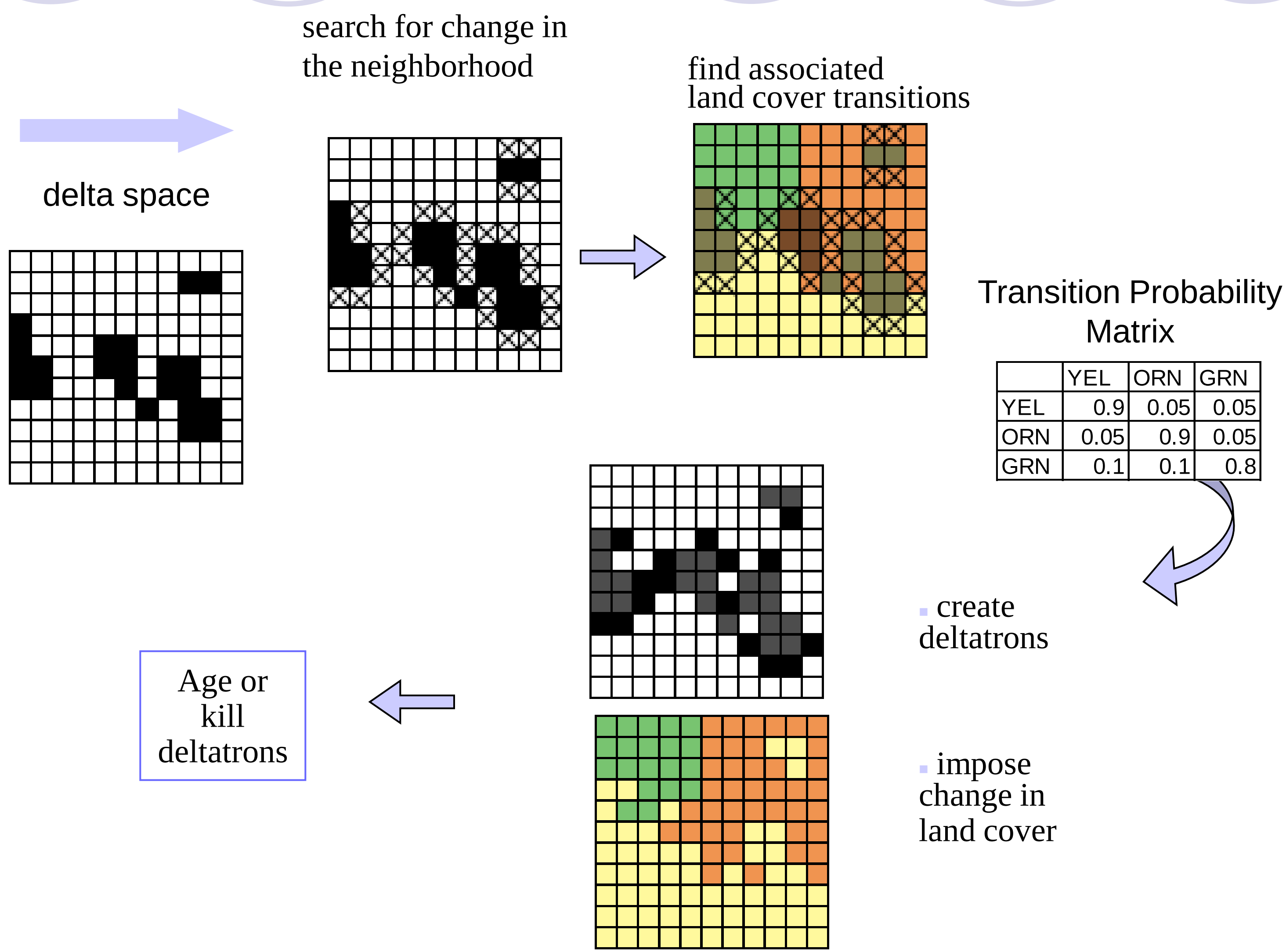
YEL	1.20%
ORN	3.30%
GRN	5.60%

Transition Probability Matrix

	YEL	ORN	GRN
YEL	0.9	0.05	0.05
ORN	0.05	0.9	0.05
GRN	0.1	0.1	0.8

Deltatron Land Cover Model

Phase 2: Perpetuate change



	YEL	ORN	GRN
YEL	0.9	0.05	0.05
ORN	0.05	0.9	0.05
GRN	0.1	0.1	0.8

UGM Process Flow
Data Set Preparation

Create Geographic Temporal Database

- Source data
  - historical maps, areal photographs, remotely sensed data, GIS vector/grid data
- Select by attribute
  - urban
  - transportation
  - landuse
  - excluded
  - slope
- Geo-registration
  - extent (lat, long)
- Data type standardization
  - vector to raster
  - ArcInfo vector data: LINEGRID or POLYGRID
- resolution (rows, columns)

Process control: The Scenario File

- To run, in the scenerio directory
- ../grow [mode] [scenariofilename]
- File contains all necessary data for run
- Sets all parameters, constants
- Names files
- Sets echo options
- Controls colors etc
- Includes #comments to guide

Scenario file: Master control: Header

- # FILE: 'scenario file' for SLEUTH land cover transition model
- #   (UGM v3.0)
- #   Comments start with #
- #
- #   I. Path Name Variables
- # II. Running Status (Echo)
- # III. Output ASCII Files
- # IV. Log File Preferences
- #   V. Working Grids
- # VI. Random Number Seed
- # VII. Monte Carlo Iteration
- #VIII. Coefficients
- #   A. Coefficients and Growth Types
- #   B. Modes and Coefficient Settings
- # IX. Prediction Date Range
- #   X. Input Images
- # XI. Output Images
- # XII. Colortable Settings
- #   A. Date_Color
- #   B. Non-Landuse Colortable
- #   C. Land Cover Colortable
- #   D. Growth Type Images
- #   E. Deltatron Images
- #XIII. Self Modification Parameters

Scenario file: Basic settings, I/O

- # I.PATH NAME VARIABLES
- #   INPUT_DIR: relative or absolute path where input image files and
- #   (if modeling land cover) 'landuse.classes' file are
- #   located.
- #   OUTPUT_DIR: relative or absolute path where all output files will
- #   be located.
- #   WHIRLGIF_BINARY: relative path to 'whirlgif' gif animation program.
- #   These must be compiled before execution.
- INPUT_DIR=../Input/demo200/
- OUTPUT_DIR=../Output/demo200_land_test/
- WHIRLGIF_BINARY=../Whirlgif/whirlgif

- # II. RUNNING STATUS (ECHO)
- # Status of model run, monte carlo iteration, and year will be
- # printed to the screen during model execution.
- ECHO(YES/NO)=yes

- # III. Output Files
- # INDICATE TYPES OF ASCII DATA FILES TO BE WRITTEN TO OUTPUT_DIRECTORY.
- #
- #   COEFF_FILE: contains coefficient values for every run, monte carlo
- #   iteration and year.
- #   AVG_FILE: contains measured values of simulated data averaged over
- #   monte carlo iterations for every run and control year.
- #   STD_DEV_FILE: contains standard diviation of averaged values
- #   in the AVG_FILE.
- #   MEMORY_MAP: logs memory map to file 'memory.log'
- #   LOGGING: will create a 'LOG_#' file where # signifies the processor
- #   number that created the file if running code in parallel.
- #   Otherwise, # will be 0. Contents of the LOG file may be
- #   described below.
- WRITE_COEFF_FILE(YES/NO)=yes
- WRITE_AVG_FILE(YES/NO)=yes
- WRITE_STD_DEV_FILE(YES/NO)=yes
- WRITE_MEMORY_MAP(YES/NO)=no
- LOGGING(YES/NO)=YES

Scenario file: Logging control

```
# IV. Log File Preferences
# INDICATE CONTENT OF LOG_# FILE (IF LOGGING == ON).
#   LANDCLASS_SUMMARY: (if landuse is being modeled) summary of input
#                    from 'landuse.classes' file
#   SLOPE_WEIGHTS(YES/NO): annual slope weight values as effected
#                       by slope_coeff
#   READS(YES/NO)= notes if a file is read in
#   WRITES(YES/NO)= notes if a file is written
#   COLORTABLES(YES/NO)= rgb lookup tables for all colortables generated
#   PROCESSING_STATUS(0:off/1:low verbosity/2:high verbosity)=
#   TRANSITION_MATRIX(YES/NO)= pixel count and annual probability of
#                          land class transitions
#   URBANIZATION_ATTEMPTS(YES/NO)= number of times an attempt to urbanize
#                              a pixel occurred
#   INITIAL_COEFFICIENTS(YES/NO)= initial coefficient values for
#                            each monte carlo
#   BASE_STATISTICS(YES/NO)= measurements of urban control year data
#   DEBUG(YES/NO)= data dump of igrid object and grid pointers
#   TIMINGS(0:off/1:low verbosity/2:high verbosity)= time spent within
#     each module. If running in parallel, LOG_0 will contain timing for
#     complete job.
LOG_LANDCLASS_SUMMARY(YES/NO)=yes
LOG_SLOPE_WEIGHTS(YES/NO)=no
LOG_READS(YES/NO)=no
LOG_WRITES(YES/NO)=no
LOG_COLORTABLES(YES/NO)=no
LOG_PROCESSING_STATUS(0:off/1:low verbosity/2:high verbosity)=1
LOG_TRANSITION_MATRIX(YES/NO)=yes
LOG_URBANIZATION_ATTEMPTS(YES/NO)=no
LOG_INITIAL_COEFFICIENTS(YES/NO)=no
LOG_BASE_STATISTICS(YES/NO)=yes
LOG_DEBUG(YES/NO)= yes
LOG_TIMINGS(0:off/1:low verbosity/2:high verbosity)=1
```

Monte Carlo Iterations/Working Grids

- # V. WORKING GRIDS
- # The number of working grids needed from memory during model execution is
- # designated up front. This number may change depending upon modes. If
- # NUM_WORKING_GRIDS needs to be increased, the execution will be exited
- # and an error message will be written to the screen and to 'ERROR_LOG'
- # in the OUTPUT_DIRECTORY. If the number may be decreased an optimal
- # number will be written to the end of the LOG_0 file.
- NUM_WORKING_GRIDS=4
- # VI. RANDOM NUMBER SEED
- # This number initializes the random number generator. This seed will be
- # used to initialize each model run.
- RANDOM_SEED=20190607
- # VII. MONTE CARLO ITERATIONS
- # Each model run may be completed in a monte carlo fashion.
- # For CALIBRATION or TEST mode measurements of simulated data will be
- # taken for years of known data, and averaged over the number of monte
- # carlo iterations. These averages are written to the AVG_FILE, and
- # the associated standard diviation is written to the STD_DEV_FILE.
- # The averaged values are compared to the known data, and a Pearson
- # correlation coefficient measure is calculated and written to the
- # control_stats.log file. The input per run may be associated across
- # files using the 'index' number in the files' first column.
- #
- MONTE_CARLO_ITERATIONS=4

Calibration Instructions

- # VIII. COEFFICIENTS
- # The coefficients effect how the growth rules are applied to the data.
- # Setting requirements:
- #   *_START values >= *_STOP values
- #   *_STEP values > 0
- #   if no coefficient increment is desired:
- #   *_START == *_STOP
- #   *_STEP == 1
- # For additional information about how these values affect simulated
- # land cover change see our publications and PROJECT GIGALOPOLIS
- # site: (www.ncgia.ucsb.edu/project/gig/About/abGrowth.htm).
- # A. COEFFICIENTS AND GROWTH TYPES
- #   DIFFUSION: affects SPONTANEOUS GROWTH and search distance along the
- #   road network as part of ROAD INFLUENCED GROWTH.
- #   BREED: NEW SPREADING CENTER probability and affects number of ROAD
- #   INFLUENCED GROWTH attempts.
- #   SPREAD: the probabilty of ORGANIC GROWTH from established urban
- #   pixels occuring.
- #   SLOPE_RESISTANCE: affects the influence of slope to urbanization. As
- #   value increases, the ability to urbanize
- #   ever steepening slopes decreases.
- #   ROAD_GRAVITY: affects the outward distance from a selected pixel for
- #   which a road pixel will be searched for as part of
- #   ROAD INFLUENCED GROWTH.

Calibration settings

- #
- # B. MODES AND COEFFICIENT SETTINGS
- # TEST: TEST mode will perform a single run through the historical
- # data using the CALIBRATION_*_START values to initialize
- # growth, complete the MONTE_CARLO_ITERATIONS, and then conclude
- # execution. GIF images of the simulated urban growth will be
- # written to the OUTPUT_DIRECTORY.
- # CALIBRATE: CALIBRATE will perform monte carlo runs through the
- # historical data using every combination of the
- # coefficient values indicated. The CALIBRATION_*_START
- # coefficient values will initialize the first run. A
- # coefficient will then be increased by its *_STEP value,
- # and another run performed. This will be repeated for all
- # possible permutations of given ranges and increments.
- # PREDICTION: PREDICTION will perform a single run, in monte carlo
- # fashion, using the PREDICTION_*_BEST_FIT values
- # for initialization.

- CALIBRATION_DIFFUSION_START= 0
- CALIBRATION_DIFFUSION_STEP= 20
- CALIBRATION_DIFFUSION_STOP= 100

- CALIBRATION_BREED_START= 0
- CALIBRATION_BREED_STEP= 20
- CALIBRATION_BREED_STOP= 100

- CALIBRATION_SPREAD_START= 0
- CALIBRATION_SPREAD_STEP= 20
- CALIBRATION_SPREAD_STOP= 100

- CALIBRATION_SLOPE_START= 0
- CALIBRATION_SLOPE_STEP= 20
- CALIBRATION_SLOPE_STOP= 100

- CALIBRATION_ROAD_START= 0
- CALIBRATION_ROAD_STEP= 20
- CALIBRATION_ROAD_STOP= 100

- PREDICTION_DIFFUSION_BEST_FIT= 20
- PREDICTION_BREED_BEST_FIT= 20
- PREDICTION_SPREAD_BEST_FIT= 20
- PREDICTION_SLOPE_BEST_FIT= 20
- PREDICTION_ROAD_BEST_FIT= 20

Input

- # IX. PREDICTION DATE RANGE
- # The urban and road images used to initialize growth during
- # prediction are those with dates equal to, or greater than,
- # the PREDICTION_START_DATE. If the PREDICTION_START_DATE is greater
- # than any of the urban dates, the last urban file on the list will be
- # used. Similarly, if the PREDICTION_START_DATE is greater
- # than any of the road dates, the last road file on the list will be
- # used. The prediction run will terminate at PREDICTION_STOP_DATE.
- #
- PREDICTION_START_DATE=1990
- PREDICTION_STOP_DATE=2010

- # X. INPUT IMAGES
- # The model expects grayscale, GIF image files with file name
- # format as described below. For more information see our
- # PROJECT GIGALOPOLIS web site:
- # (www.ncgia.ucsb.edu/project/gig/About/dtInput.htm).
- #
- # IF LAND COVER IS NOT BEING MODELED: Remove or comment out
- # the LANDUSE_DATA data input flags below.
- #
- #   < > = user selected fields
- #   [< >] = optional fields
- #
- # Urban data GIFs
- # format: <location>.urban.<date>.[<user info>].gif
- #
- #
- URBAN_DATA= demo200.urban.1930.gif
- URBAN_DATA= demo200.urban.1950.gif
- URBAN_DATA= demo200.urban.1970.gif
- URBAN_DATA= demo200.urban.1990.gif
- #

Input (ctd)

- # Road data GIFs
- # format: <location>.roads.<date>.[<user info>].gif
- #
- ROAD_DATA= demo200.roads.1930.gif
- ROAD_DATA= demo200.roads.1950.gif
- ROAD_DATA= demo200.roads.1970.gif
- ROAD_DATA= demo200.roads.1990.gif
- #
- # Landuse data GIFs
- # format: <location>.landuse.<date>.[<user info>].gif
- #
- LANDUSE_DATA= demo200.landuse.1930.gif
- LANDUSE_DATA= demo200.landuse.1990.gif
- #
- # Excluded data GIF
- # format: <location>.excluded.[<user info>].gif
- #
- EXCLUDED_DATA= demo200.excluded.gif
- #
- # Slope data GIF
- # format: <location>.slope.[<user info>].gif
- #
- SLOPE_DATA= demo200.slope.gif
- #
- # Background data GIF
- # format: <location>.hillshade.[<user info>].gif
- #
- #BACKGROUND_DATA= demo200.hillshade.gif
- BACKGROUND_DATA= demo200.hillshade.water.gif

Output

- # XI. OUTPUT IMAGES
- #   WRITE_COLOR_KEY_IMAGES: Creates image maps of each colortable.
- #   File name format: 'key_[type]_COLORMAP'
- #   where [type] represents the colortable.
- #   ECHO_IMAGE_FILES: Creates GIF of each input file used in that job.
- #   File names format: 'echo_of_[input_filename]'
- #   where [input_filename] represents the input name.
- #   ANIMATION: if whirlgif has been compiled, and the WHIRLGIF_BINARY
- #   path has been defined, animated gifs begining with the
- #   file name 'animated' will be created in PREDICT mode.
- WRITE_COLOR_KEY_IMAGES(YES/NO)=yes
- ECHO_IMAGE_FILES(YES/NO)=yes
- ANIMATION(YES/NO)= yes

The Color Tables

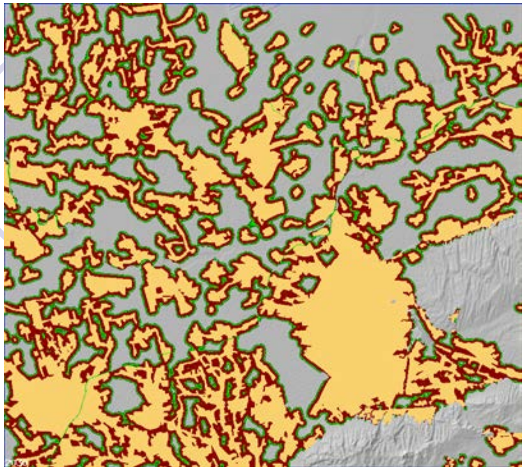
- # XII. COLORTABLE SETTINGS
- # A. DATE COLOR SETTING
- #   The date will automatically be placed in the lower left corner
- #   of output images. DATE_COLOR may be designated in with red, green,
- #   and blue values (format: <red_value, green_value, blue_value>)
- #   or with hexadecimal begining with '0X' (format: <0X######>).
- #default DATE_COLOR= 0XFFFFFF white
- DATE_COLOR=   0XFFFFFF #white

- # B. URBAN (NON-LANDUSE) COLORTABLE SETTINGS
- #   1. URBAN MODE OUTPUTS
- #   TEST mode: Annual images of simulated urban growth will be
- #   created using SEED_COLOR to indicate urbanized areas.

- #   CALIBRATE mode: Images will not be created.
- #   PREDICT mode: Annual probability images of simulated urban
- #   growth will be created using the PROBABILITY
- #   _COLORTABLE. The initializing urban data will be
- #   indicated by SEED_COLOR.
- #
- #   2. COLORTABLE SETTINGS
- #   SEED_COLOR: initializing and extrapolated historic urban extent

- #   WATER_COLOR: BACKGROUND_DATA is used as a backdrop for
-
- #   simulated urban growth. If pixels in this file
- #   contain the value zero (0), they will be filled
- #   with the color value in WATER_COLOR. In this way,
- #   major water bodies in a study area may be included
- #   in output images.
- #SEED_COLOR= 0XFFFF00 #yellow
- SEED_COLOR= 249, 209, 110 #pale yellow
- #WATER_COLOR= 0X0000FF # blue
- WATER_COLOR= 20, 52, 214 # royal blue

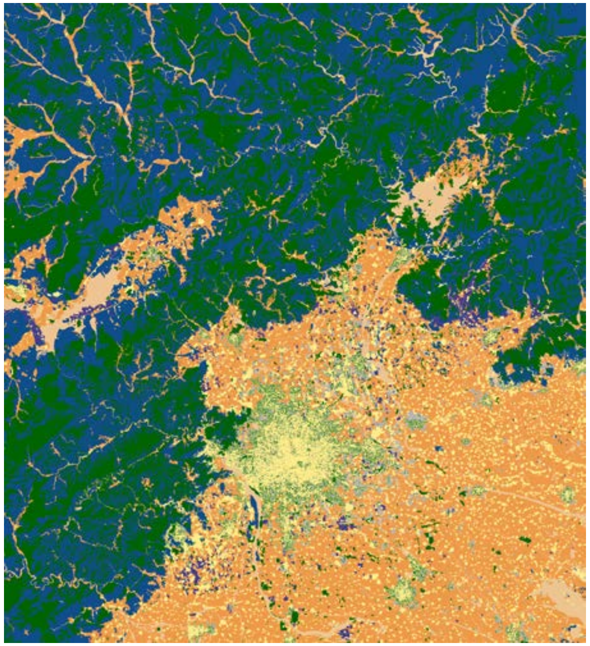
Forecast image



- #   3. PROBABILITY COLORTABLE FOR URBAN GROWTH
- #   For PREDICTION, annual probability images of urban growth
- #   will be created using the monte carlo iterations. In these
- #   images, the higher the value the more likely urbanizaion is.
- #   In order to interpret these 'continuous' values more easily
- #   they may be color classified by range.
- #
- #   If 'hex' is not present then the range is transparent.
- #   The transparent range must be the first on the list.
- #   The max number of entries is 100.
- #   PROBABILITY_COLOR: a color value in hexadecimal that indicates
- #   a probability range.
- #   low/upper: indicate the boundaries of the range.
- #
- #   low, upper,   hex, (Optional Name)
- PROBABILITY_COLOR=   0,   50,   , #transparent
- PROBABILITY_COLOR=   50,   60, 0X005A00, #0, 90,0 dark green
- PROBABILITY_COLOR=   60,   70, 0X008200, #0,130,0
- PROBABILITY_COLOR=   70,   80, 0X00AA00, #0,170,0
- PROBABILITY_COLOR=   80,   90, 0X00D200, #0,210,0
- PROBABILITY_COLOR=   90,   95, 0X00FF00, #0,255,0 light green
- PROBABILITY_COLOR=   95, 100, 0X8B0000, #dark red

Land use color table

Beijing



- # C. LAND COVER COLORTABLE
- # Land cover input images should be in grayscale GIF image format.
- # The 'pix' value indicates a land class grayscale pixel value in
- # the image. If desired, the model will create color classified
- # land cover output. The output colortable is designated by the
- # 'hex/rgb' values.
- # pix: input land class pixel value
- # name: text string indicating land class
- # flag: special case land classes
- # URB - urban class (area is included in urban input data
- # and will not be transitioned by deltatron)
- # UNC - unclass (NODATA areas in image)
- # EXC - excluded (land class will be ignored by deltatron)
- # hex/rgb: hexidecimal or rgb (red, green, blue) output colors
- #
- # pix, name, flag, hex/rgb, #comment
- LANDUSE_CLASS= 0, Unclass , UNC , 0X000000
- LANDUSE_CLASS= 1, Urban , URB , 0X8b2323 #dark red
- LANDUSE_CLASS= 2, Agric , , 0Xffec8b #pale yellow
- LANDUSE_CLASS= 3, Range , , 0Xee9a49 #tan
- LANDUSE_CLASS= 4, Forest , , 0X006400
- LANDUSE_CLASS= 5, Water , EXC , 0X104e8b
- LANDUSE_CLASS= 6, Wetland , , 0X483d8b
- LANDUSE_CLASS= 7, Barren , , 0Xeec591

Growth rule image

- # D. GROWTH TYPE IMAGE OUTPUT CONTROL AND COLORTABLE
- #
- # From here you can control the output of the Z grid
- # (urban growth) just after it is returned from the spr_spread()
- # function. In this way it is possible to see the different types
- # of growth that have occured for a particular growth cycle.
- #
- # VIEW_GROWTH_TYPES(YES/NO) provides an on/off
- # toggle to control whether the images are generated.
- #
- # GROWTH_TYPE_PRINT_WINDOW provides a print window
- # to control the amount of images created.
- # format: <start_run>,<end_run>,<start_monte_carlo>,
- #   <end_monte_carlo>,<start_year>,<end_year>
- # for example:
- # GROWTH_TYPE_PRINT_WINDOW=run1,run2,mc1,mc2,year1,year2
- # so images are only created when
- # run1<= current run <=run2 AND
- # mc1 <= current monte carlo <= mc2 AND
- # year1 <= currrent year <= year2
- #
- # 0 == first
- VIEW_GROWTH_TYPES(YES/NO)=NO
- GROWTH_TYPE_PRINT_WINDOW=0,0,0,0,1995,2020
- PHASE0G_GROWTH_COLOR= 0xff0000 # seed urban area
- PHASE1G_GROWTH_COLOR= 0X00ff00 # diffusion growth
- PHASE2G_GROWTH_COLOR= 0X0000ff # NOT USED
- PHASE3G_GROWTH_COLOR= 0Xffff00 # breed growth
- PHASE4G_GROWTH_COLOR= 0Xffffff # spread growth
- PHASE5G_GROWTH_COLOR= 0X00ffff # road influenced growth

Deltatron behavior

- #***
- #
- # E. DELTATRON AGING SECTION
- #
- # From here you can control the output of the deltatron grid
- # just before they are aged
- #
- # VIEW_DELTATRON_AGING(YES/NO) provides an on/off
- # toggle to control whether the images are generated.
- #
- # DELTATRON_PRINT_WINDOW provides a print window
- # to control the amount of images created.
- # format: <start_run>,<end_run>,<start_monte_carlo>,
- #   <end_monte_carlo>,<start_year>,<end_year>
- # for example:
- # DELTATRON_PRINT_WINDOW=run1,run2,mc1,mc2,year1,year2
- # so images are only created when
- # run1<= current run <=run2 AND
- # mc1 <= current monte carlo <= mc2 AND
- # year1 <= currrent year <= year2
- #
- # 0 == first
- VIEW_DELTATRON_AGING(YES/NO)=NO
- DELTATRON_PRINT_WINDOW=0,0,0,0,1930,2020
- DELTATRON_COLOR= 0x000000 # index 0 No or dead deltatron
- DELTATRON_COLOR= 0X00FF00 # index 1 age = 1 year
- DELTATRON_COLOR= 0X00D200 # index 2 age = 2 year
- DELTATRON_COLOR= 0X00AA00 # index 3 age = 3 year
- DELTATRON_COLOR= 0X008200 # index 4 age = 4 year
- DELTATRON_COLOR= 0X005A00 # index 5 age = 5 year

- # XIII. SELF-MODIFICATION PARAMETERS

Finally, the constants

- #	SLEUTH is a self-modifying cellular automata. For more
- #	information see our PROJECT GIGALOPOLIS web site
- #	(www.ncgia.ucsb.edu/project/gig/About/abGrowth.htm)
- #	and publications (and/or grep 'self modification' in code).
- ROAD_GRAV_SENSITIVITY=0.01
- SLOPE_SENSITIVITY=0.1
- CRITICAL_LOW=0.97
- CRITICAL_HIGH=1.3
- #CRITICAL_LOW=0.0
- #CRITICAL_HIGH=10000000000000.0
- CRITICAL_SLOPE=21.0
- BOOM=1.01
- BUST=0.9

UGM Process Flow

Data Set Preparation

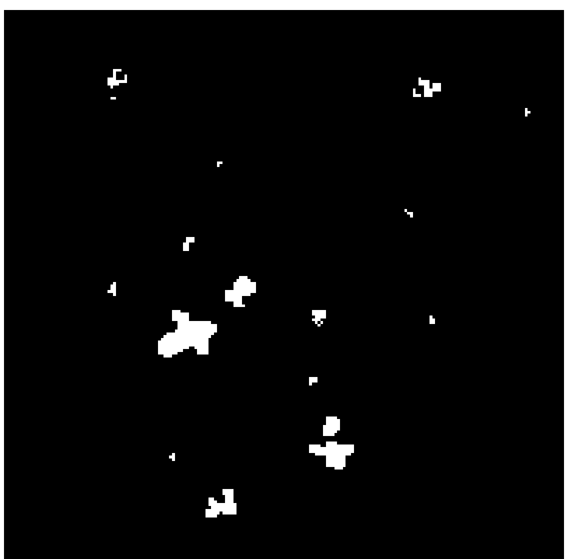
Image Format Specifics

Urban

Values: 0 = not urban, 0 < n < 255 = urban

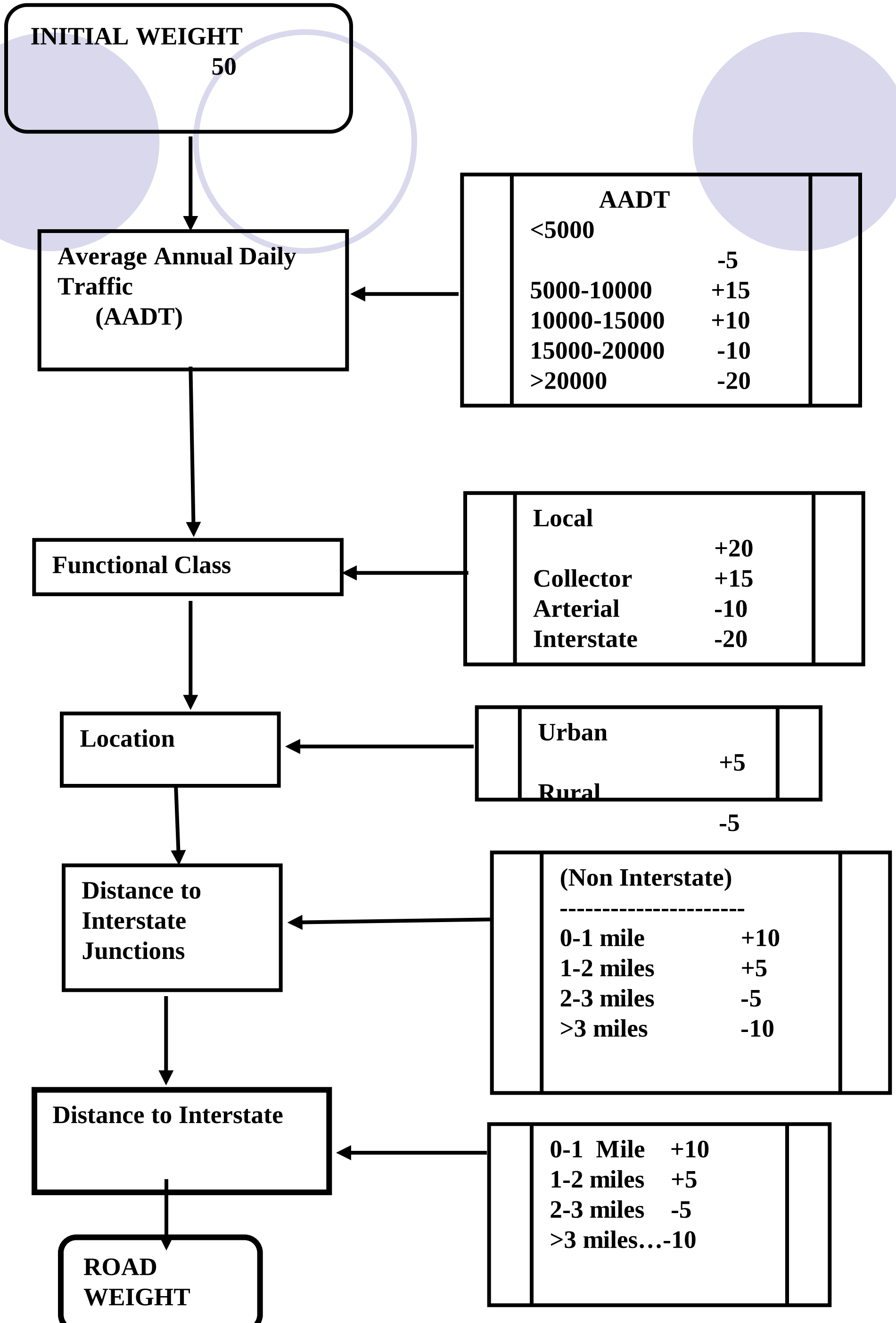
Roads

Values: 0 = not road, 0 < n < 255 = road



Clarke Urban Growth Model

Road Weight Algorithm



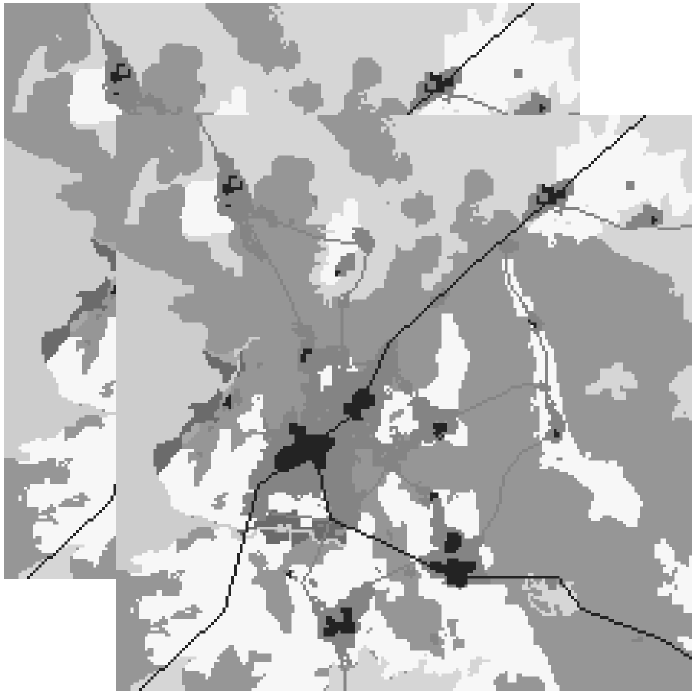
Toby N. Carlson, Dept. of Meteorology;

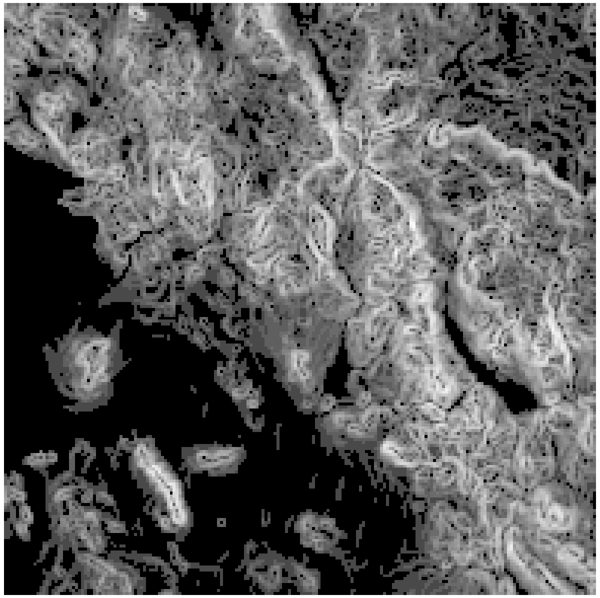
John T. Marker, Kostas Goulias, Pennsylvania Transportation Institute

Penn State University

UGM Process Flow
Data Set Preparation
Image Format Specifics

- Landuse: any method can be used
- Values: Each value matches a given classification value.
  - 1 = urban, 2 = agriculture, 3 = rangeland, etc.



- Slope: the average percent slope of the terrain is derived from a DEM
- 

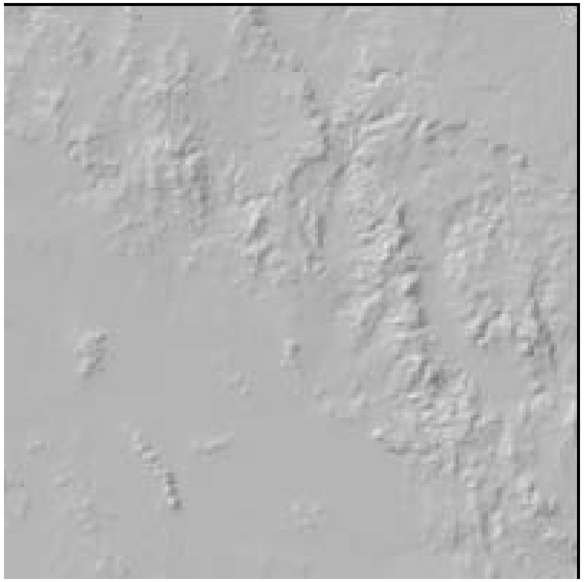
UGM Process Flow

Data Set Preparation

Image Format Specifics

- Excluded Areas: water bodies and land where urbanization cannot occur.
  - This layer may contain binary data (0 and 99) or ranged values indicating probabilities of exclusion.
  - Values: 0-99 = not excluded, 100 = excluded

- Background: hillshaded image of region (animation)



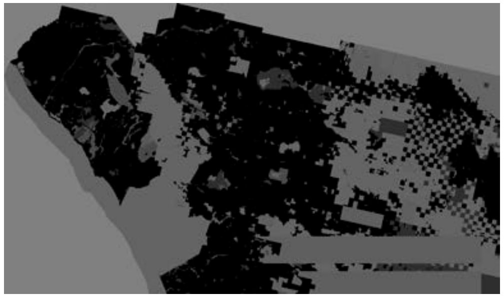
UGM Process Flow
Data Set Preparation

- Final data format must be as a GIF image.
  - ArcInfo: GRIDIMAGE -> TIF
  - xv: TIF -> GIF
  - Photoshop/GIMP
- Naming convention
- Build scenario file
- Test, calibrate, predict

Thematic Data Input

Exclusion Feature Hierarchy and Probability

- The exclusion layer
  - Previously: Binary
    - static possibility of growth occurring
  - The Latest: a range (0 - 100)



- Enables the exploration of zoning scenarios
  - e.g.; green zones and urban corridors

Calibration

Calibration

- Most essential element
- Ensures realism
- Ensures accountability and repeatability
- Tests sensitivity
- Required for complex systems models
- Conducted in Monte Carlo mode

The Traditional Method

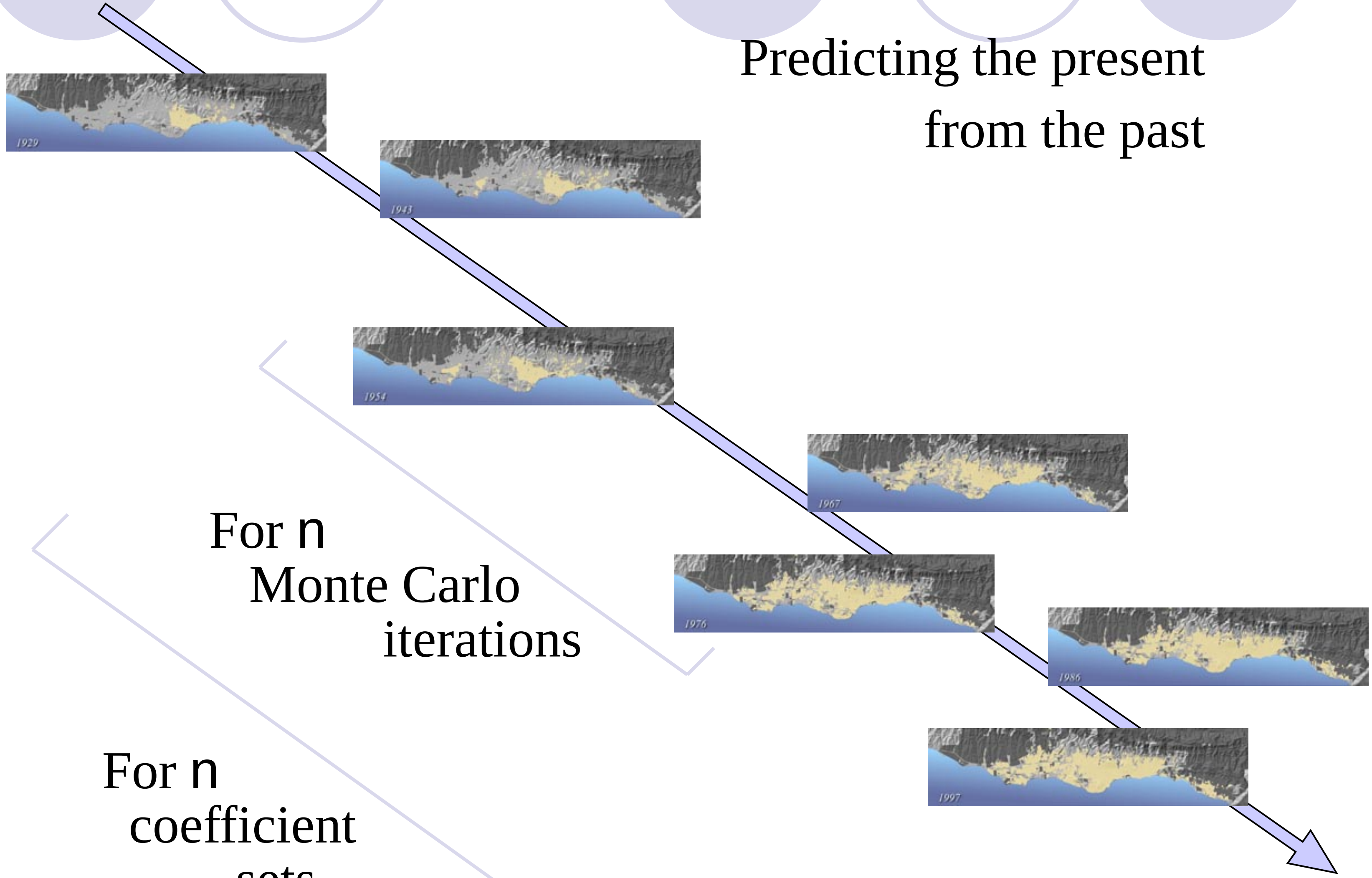
- “Brute force calibration”
- Phased exploration of parameter space
- Start with coarse parameter steps and coarsened spatial data
- Step to finer and finer data as calibration proceeds
- Good rather than best solution
- 5 parameters 0-100 = 101^5 permutations

The Problem

- “Model calibration for a medium sized data set and minimal data layers requires about 1200 CPU hours on a typical workstation”
- CS calls problem tractability

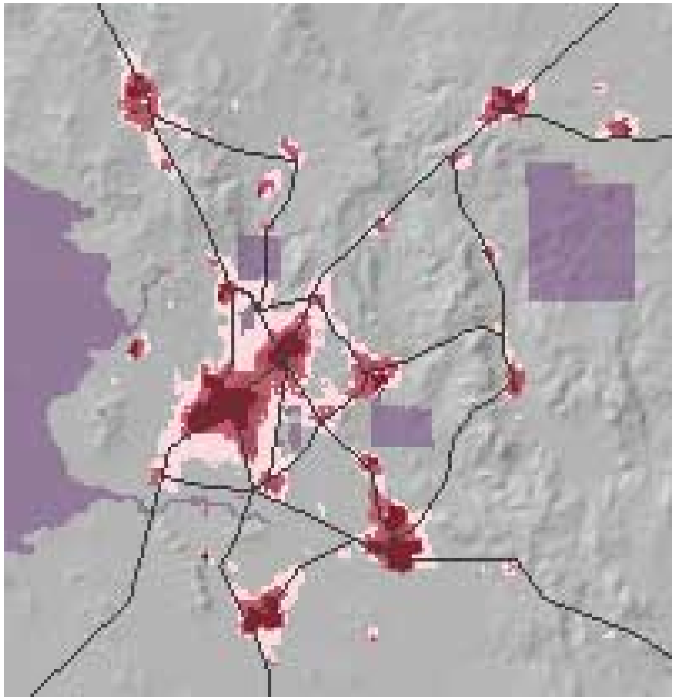
Calibration

Predicting the present from the past



UGM Process Flow
UGM Compilation

- Install cygwin, linux or other UNIX OS
- Download Programs and Data (into a new directory)
- Contents of downloaded UGM.tar.gz
  - Clarke Urban Growth Model
  - Land Cover Deltatron Model
  - gd libraries
  - Sample scenario file set to accept demo_city
  - demo_city data set
  - Utilities, e.g. read_data3.c computes OSM



SLEUTH Process Flow
UGM Compilation

- Set Up Model and Utilities
  - gunzip and untar the SLEUTH zip file
- Compile the gd libraries
  - by entering “make” in the GD subdirectory
- In the Model Directory
  - enter: "make" to compile the model
- Type: “./grow.exe"
  - this will begin the program
- The arguments determine what type of run, output and coeffiecient values are extracted form the scenario file.
  - Modes are: test, calibrate, (average) and predict (+evolve)
  - OSM or Lee Sallee often used as performance metric
- Verify results
  - compare stats from demo_city with documented results
  - Many use contingency matrix statistics (e.g. kappa) and landscape metrics

SLEUTH Brute Force Calibration

- Phases of calibration
  - Coarse
    - Iterations in large increments spanning coefficients' full range
    - images 1/4 full size (now full)
    - (e.g. START=0, STEP=20, STOP=100)
  - Fine
    - Increments are smaller with a more focused coefficient range
    - images 1/2 full size (now full)
    - (e.g. START=50, STEP=5;STOP=75)
  - Final
    - The coefficient range should be narrowed to single increments
    - (e.g. START=60, STEP=1, STOP=64)
    - images are full size

SLEUTH Calibration stages

Set constants and verify

- Update working directory
  - move project data (your GIF image files) into the model directory
  - edit *.dates and landuse classes to reflect your datasets
- Check values in scenario file, run grow.exe in test mode
- Examine the numbers computed to standard output
  - Make sure they make sense for your data.
  - Values to examine are in the stats file, and are echoed by the program. control_stats.log is key file
- Use a viewing tool such as xv to examine the file cumulate.final.gif which should show a map of the result.

SLEUTH Brute Force calibration

Coarse Calibration Run

Set Start, step, stop values

Select number of Monte Carlo iterations (4?)

Run calibration

enter: ”./grow.exe calibrate senario_file"

Monitor results as they are written to the file control.stats until the script completes with “wc”

Select "best" results, can rank with Excel, or use readdata3.exe for OSM

Version 3.0 innovations

- Recoded into modular flow ANSI C
- Dynamic memory allocation returned to flat memory
- Optimized for Cray memory model
- Parallelized
- Built MPI (message passing interface) link
- Several code speed-ups and fixes
- Code tested and verified against Version 2.1
- Visualization of rules, color specification

SLEUTH Calibration

- Brute force methodolgy
- Partitions and explores parameter space
- Scales across spatial resolutions
- Works in phases with increasing parametric and spatial detail
- Is embarrassingly parallel!
- Massive speed-up attained

The Cost: MPI

- MPI is a library of Fortran and C callable routines
- Handles inter-process communication
- Standard since 1993
- Queries environment for number of available processors
- If processors=1, runs serially

SLEUTH Calibration by GA

- Genetic algorithm coded for GeoComputation in 2011, uses OSM
- Code posted on GitHub, latest on Gigalopolis site
- Improved to take command line arguments
- 2 papers in 2017
- Considerable speed up (about 1/5)
- SLEUTH-GA posted to site, substitutes for calibration phase only

How GA works

- Generates random chromosome with *n* genes, each with one parameter combination {2,89,5,67,98}
- Calibrates for all genes, computes OSM and ranks the genes
- Successful genes can “mate” using recombination e.g.
- 1. {17,2,34,98,12}
- 2. {45,7,14,12,78} -> {17,2,34,12,78}

Mutation

- Mutations are of three types: mixing, boosting and randomization
- {34,25,12,87,99} -> {34,25,87,12,99}
- {34,25,12,87,99}->{35,25,12,87,99}
- {34,25,12,87,99}->{45,56,12,87,99}

Selection

- Rank all genes by OSM
- Select some for breeding
- Select some for mutation
- Replace remainder with random combinations
    - Important to get out of local maximum

The GA at work

- Repeat for all genes in chromosome=one generation + apply breeding, mutation and selection
- Repeat for next generation until fitness no longer improves
- Select based on best gene, or best chromosome

Coefficient Changes during GA calibration

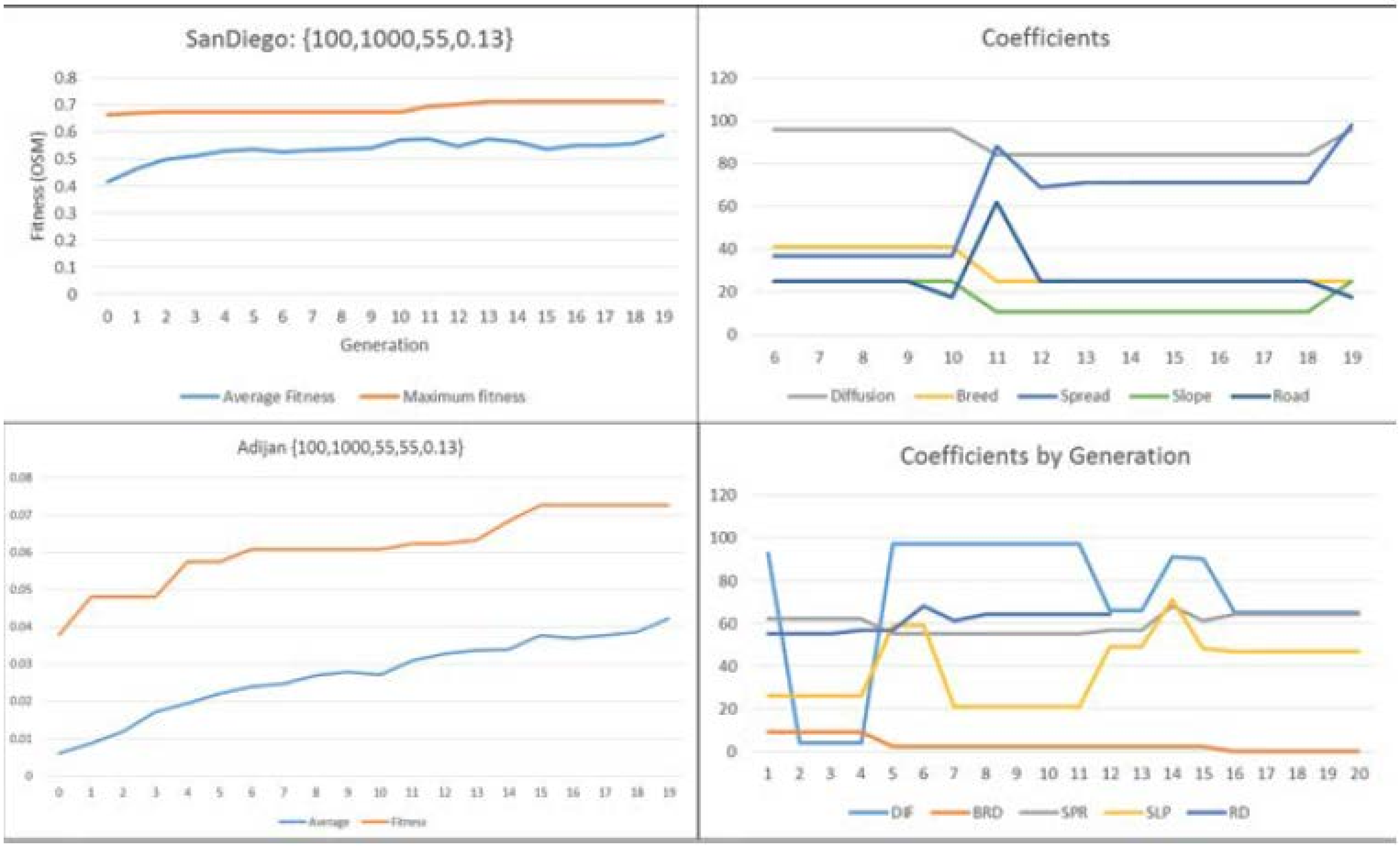


Fig. 2. Values during test GA calibration runs for San Diego and Andijan, showing coefficient evolution and fitness improvement.

SLEUTHGA Output

- Mon, Feb 11, 2019 1:18:12 PM
- GA constants
- Population size: 55 Generations: 100
- Mutation Rate: 0.130000 Number of offspring: 55
- Number replaced: 50
- …………
- growth.c 122 **
- growth.c 130 Monte Carlo = 4 of 4
- growth.c 133 proc_GetCurrentYear=2001
- growth.c 135 proc_GetStopYear=2017
- 2002 2003 2004 2005 2006 2007 2008 2009
- 2010 2011 2012 2013 2014 2015 2016 2017
- Standard Dev : 0.034988
- Generation: 19, Average: 0.181214
- Sub Population: 0
- Gene: 0 89, 57, 41, 57, 98, 0.231827
- Gene: 1 72, 21, 25, 56, 69, 0.227962
- Gene: 2 54, 17, 73, 4, 30, 0.226011
- ……………………………..
- Gene: 20 2, 57, 19, 44, 45, 0.192516
- Sub Population: 6
- Gene: 54 38, 9, 96, 42, 10, 0.075119
- Gene: 55 80, 57, 19, 42, 45, 0.034811
- Evolution begins...
- Wed, Feb 13, 2019 12:21:09 AM

Model Outputs

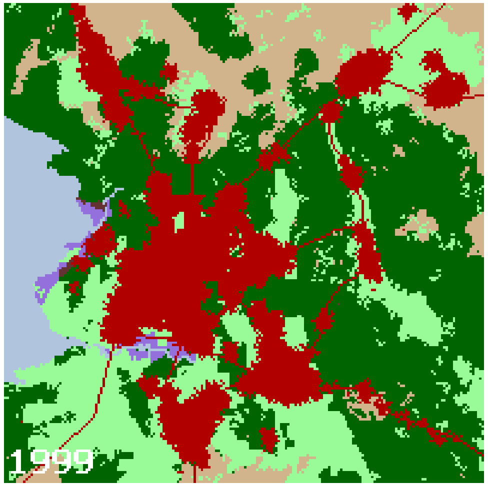
UGM Process Flow
UGM Products

- Numeric
  - The numerical output consists of goodness-of-fit calculations contained in the stats file.
- Graphic
  - single images
    - single run: a snapshot of a particular year
    - Monte Carlo: a cumulative Monte Carlo image that results from multiple runs. These Monte Carlo images will show a probability of urbanization for a given year.
  - animations
    - The model can merge these images together to produce an animated gif of urban growth over time.
- Integration
  - The images can also be introduced back into a GIS environment and used as data layers for further analysis in their spatial context.
  - ArcInfo (for example)
    - Transform images into Arc acceptable format (e.g.: TIFF)
    - Transform images into grids with Arc: GRIDIMAGE
    - Georeference grids with Grid: CONTROLPOINTS

SLEUTH Outputs

- Statistics
- Logs
- Images
- Uncertainty maps
- Animations

Land cover predictions and model calibration



UGM Process Flow Prediction

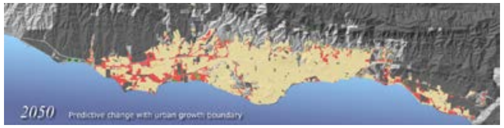
- Select best parameter combination
- Make this the set in calibration mode e.g.
  - DIFFUSION_START=54
  - DIFFUSION_STEP=1
  - DIFFUSION_STOP=54
- Increase # of Monte Carlo iterations, e.g. to 100
- Run again in calibration mode
- Select “final” calibration parameters from avg.log
- Enter these in
  - PREDICTION_DIFFUSION_BEST_FIT= 100
  - PREDICTION_BREED_BEST_FIT= 100
  - PREDICTION_SPREAD_BEST_FIT= 100
  - PREDICTION_SLOPE_BEST_FIT= 1
  - PREDICTION_ROAD_BEST_FIT= 69
- Select desired outputs, e.g. end date, animation choice, and run again in predict mode

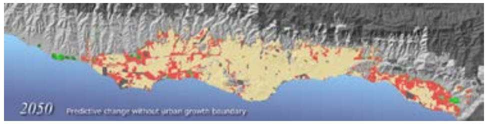
Prediction (the future from the present)

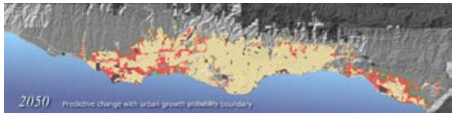
- Probability Images



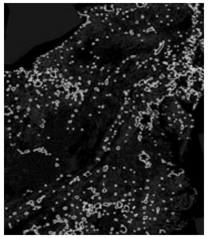
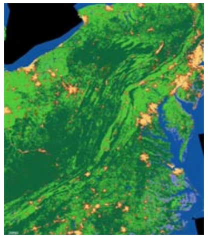
- Alternate Scenarios







- Land Cover Uncertainty

- Color by phase, or show deltatrons

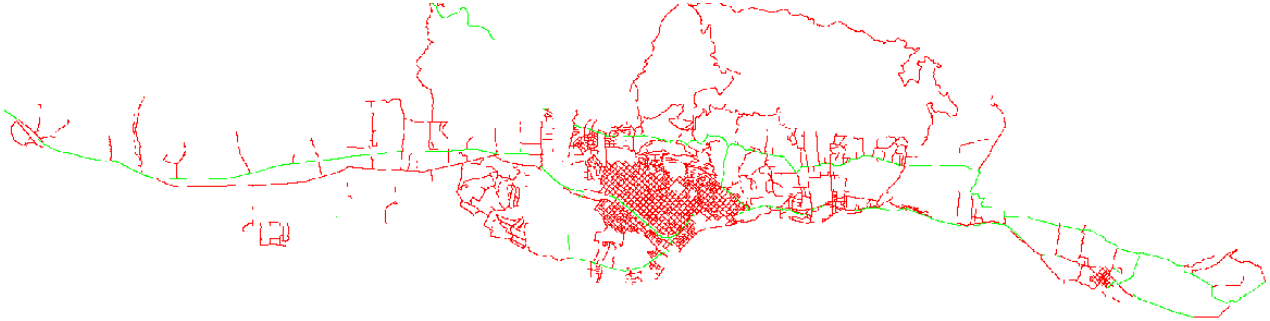
Some lessons learned

- Scale can matter
- LU class aggregation effects as expected
- Ways to speed-up, e.g. genetic algorithms
- Overfit possible, but calibration procedure still works!
- Optimal SLEUTH metric
- DNA experiments possible

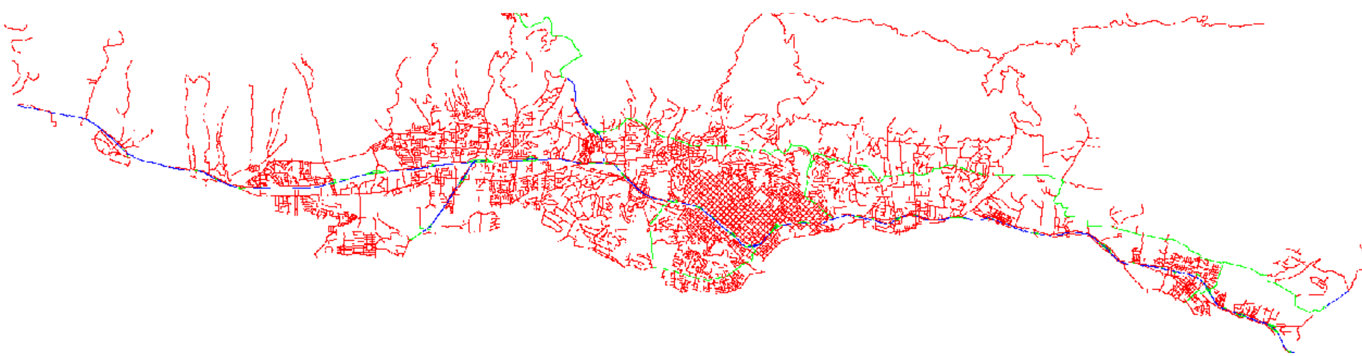
Suggested Issues list

- Uncertainty. Monte Carlo for urban. Uncertainty computed for LU.
- Detailed urban LU, very high resolution
- Rule modification
- Integrate with MCE
- Add density—Saxena
- Use slope layer for land suitability—Chaudhuri, Kolkatta

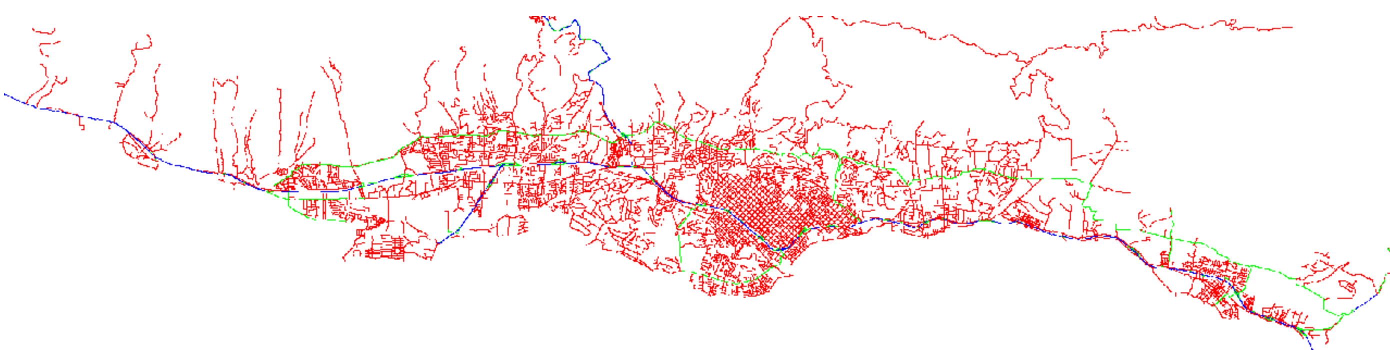
Roads input



1929

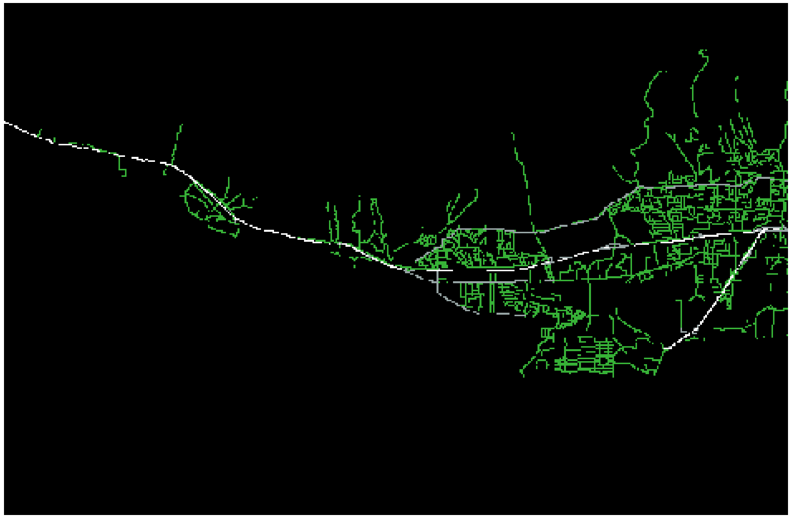


1999

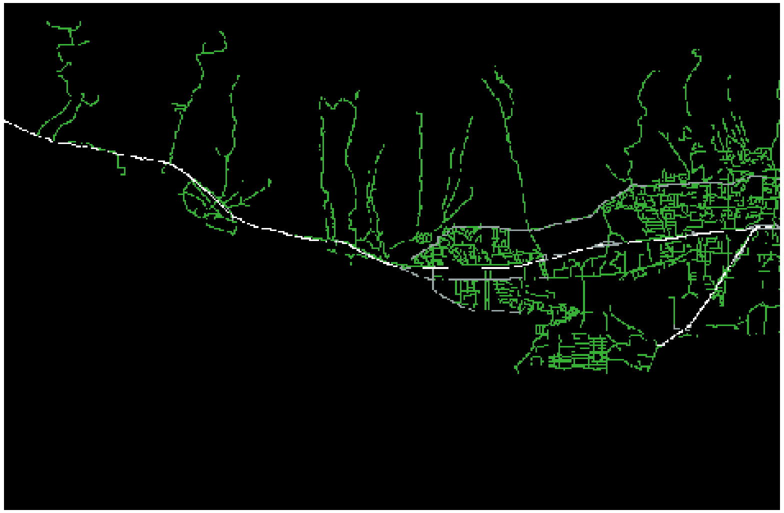


2005

Roads scenarios for 2005

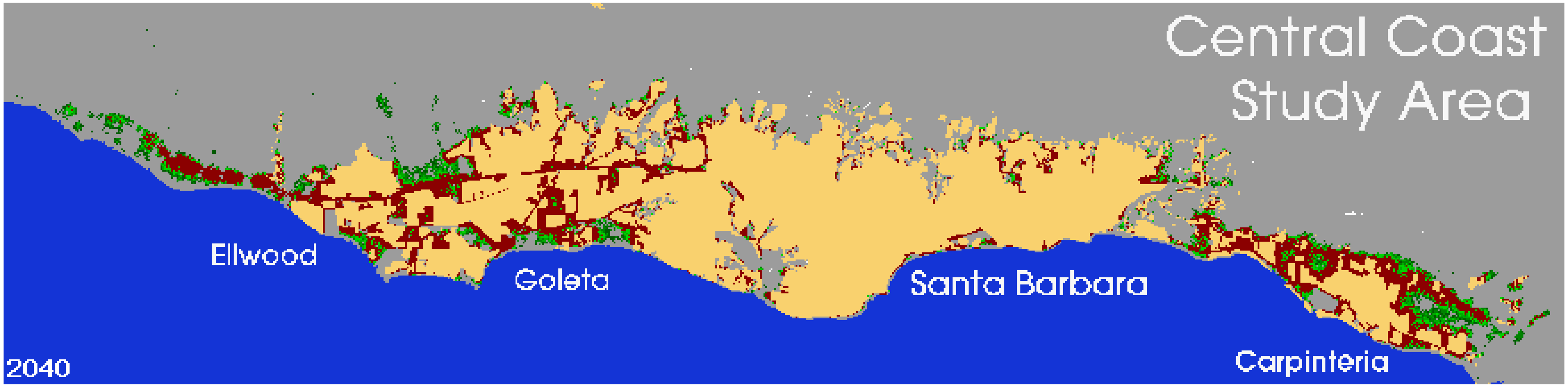


Use current roads

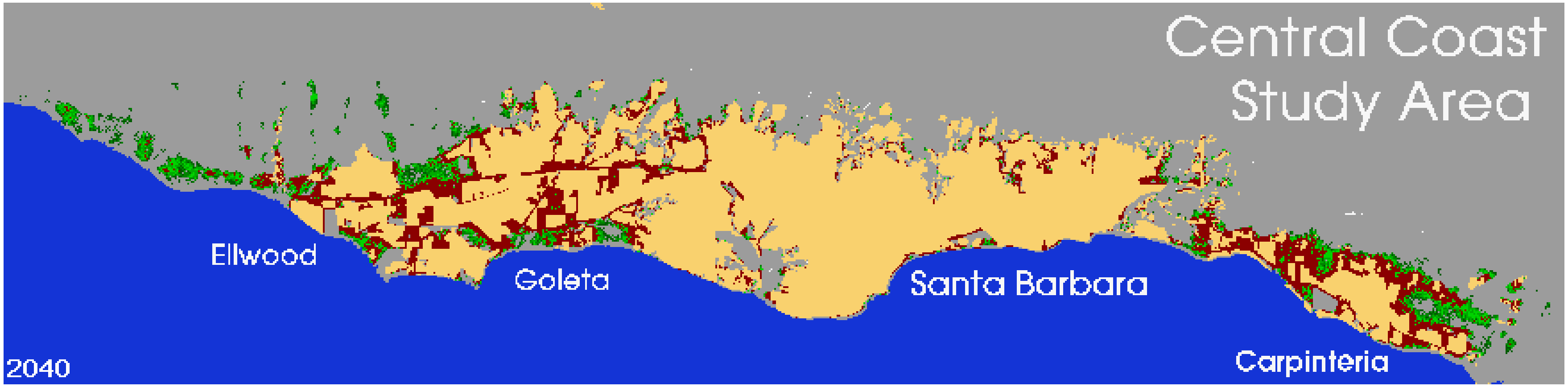


Upgrade all local access roads

Urban growth to 2040



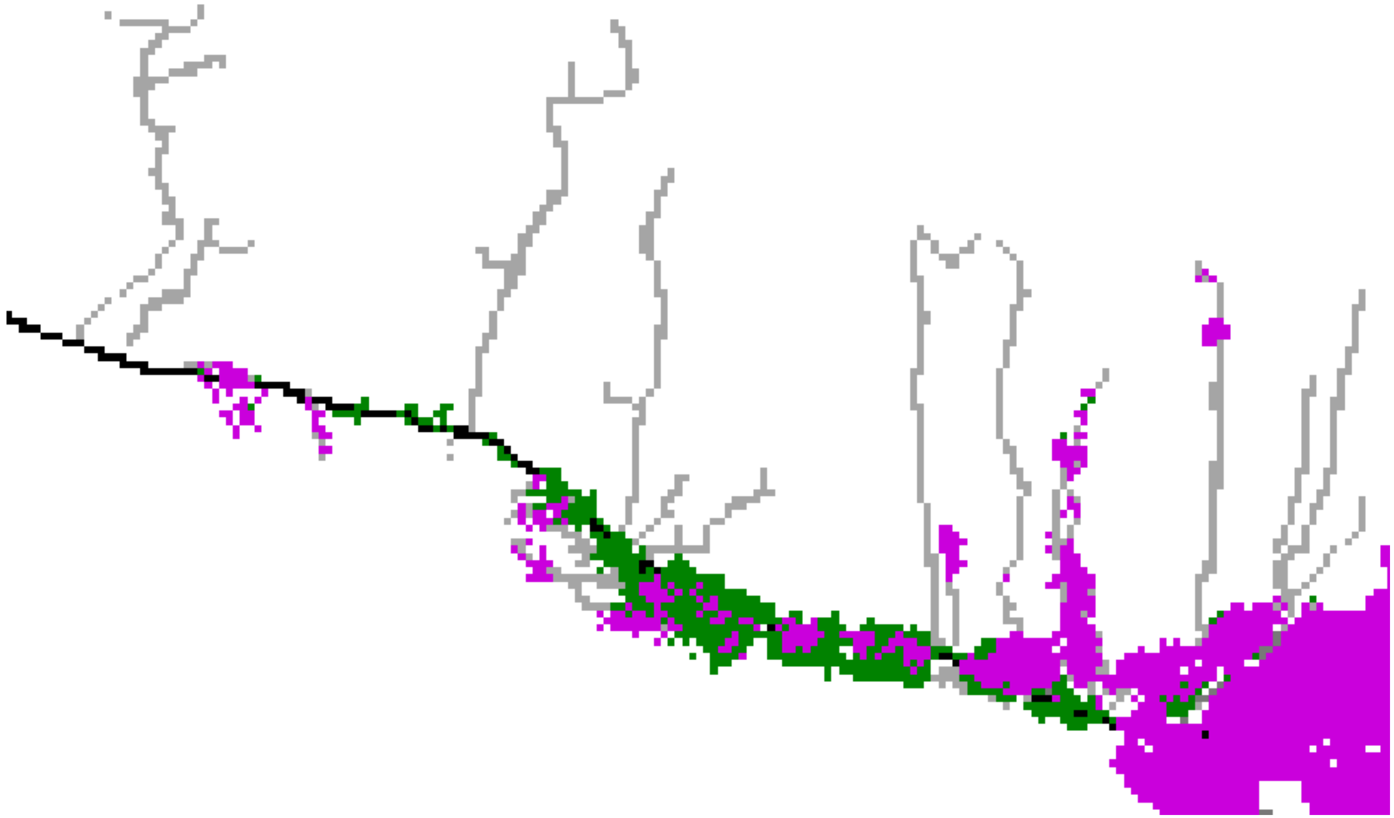
No new roads



Upgrade all local roads

Scenario 2: Upgrade local roads

Scenario Differences



Green: no new roads Magenta: Upgrade local roads

SLEUTH Master Class

- Covered model data needs and functioning
- Covered calibration in 2 different ways
- Examined outputs
- Open Source was key to the model's survivability, modification, use and success
- If "SLEUTH model" was a person, they would have an h-index over 30!