The integration of land change modeling framework FUTURES into GRASS GIS 7

Anna Petrasova, Vaclav Petras, Douglas A. Shoemaker, Monica A. Dorning, Ross K. Meentemeyer

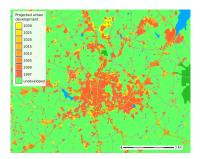
> NCSU OSGeo Research and Education Lab Center for Geospatial Analytics North Carolina State University

> > July 16, 2015

FUTURES

FUTure Urban-Regional Environment Simulation (FUTURES)

- stochastic, patch-based land change model
- simulates urban growth
- model accounts for location, quantity, and pattern of change
- incorporates positive feedbacks (new development attracts more development)
- allows spatial non-stationarity



[Meentemeyer et al., 2013]

FUTURES: Multilevel Simulations of Emerging Urban–Rural Landscape Structure Using a Stochastic Patch-Growing Algorithm

Ross K. Meentemeyer,* Wenwu Tang,* Monica A. Dorning,* John B. Vogler,* Nik J. Cunniffe,[†] and Douglas A. Shoemaker*

> *Department of Geography and Earth Sciences, University of North Carolina at Charlotte [†]Department of Plant Sciences, University of Cambridge

We present a multilevel modeling framework for simulating the emergence of landscape spatial structure in urbanizing regions using a combination of field-based and object-based representations of land change. The FUTure Urban-Regional Environment Simulation (FUTURES) produces regional projections of landscape patterns using coupled submodels that integrate nonstationary drivers of land change: per capita demand, site suitability, and the spatial structure of conversion events. Patches of land change events are simulated as discrete spatial objects using a stochastic region-growing algorithm that aggregates cell-level transitions based on empirical estimation of parameters that control the size, shape, and dispersion of patch growth. At each time step, newly constructed patches reciprocally influence further growth, which agglomerates over time to produce patterns of urban form and landscape fragmentation. Multilevel structure in each submodel allows drivers of land change to vary in space (e.g., by jurisdiction), rather than assuming spatial stationarity across a heterogeneous region. We applied FUTURES to simulate land development dynamics in the rapidly expanding metropolitan region of Charlotte, North Carolina, between 1996 and 2030, and evaluated spatial variation in model outcomes along an urban-rural continuum, including assessments of cell- and patch-based correctness and error. Simulation experiments reveal that changes in per capita land consumption and parameters controlling the distribution of development affect the emergent spatial structure of forests and farmlands with unique and sometimes counterintuitive outcomes. Key Words: fragmentation, land change model, nonstationarity, object-based, region growing algorithm.

The dark side of FUTURES

- no documentation
- know-how limited to certain colleagues
- code not reusable
- different versions of code
- no central storage for code
- extensive manual data preparation
- bad choice of programming language

```
int getRandomNumber()
{
return 4; // chosen by fair dice roll.
// guaranteed to be random.
}
```

https://xkcd.com/221

We hope FUTURES can make broader impact in land change community...

...but more than just releasing the code online is needed.

We can take example from many open source projects which have established ways and technologies for

- collaboration
- code development and distribution
- providing documentation, user support



Integration

Standalone vs. integrated into open source GIS:

- + all standard GIS tools and algorithms available
- + distribution and installation across all platform solved
- + established ways to provide documentation

+ existing user base with

- less flexibility in certain aspects
- need to learn to use the particular GIS



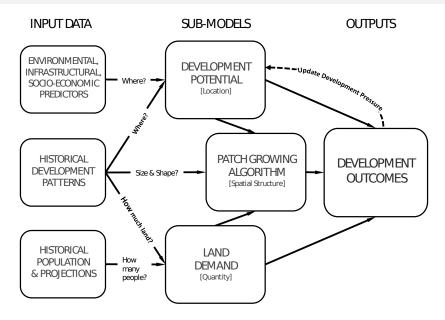
GRASS GIS

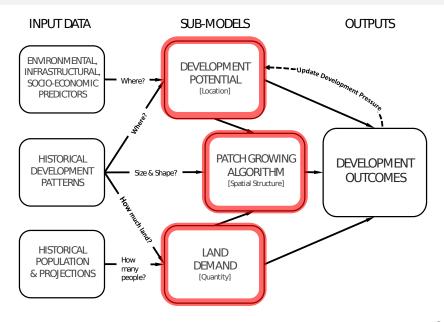
Why integrate FUTURES into GRASS GIS 7?

