Supplementary Material

# Integrating aerial LiDAR and Very-High-Resolution images for urban functional zone mapping

**Siji Sanlang [1], Shisong Cao [1,*], Mingyi Du [1], You Mo [2], Qiang Chen [1] and Wen He [1]**

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 100044, China; 201804010206@stu.bucea.edu.cn (S.S.); dumingyi@bucea.edu.cn (M.D.); chenqiang@bucea.edu.cn (Q.C.); 210813J120023@stu.bucea.edu.cn (W.H.)
[2] China Aero Geophysical Survey and Remote Sensing Center for Natural Resources, Beijing 100083, China; 2173601004@cnu.edu.cn
* Correspondence: caoshisong@bucea.edu.cn

# Content

## Table S1 List of abbreviations

| Abbreviations | Definition |
|---|---|
| **UFZ** | Urban Functional Zone |
| **3D** | three-dimensional |
| **USP** | Urban Structure Parameter |
| **2D** | two-dimensional |
| **VHR** | Very-High-Resolution |
| **LiDAR** | Light Detection And Ranging |
| **MLC** | Multi-Classifier |
| **RF** | Random Forest |
| **KNN** | K-Nearest Neighbor |
| **LDA** | Linear Discriminant Analysis |
| **HSC** | Hierarchical Semantic Cognition |
| **IHSC** | Inverse Hierarchical Semantic Cognition |
| **OA** | Overall Accuracy |
| **POI** | Point of Interest |
| **SVF** | Sky View Factor |
| **BH** | Building Height |
| **SAR** | Street Aspect Ratio |
| **FAR** | Floor Area Ratio |
| **NYITS** | New York City Office of Information Technology Services |
| **NYCDITT** | New York City Department of Information Technology and Telecommunications |
| **DSM** | Digital Surface Model |
| **OSM** | Open Street Map |
| **NNI** | Nearest Neighbor Index |
| **NDVI** | Normalized Difference Vegetation Index |
| **RVI** | Ratio Vegetation Index |
| **DVI** | Difference Vegetation Index |
| **NDWI** | Normalized Difference Water Index |
| **Mean. diff.** | Mean diff. to neighbor |
| **Std. Dev** | Standard Deviation |
| **GLCM** | Gray-Level Co-occurrence Matrix |
| **GLDV** | Grey Level Difference Vector |
| $B_R$ | Red Band |
| $B_B$ | Green Band |
| $B_G$ | Blue Band |
| $B_{NIR}$ | Near-Infrared Band |
| **GI** | Gini Index |
| **RB** | Relative Border |
| **DG** | Distance to Grass |
| **DB** | Distance to Building |
| **H** | Height |
| **PA** | Producer's Accuracy |
| **UA** | User's Accuracy |
| **BC** | Building Coverage |
| **TC** | Tree Coverage |
| **GC** | Grass Coverage |

| | |
|---|---|
| **SC** | Soil Coverage |
| **ISC_G** | Impervious Surface Coverage at Ground level |
| **WC** | Water Coverage |
| **BNNI** | Building Nearest Neighbor Index |
| **TNNI** | Tree Nearest Neighbor Index |
| **GNNI** | Grass Nearest Neighbor Index |
| **SNNI** | Soil Nearest Neighbor Index |
| **INNI** | Impervious Ground Nearest Neighbor Index |
| **WNNI** | Water Nearest Neighbor Index |
| **Exp.** | Experiment |

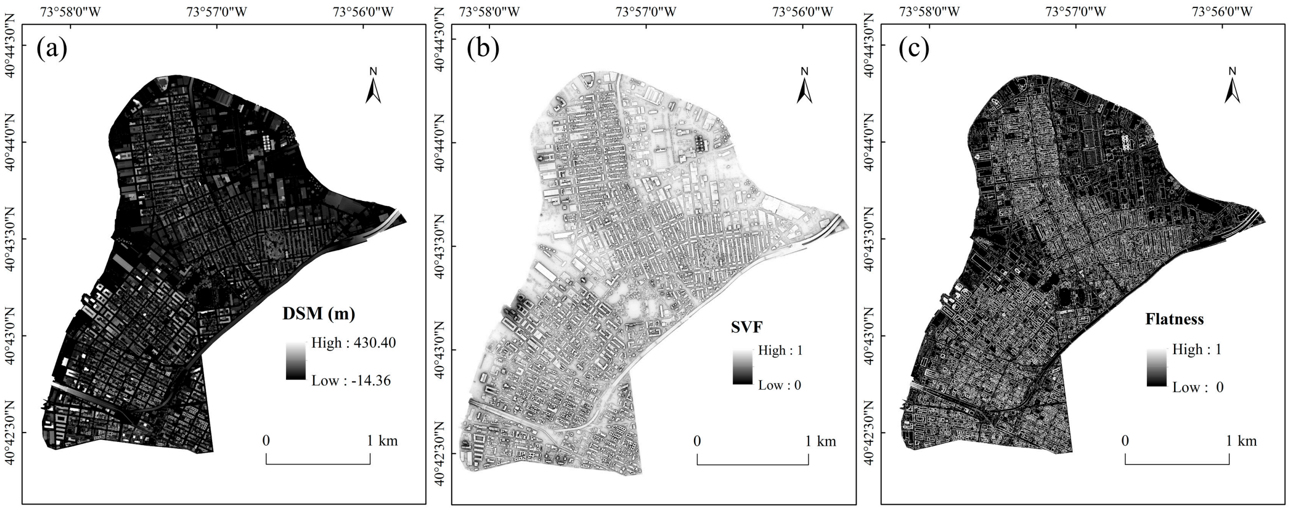


**Figure S1 3D USPs used for land cover mapping.**
**(a) Digital Surface Model (DSM), (b) Sky View Factor (SVF) and (c) Flatness.**

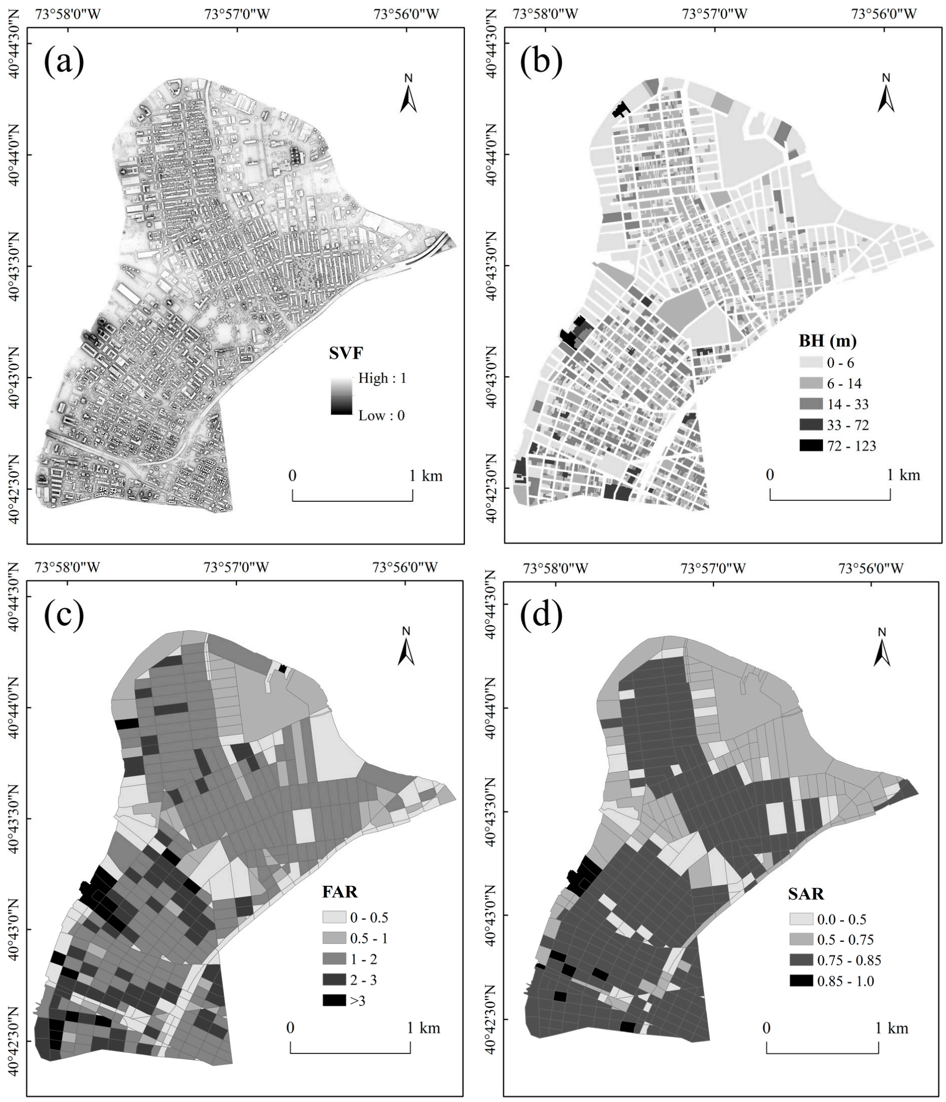


**Figure S2 3D USPs used for UFZ mapping.**
**(a) Sky View Factor (SVF), (b) Building Height (BH), (c) Floor Area Ratio (FAR) and (b) Street Aspect Ratio (SAR).**

**Table S2 Selected optimal variables for land cover mapping.**

| Category | Optimal feature | Number |
|---|---|---|
| Spectral feature | $B_G$, $B_B$, $B_{NIR}$, NDVI, RNI, DVI, NDWI, $Mean_R$, $Mean_G$, $Mean_B$, $Mean_{NIR}$, Brightness, Ratio, Mean. diff., Std. $Dev_{NIR}$ | 15 |
| Textural feature | $Variance_{GLCM}$, $Correlation_{GLCM}$, $Energy_{GLCM}$ | 3 |
| Geometrical feature | Border Length, Length/Width, Main Direction | 3 |
| 3D USP | DSM, SVF, Flatness | 3 |

**Table S3 Confusion matrix of the land covers using RF algorithm.**

| Category | Building | Tree | Grass | Soil | Impervious ground | Water | UA (%) |
|---|---|---|---|---|---|---|---|
| Building | 1177 | 3 | 0 | 0 | 147 | 2 | 88.6 |
| Tree | 5 | 334 | 36 | 0 | 2 | 0 | 88.6 |
| Grass | 2 | 68 | 285 | 0 | 8 | 0 | 78.5 |
| soil | 14 | 0 | 1 | 189 | 21 | 0 | 84.0 |
| Impervious ground | 180 | 2 | 7 | 35 | 1675 | 3 | 88.1 |
| Water | 1 | 0 | 0 | 0 | 1 | 63 | 96.9 |
| PA (%) | 85.4 | 82.1 | 86.6 | 84.4 | 90.3 | 92.6 | 87.4 (OA) |

**Table S4 Confusion matrix of the land covers using LDA algorithm.**

| Category | Build-ing | Tree | Grass | Soil | Impervious ground | Water | UA (%) |
|---|---|---|---|---|---|---|---|
| Building | 1032 | 4 | 0 | 5 | 479 | 3 | 67.8 |
| Tree | 26 | 296 | 56 | 0 | 6 | 0 | 77.1 |
| Grass | 1 | 73 | 262 | 0 | 1 | 0 | 77.7 |
| soil | 0 | 0 | 8 | 161 | 16 | 2 | 86.1 |
| Impervi-ous ground | 317 | 34 | 3 | 57 | 1352 | 12 | 76.2 |
| Water | 3 | 0 | 0 | 1 | 0 | 51 | 92.7 |
| PA (%) | 74.8 | 72.7 | 79.6 | 71.9 | 72.9 | 75.0 | 74.0 (OA) |

**Table S5 Confusion matrix of the land covers using KNN algorithm.**

| Category | Building | Tree | Grass | Soil | Impervious ground | Water | UA (%) |
|---|---|---|---|---|---|---|---|
| Building | 1036 | 3 | 0 | 0 | 355 | 0 | 74.3 |
| Tree | 18 | 305 | 59 | 0 | 9 | 0 | 78.0 |
| Grass | 4 | 77 | 264 | 0 | 3 | 0 | 75.4 |
| soil | 0 | 1 | 6 | 178 | 32 | 0 | 82.0 |
| Impervious ground | 320 | 21 | 0 | 45 | 1455 | 8 | 78.7 |
| Water | 1 | 0 | 0 | 1 | 0 | 61 | 96.7 |
| PA (%) | 75.1 | 74.9 | 80.2 | 79.5 | 78.5 | 86.8 | 77.4 (OA) |

**Table S6 Confusion matrix of land cover classification using RF algorithm (only using 2D features).**

| Category | Building | Tree | Grass | Soil | Impervious ground | Water | UA (%) |
|---|---|---|---|---|---|---|---|
| Building | 1112 | 1 | 0 | 4 | 169 | 2 | 86.3 |
| Tree | 13 | 306 | 55 | 0 | 6 | 0 | 80.5 |
| Grass | 0 | 93 | 268 | 0 | 2 | 0 | 73.8 |
| Soil | 18 | 0 | 5 | 183 | 16 | 0 | 82.4 |
| Impervious ground | 235 | 7 | 1 | 37 | 1659 | 4 | 85.4 |
| Water | 1 | 0 | 0 | 0 | 2 | 62 | 95.4 |
| PA (%) | 80.6 | 75.2 | 81.5 | 81.7 | 89.5 | 91.2 | 84.3 (OA) |

**Table S7 Confusion matrix of land cover classification using KNN algorithm (only using 2D features).**

| Category | Building | Tree | Grass | Soil | Impervious ground | Water | UA (%) |
|---|---|---|---|---|---|---|---|
| Building | 1014 | 0 | 0 | 0 | 408 | 4 | 71.1 |
| Tree | 14 | 274 | 65 | 0 | 1 | 1 | 77.2 |
| Grass | 1 | 128 | 256 | 0 | 0 | 0 | 66.5 |
| Soil | 0 | 2 | 5 | 174 | 18 | 0 | 87.4 |
| Impervious ground | 346 | 3 | 3 | 50 | 1424 | 7 | 77.7 |
| Water | 4 | 0 | 0 | 0 | 3 | 56 | 89.1 |
| PA (%) | 73.5 | 67.3 | 77.8 | 77.7 | 76.8 | 83.8 | 75.1 (OA) |

**Table S8 Confusion matrix of land cover classification using LDA algorithm (only using 2D features).**

| Category | Building | Tree | Grass | Soil | Impervious ground | Water | UA (%) |
|---|---|---|---|---|---|---|---|
| Building | 984 | 0 | 0 | 1 | 487 | 4 | 66.6 |
| Tree | 13 | 273 | 65 | 0 | 26 | 1 | 72.2 |
| Grass | 0 | 97 | 251 | 0 | 0 | 0 | 72.1 |
| Soil | 0 | 1 | 9 | 163 | 47 | 0 | 72.9 |
| Impervious ground | 380 | 23 | 4 | 60 | 1293 | 16 | 72.6 |
| Water | 2 | 13 | 0 | 0 | 1 | 47 | 75.8 |
| PA (%) | 71.4 | 67.1 | 76.3 | 68.3 | 69.7 | 73.5 | 70.5 (OA) |

**Table S9 Selected optimal variables for UFZ mapping.**

| Category | Optimal feature | Number |
|---|---|---|
| 2D USP | BC, TC, GC, SC, ISC_G | 5 |
| 3D USP | SVF, BH, SAR, FAR | 4 |
| Spatial pattern | BNNI, TNNI, GNNI, SNNI, INNI | 5 |

**Table S10 Confusion matrix of UFZ classification using RF algorithm.**

| Category | Commercial zone | Residential zone | