Development of Python-based ArcGIS Tools for Spatially Balanced Forest Sampling Design

Mingyang Li
Department of Forest Management
Nanjing Forestry University
Nanjing, China, +86-25-84638926
e-mail:lmy196727@126.com

Ting Xu
Department of Forest Management
Nanjing Forestry University
Nanjing, China

Qi Zhou
Department of Forest Management
Nanjing Forestry University
Nanjing, China

***Abstract*—The current forest survey sampling methods are based on classical statistics, can not solve the problems of close spatial autocorrelation and poor adaptability. General randomized tessellation stratified (GRTS), a commonly used algorithm to implement spatial balanced sampling (SBS) has gained popularity since 1997. In this paper, Python was used to make ArcGIS Tools for GRTS, followed by a case study of forest biodiversity computer simulation sampling in Hunan Province. To compare the performance of SBS with simple random sampling, systematic sampling, four index were calculated from three aspects of spatial autocorrelation, sampling efficiency, sampling precision. Research results show that, compared with simple random sampling and systematic sampling, SBS has obviously advantages in reducing spatial autocorrelation and improving sampling efficiency and precision.**

Keywords- spatial balanced sampling (SBS), Python; ArcGIS tool; forest survey

I. INTRODUCTION

Due to the constrictions of labor, transportation and financial conditions, sampling methods including simple random sampling and systematic sampling are commonly used to survey forest resources. Many forest survey factors such as stock volume and elevation are not purely random variables, but variables with dual characteristics of randomness and structure [1]. Since the existing forest resources survey sampling methods are based on the theories of classical statistics which only deals with random variables, there are many defects such as close spatial autocorrelation and poor adaptability when they are used in the study of tree individuals, wild animal populations and spatial heterogeneity of forest community. Therefore, to design a cost-effective sampling method with flexible adaptability and strictly statistical basis has become a hot spot for forest researchers and statisticians.

Spatial balanced sampling (SBS) was firstly suggested by Stevens, a scientist from Department of Statistics, University of Oregon, USA, in 1997 [2].As an innovative solution to the problems of spatial autocorrelation and uncertainly in the research field of natural resources survey, it has been paid more and more attention by natural resources scientists and survey practitioners since then. General randomized tessellation stratified (GRTS), a frequently used algorithm to implement SBS, is gaining popularity as a sampling scheme for large-scale long-term environmental surveys. GRTS samples with reverse hierarchical ordering are designed such that for any sample size, say n, the first n units in the sample will be spatially balanced (i.e., "spread out"). In fact, any contiguous set of n units in a reverse hierarchically ordered GRTS sample constitutes a spatially balanced set of sample units [3]. A spatially balanced GRTS sample makes it easy to both add units in a way that does not compromise spatial balance and to maximize the overlap (co-location) of multiple studies such that all sample sizes are spread out. The ease with which spatially balanced units are added to single or multiple studies is the chief advantage of GRTS samples over the most popular classic designs of systematic sampling and simple random sampling. Another advantage, although rarely realized in practice, of GRTS samples is that they avoid the alignment problems and subsequent adverse effects on estimates that can occur with systematic sampling.

In ArcGIS, Python can be used for coarse-grained programming, meaning that users can use it to easily make geo-processing tools such as the Buffer tool. Certainly, users could code all the buffer logic themselves by using more detailed and fine-grained programming with ArcObjects, but this would be time consuming and unnecessary in most scenarios; it's easier just to call the Buffer tool from a Python script using one line of code [4]. In this paper, we make Python-based ArcGIS tools for probabilistic spatially balanced forest sampling design. There are three parts in the paper: (1) firstly, we briefly describe the process of generalized random tessellation stratified sampling design; (2) then, we described some key Python codes in ArcGIS SBS tools ;(3) finally, we compared the efficiency of SBS with simple random sampling and systematic sampling by using the developed sampling ArcGIS tools through a case study of forest biodiversity survey in Hunan Province.

II. DESIGN AND PROCESS OF GRTS

A. Design of GRTS

Design of GRTS is based on a random function that maps the unit square into the unit interval. The random function is constructed so that it is 1–dimension and preserves some 2-dimensional proximity relationships in the 1-dimensional image. GRTS can accommodate variable sample point density, sample augmentation, and spatially-structured temporal samples. It uses a spatial address based on recursive

The project was sponsored by Chinese National Nature Science Foundation (No.30972298 and No. 31170592)

partitioning to order points in 2-space: first, split a quadrant into similar sub-quadrants, then split each sub-quadrant into sub-sub-quadrants, and so on until an infinite number of points are created. The address of points follows splitting order. Each of the sub-quadrants will be given an address number. The first digit of the number identifies sub-quadrant, while the second digit identifies sub-sub-quadrant, and so on (Fig.1).

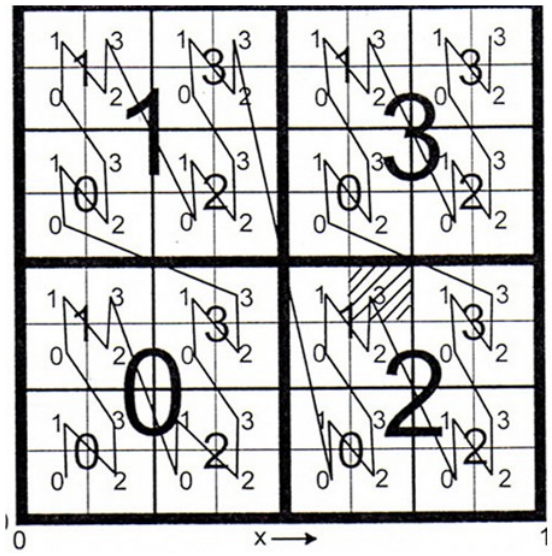

Figure 1. Design of GRTS (Stevens, 1997)

B. *Process of GRTS*

GRTS can sample multiple features including discrete points, liner networks and area polygons. Since the basic steps for different sample feature are almost the same, using area resources as an example, we describe sampling steps of GRTS: ①obtain up-to-date satellite images or aerial photos of survey area and digitalize them to establish a sample frame; ② randomly place a square over the sample frame geographic region; ③ construct a hierarchical grid of quad-tree with hierarchical addressing in the square;④ Peano mapping of two space to one space using hierarchical addressing; ⑤ complete hierarchical randomization of Peano map;⑥ place sample elements on line in one space using hierarchical randomization order, assigning length to elements based on frame and inclusion density;⑦ select systematic sample with random start from line;⑧ place sample in reverse hierarchical order.

III. EXAMPLE PYTHON CODES IN ARCGIS SBS TOOLS

A. *Requirements of the SBS ArcGIS Tool*

As add-in ArcGIS tools, Python-based SBS software should meet some basic requirements of sampling programs. First, the tool should be able to input multiple data of point, line and area. Second, it is capable of dealing with both continuous (gradients) and discrete (strata) inclusion probability. Third, it can visualize sample design on the platform of ArcGIS. The fourth requirement is it can modify inclusion probability based on accessibility constrains .The last requirements is it can develop a map of inferred population.

B. *Example Python Codes in the ArcGIS SBS Tool*

Compared with other high-level programming languages such as VB and VC, users do not need to master many components and functions when they make ArcGIS tools written in Python script. It is through the access to the attributes of Geoprocessor object that Python script processes data. Users can generate Geoprocessor object in two ways. One is to import the COM interface, which can only call the object on Windows platform.

```
import win32com.client # import the win32com.client module

gp=win32com.client.Dispatch ("esriGeoprocessing.GpDispatch.1") # generate a Geoprocessor variable.
```

Another way is to import ArcGISscripting module which is provided in ArcGIS version 9.2. The module can be cross-platform called and is the most commonly used way to generate Geoprocessor object. Following lines are the codes of the second way:

```
import ArcGISscripting

gp = ArcGISscripting. create ( )
```

When variable gp is created, users can access to the properties and methods of the module, such as:

```
gp.Workspace = "c:\ \workspace" # define the work spacegp.Clip (parameter 1, parameter 2, ...) # clipping
gp.Buffer (parameter 1, parameter 2, ... ) # buffer analysis
```

SBS tools work as three step process that includes generating a global sequence raster, filtering the sequence raster against a random raster, and extracting n number of sample points from the filtered raster based on the inclusion probability raster. The sequence raster is a function of a hierarchical quadrant-recursive ordering of space based on Morton order, which fills space with a unique address. The sample points that are extracted from the filtered sequence raster based on a probability are sequenced so that the top n samples from the extracted N samples are random and spatially well balanced. Following lines are the program codes of Python-based SBS ArcGIS tools.

```
# Check out Spatial Analyst extension license

gp.CheckOutExtension("spatial")

if gp.CheckExtension("spatial") == "Available":

  gp.CheckOutExtension("spatial")

else:

raise "LicenseError"

# filter against inclusion probabilities

if InputRandRaster == "#":

strExp = "int(con( " + gFrame + " > rand_ras, " + InSelRas + " ))"

else:
```

```
InRandraster=gp.Describe(InputRandRaster).Path+"//" + gp.Describe(InputRandRaster).Name

strExp = "int(con( " + gFrame + " > " + InRandraster + ", " + InSelRas + " ))"

gp.SingleOutputMapAlgebra_sa(strExp, OutSelRas)

# find a threshold to identify a reasonable number of points -- should provide only

# up to 0.1% (1/1000) of points, otherwise not enough resolution in GRID

pHeight = gp.describe(gFrame).Height

pWidth = gp.describe(gFrame).Width

pNumCells = pHeight * pWidth

gp.AddMessage(" ")

gp.AddMessage("Converting " + gSeqRaster + " to point featureclass")

gp.AddMessage(" ")

gp.RasterToPoint_conversion(gSeqRaster, gp.Workspace + "\\smpts.shp") # export rater to featureclass to be sorted

gp.addfield(gp.Workspace + "\\smpts.shp", "inc", "Double", "9", "9")

gp.CreateFeatureclass(gp.Workspace, OutPtsFC + ".shp", "POINT") #Create blank feature class to be populated with sample point in i order

gp.addfield(OutPtsFC + ".shp", "order", "long", "6")

gp.addfield(OutPtsFC + ".shp", "inc_prob", "Double", "9", "9")
```

IV. PERFORMANCE ANALYSIS OF SPATIAL BALANCED SAMPLING

A. *Spatial Distribution of SBS Sampling Points*

In the paper, Hunan, a major forest province in Southern China was chosen as a case study area, while forest biodiversity was selected as a survey variable. The inclusion probability raster layer of forest biodiversity in study area was established as a sampling frame to conduct performance comparison. Simple random sampling and systematic sampling were implemented by means of an ArcGIS plug-in analysis tool of HawthTools, and SBS was implemented through developed Python-based ArcGIS tools. To compare the performance of three sampling methods, the sample is designed the same size as 331 for each method. After generation of three vector point shape files, SBS points were over-layered with a Thematic Zoning Map of Hunan Province in 12th Five-year Plan to produce a SBS sampling points distribution map of forest biodiversity in Hunan (Fig.2).Then Zonal Statistics function in spatial Analyst tools was used to calculate the number of sample point for each provincial zone.

It can be seen from Fig.2, the number of SBS point for each provincial zone is adaptable to the thematic function of zones. The SBS sampling points located in national eco-zone, provincial eco-zone, national development zone, provincial development zone and crop product zone are 74, 95, 63,39,59, respectively. The total land area of national eco-zone and provincial eco-zone accounts for 47.29% of the total study area, but the number of SBS sampling points accounts for as high as 51.06% of the total sampling number. For national and provincial development zone, the land area is 34.79%, but the number of SBS sample points is only 30.82%. For crop product zone, the land area is 17.90%, while the number of point is 17.82%. Compared to the area ratio of each provincial zone, it can be seen that the relative proportion of sampling points located in the eco-zone with the highest biodiversity is also the highest, the relative proportion for the development with lowest biodiversity is the lowest, and for the crop production zone with moderate biodiversity is medium.

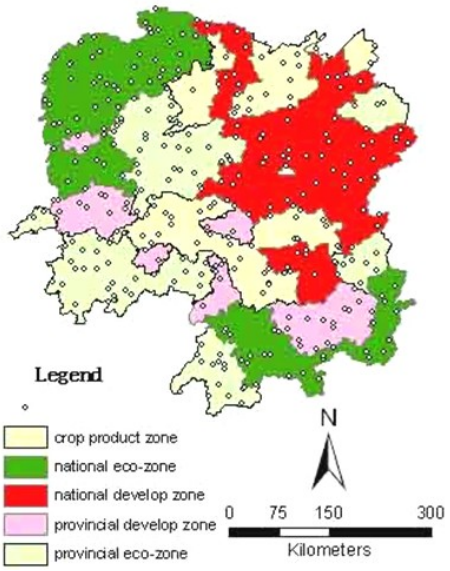

Figure 2. Sampling points of forest biodiversity in Hunan

B. *Performance Aanalysis of SBS with Other Sampling Methods*

Since Normalized Difference Vegetation Index (NDVI) is closely correlated with vegetation cover and biomass, we used Spot Vegetation NDVI in 2010 to evaluate the level of forest biodiversity in study area. To compare the performance of SBS with simple random sampling and systematic sampling, four index were calculated from three aspects of spatial autocorrelation, sampling efficiency, sampling precision. Two index of Moran's *I* and *Z* were selected to analyze the spatial autocorrelation of three sampling methods, while mean NDVI and coefficient of NDVI variation were figured out to evaluate sampling efficiency and sampling precision of three sampling methods, respectively. A positive Moran's *I* index value indicates tendency toward clustering while a negative Moran's *I* index value indicates tendency toward dispersion. *Z* scores are measures of standard deviation [5]. The critical *Z* score values when using a 95% confidence level are -1.96 and +1.96 standard deviations. The p-value associated with a 95% confidence level is 0.05. If *Z* score is between -1.96 and +1.96, p-value will be larger than 0.05, and the pattern exhibited is a pattern that could very likely be one version of a random

pattern. If the Z score falls outside that range (for example -2.5 or +5.4), the pattern exhibited is probably discrete or clustered. Mean NDVI is an index reflecting the level of forest biodiversity indicators, while the coefficient of NDVI variation reflects the degree of fluctuations of the sample points. The greater the coefficient of variation is, the lower the sampling accuracy.

After three vector point shape files were created, each point was intersected with NDVI raster layer to get a NDVI attribute, followed by generation of the four index (Table 1).To reduce the impact of inconsistent spatial distribution of simple random sampling points on the performance, the four index for the random sampling were calculated as the mean values of 10 running.

TABLE I. PERFORMANCE ANALYSIS OF SBS WITH SIMPLE RANDOM AND SYSTEMATIC SAMPLING

Sampling method	Evaluation index			
	Spatial autocorrelation		*Sampling efficiency*	*Sampling precision*
	Moran I	***Z***	***Mean DVI***	***CV NDVI***
Simple random	0.27	9.52	28.882	0.8127
Systematic	0.21	4.46	30.716	0.7550
Spatial balanced	0.22	5.96	30.930	0.7432
Mean	0.23	6.65	30.176	0.7694

It can be seen from Table 1, simple random sampling ranks the last in terms of spatial autocorrelation, sampling efficiency and sampling precision. Though with the lowest spatial autocorrelation, systematic ranks the second in sampling efficiency and sampling precision when it is compared with spatial balanced sampling. Except for the performance of spatial autocorrelation, SBS outperforms other two sampling methods both in sampling efficiency and sampling precision. The reason lies in the fact that SBS combines the advantages of random sampling and systematic sampling, generates spatial evenly distributed sample points, and filters the sampling points with very low inclusion probability.

V. CONCLUSIONS

Many forest survey factors are not purely random variables, but are random and structural variables. The current forest survey sampling methods which are based on classical statistics can not solve the problems of close spatial autocorrelation and poor adaptability when it is used in the study of forest community [6]. Since it was firstly suggested by Stevens in 1997, spatial balanced sampling (SBS), has attracted extensive attention in the research field of natural resources management and field survey practitioners. GRTS is a commonly used algorithm to implement SBS. Because drawing a GRTS sample can be very complicated, add-in ArcGIS sampling software is needed by forest workers and land owners.

Python is an open source, free software, which can be interpreted on cross-platforms. Besides, Python syntax is easy to learn, meaning users do not need to learn a lot of programming knowledge. In ArcGIS, Python can be used for making coarse-grained add-in tools. It should be noticed that Python-based SBS software should meet some basic requirements of sampling programs, such as multiple input of vector data, visualization of sample design on the platform of ArcGIS, modification of inclusion probability.

In the paper, Hunan, a major forest province in Southern China was chosen as a case study area, while forest biodiversity was selected as a survey variable. To compare the performance of SBS with simple random sampling and systematic sampling, four index were calculated from three aspects of spatial autocorrelation, sampling efficiency, sampling precision. Compared with simple random sampling and systematic sampling, SBS has obviously advantages in reducing spatial autocorrelation and improving sampling efficiency and precision.

ACKNOWLEDGMENT

This research was jointly funded by Chinese National Nature Science Foundation (No.30972298) and Chinese National Nature Science Foundation (No. 31170592). We thank postgraduates at Department of Forest Management, Nanjing Forestry University, for their enthusiasm and support.

REFERENCES

[1] M. Y. Li, M. Liu, and M.L.Liu,"GIS -based geo-statistical analysis of forest inventory factors," Journal of Nanjing Forestry University (Natural Science Edition), vol.34, pp. 66–77, December 2010.

[2] D. L. Stevens, "Variable density grid-based sampling designs for continuous spatial populations," Environmetrics, vol.8, pp.164–195, August 1997.

[3] D. L. Stevens and A. R. Olsen,"Spatially balanced sampling of natural resources," Journal of the American Statistical Association, vol.99, pp.262–278, January 2004.

[4] H. B. Peng and H. P. Xiang,"Spatial data batch processing using Python," Geomatics & Spatial Information, vol.34, pp. 81–87, April 2011.

[5] G. Arthur and J. K. Ord, "The analysis of spatial association by use of distance statistics ,"Geographical Analysis, vol.24, pp. 189–206, March 1992.

[6] W. S. Zeng and Y. M. Zhou, "Discussion on main problems and countermeasures to continuous forest inventory and forest management inventory," Central South Forest Inventory and Planning, pp.8–11, December 2003.