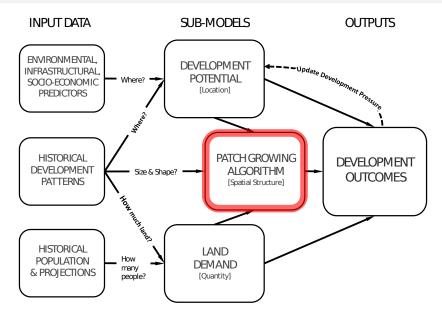
- able to process large datasets
- runs on HPC clusters
- efficient I/O libraries



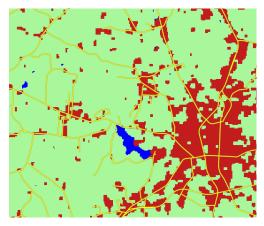
- modular architecture modules in C/C++ and Python, R
- automatically generated CLI and GUI
- addons hosted on OSGeo servers
 - easy installation: > g.extension r.futures
 - daily compiled binaries for Windows (thanks to M. Landa, FCE CTU in Prague)
 - maintained by community and developers (API changes)





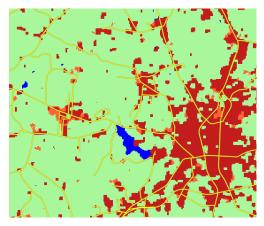


- stochastic algorithm
- converts land in discrete patches



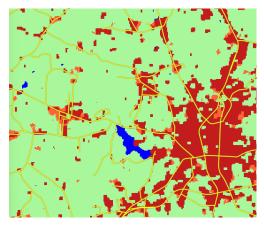
- \blacksquare implemented in C/C++
- computationally demanding
- inputs are patch characteristics (distribution of patch sizes and compactness) derived from historical data

- stochastic algorithm
- converts land in discrete patches



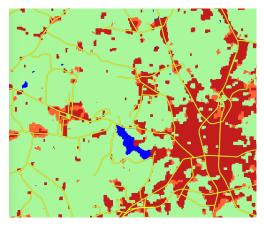
- \blacksquare implemented in C/C++
- computationally demanding
- inputs are patch characteristics (distribution of patch sizes and compactness) derived from historical data

- stochastic algorithm
- converts land in discrete patches



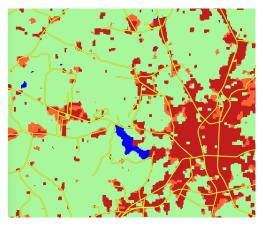
- \blacksquare implemented in C/C++
- computationally demanding
- inputs are patch characteristics (distribution of patch sizes and compactness) derived from historical data

- stochastic algorithm
- converts land in discrete patches

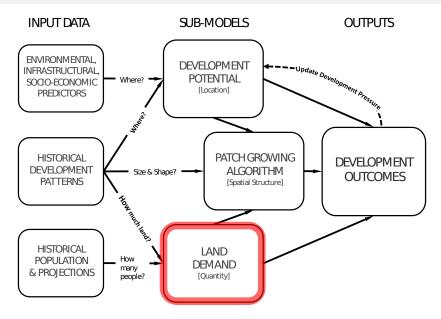


- \blacksquare implemented in C/C++
- computationally demanding
- inputs are patch characteristics (distribution of patch sizes and compactness) derived from historical data

- stochastic algorithm
- converts land in discrete patches

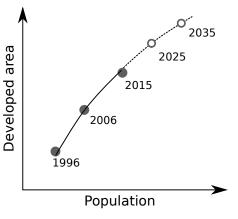


- \blacksquare implemented in C/C++
- computationally demanding
- inputs are patch characteristics (distribution of patch sizes and compactness) derived from historical data

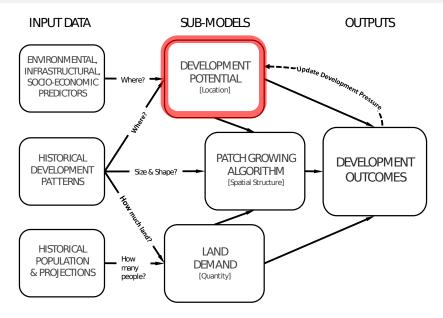


DEMAND submodel

- estimates the rate of per capita land consumption specific to each subregion
- extrapolates between historical changes in population and land conversion

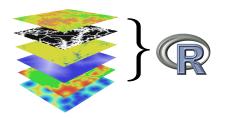


- inputs are historical landuse, population data, population projection
- ordinary least squares regression (linear or logarithmic relationships)
- informally implemented as R scripts and ArcGIS workflows

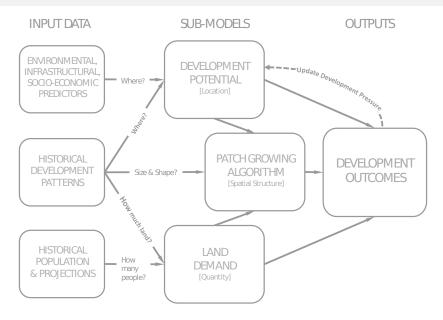


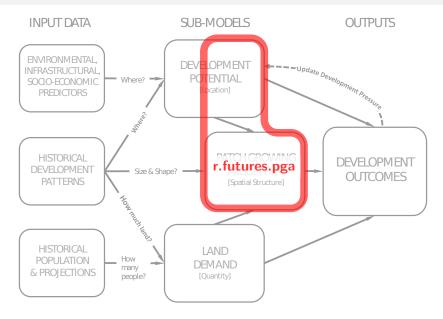
POTENTIAL submodel

- multilevel logistic regression for development suitability
- accounts for variation among subregions (for example policies in different counties)



- inputs are uncorrelated predictors (distance to roads and development, slope, ...)
- original implementation
 - data preparation in ArcGIS
 - regression coefficients derived in R
 - potential and probability surface recomputed after each step in the main C code





- efficient raster reading and writing
- fixed segfaults
- general cleanup (dead code), revised input options

```
r.futures.pga [-s] developed=name predictors=name[,name,...]
demand=name devpot_params=name discount_factor=value
...
incentive_table=name [constrain_weight=name] [random_seed=value]
output=name [output_series=basename] [--overwrite] [--help]
[--verbose] [--quiet] [--ui]
```

- efficient raster reading and writing
- fixed segfaults
- general cleanup (dead code), revised input options

😣 🖨 🐵 r.futures.pga	[raster, patch growing, urban, landscape, modeling]	
Simulates landuse of urban-rural landsca	hange using FUTure Urban-Regional Environment Simulation (FUTURES). Module uses Patch-C se structure development.	Growing Algorithm (PGA) to simula
Basic input	Raster map of developed areas (=1), undeveloped (=0) and excluded (no data):*	(developed=name)
Potential		
Demand	Raster map of subregions with categories starting with 1:*	(subregions=name)
PGA	Number of sub-regions (e.g., counties) to be simulated:*	(num_regions=integer)
Development pressure		(num_regions=inceger)
Scenarios		
Random numbers		
Output		
Optional		
Command output		
O Manual		
	🗱 Close Run 🖷 Copy 🚱 Help	
S Add created map(s) into layer tree		
Close dialog on finish		

r.futures.pga developed=<required> predictors=<required> demand=<required> devpot_params=<required> discount_factor=<required> compactnes

- efficient raster reading and writing
- fixed segfaults
- general cleanup (dead code), revised input options

😣 🖻 🗉 r.futures.pga	[raster, patch growing, urban, landscape, modeling]	
Simulates landuse of urban-rural landsca	hange using FUTure Urban-Regional Environment Simulation (FUTURES). Module uses Patch-Grow pe structure development.	ving Algorithm (PGA) to simula
Basic input	Discount factor of patch size:*	(discount_factor=float)
Potential		
Demand	Mean value of patch compactness to control patch shapes:*	(compactness_mean=float)
PGA	Range of patch compactness to control patch shapes:*	(compactness_range=float)
Development pressure		
Scenarios	The number of neighbors to be used for patch generation (4 or 8):*	(num_neighbors=integer)
Random numbers	The way location of a seed is determined (1: uniform distribution 2: development probability):	(seed search=integer)
Output	2	
Optional	File containing list of patch sizes to use:*	(patch_sizes=name)
Command output		Browse
O Manual	or enter values directly:	
✓ Add created map(s) ○ Close dialog on finist		
crose dialog of this		

r.futures.pga developed=<required> predictors=<required> demand=<required> devpot_params=<required> discount_factor=<required> compactnes

- efficient raster reading and writing
- fixed segfaults
- general cleanup (dead code), revised input options

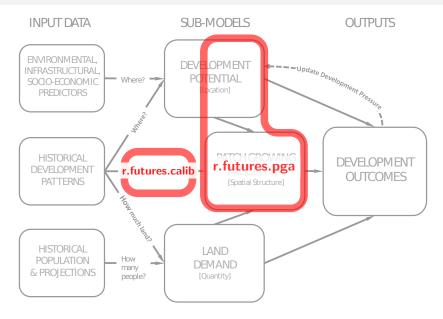
	[raster, patch growing, urban, landscape, modeling]		
Simulates landuse of urban-rural landsca	:hange using FUTure Urban-Regional Environment Simulation (FUTURES). Mod pe structure development.	lule uses Patch-Growing Algorithm (PGA) to simula	
Basic input	Raster map of development pressure:*	(development_pressure=name)	
Potential			
Demand	Size of square used to recalculate development pressure: *	(n_dev_neighbourhood=integer)	
PGA	Approaches to derive development pressure:*	(development pressure approach=string)	
Development pressure	gravity		
Scenarios	Influence of distance between neighboring cells:*	(gamma=float)	
Random numbers			
Output	Scaling factor:*	(scaling_factor=float)	
Optional			
Command output			
O Manual			
	X Close Run Copy @He	łp	
Add created map(s) into laver tree			
Close dialog on finis	-		

r.futures.pga developed=<required> predictors=<required> demand=<required> devpot_params=<required> discount_factor=<required> compactnes

- efficient raster reading and writing
- fixed segfaults
- general cleanup (dead code), revised input options

😣 🗐 🗊 r.futures.pga	[raster, patch growing, urban, landscape, modeling]	
Simulates landuse change using FUTure Urban-Regional Environment Simulation (FUTURES). Module uses Patch-Growing Algorithm (PGA) to simulai urban-rural landscape structure development.		
Basic input	0.1 International Contraction of the second se	
Potential	0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1	
Demand	P	
PGA	Figure: Effect of incentive table on development probability	
Development pressure	Additionally, parameter constrain_weight (raster map from 0 to 1) enables to include policies (such as new regulations or fees) which limit the development in certain areas. The probability surface is simply multiplied by the constrain_weight values, which results in decreased site subability an areas, where the constrain_weight values are lower than 1.	
Scenarios		
Random numbers	Output	
Output	After the simulation ends, raster specified in parameter output is written. If optional parameter output_series is specified, additional output is a series of raster maps for each step. Cells with value 0 represents the initial development, values >= 1 then represent the step in which the cell was developed. Undeveloped cells have value -1.	
Optional	step in which the cell was developed. Undeveloped cells have value -1.	
Command output		
O Manual	Previous	
	Close Run Copy @Help	
🧭 Add created map(s) into layer tree		
Close dialog on finish		

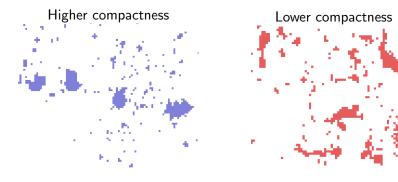
r.futures.pga developed=<required> predictors=<required> demand=<required> devpot_params=<required> discount_factor=<required> compactnes

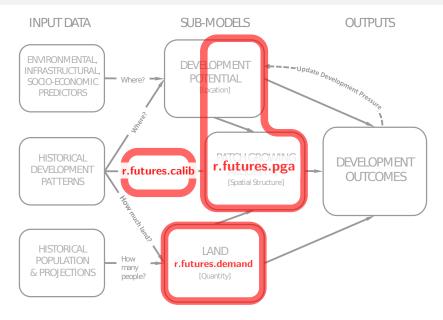


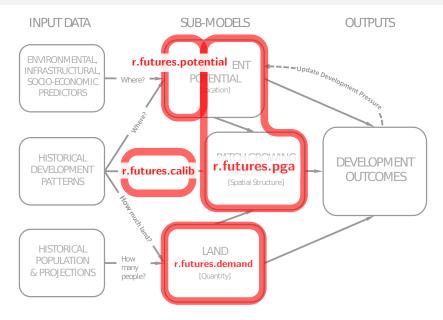
r.futures.calib

New addon written in Python for automated calibration of patch characteristics

- provides optimal patch parameters for r.futures.pga by comparing observed land change pattern with the simulated pattern
- runs in multiple processes





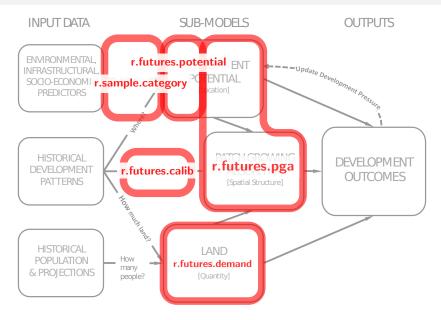


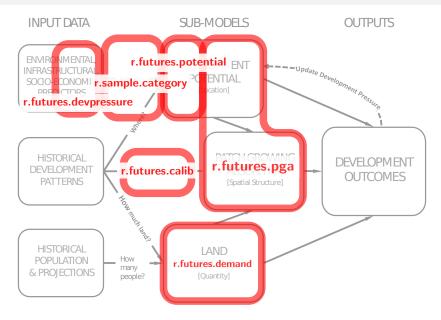
Implementation still needed for

- r.futures.demand Python and Numpy
- r.futures.potential
 - R wrapped in Python to provide standard GRASS interface
 - using dredge from R package MuMIn¹ for automated model selection (read the dredge disclaimer before using)

DEMAND and POTENTIAL statistical models can be implemented in different ways, but we need reference implementation.

¹http://cran.r-project.org/web/packages/MuMIn/MuMIn.pdf



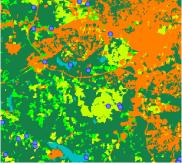


Data preprocessing for POTENTIAL

Two new Python modules needed for FUTURES data preprocessing, but can be used for other applications:

r.sample.category

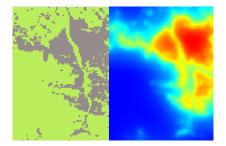
Create sampling points from each category in a raster map

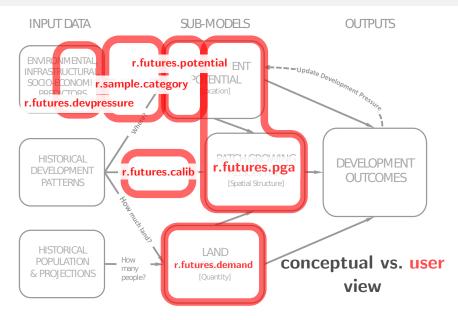


source: r.sample.category manual page

r.futures.devpressure

Moving window computation of distance decay effect





Next steps

Finish r.futures.demans and r.futures.potential More documentation needed:

- tutorial
- sample dataset Raleigh for consistency with GRASS GIS North Carolina sample dataset
 - use publicly available data (National Land Cover Database)



NLCD 2011

- Gather data (historical urban development, population projection, predictors)
- Use a high-end workstation
- Installation is simple, works on all platforms:
 - > g.extension r.futures

Code available online:

trac.osgeo.org/grass/browser/grass-addons/grass7/raster/r.futures/

- Integrate your work into FOSS
- 2 Automate tasks
- 3 Document!

Thank you! Contact: akratoc@ncsu.edu

References

 Meentemeyer, R. K., Tang, W., Dorning, M. A., Vogler, J. B., Cunniffe, N. J., & Shoemaker, D. A. (2013)
 FUTURES: Multilevel Simulations of Emerging Urban–Rural Landscape Structure Using a Stochastic Patch-Growing Algorithm Annals of the Association of American Geographers, 103(4), 785–807.

Dorning, M. A., Koch, J., Shoemaker, D. A., & Meentemeyer, R. K. (2015) Simulating urbanization scenarios reveals tradeoffs between conservation

planning strategies

Landscape and Urban Planning, 136, 28–39.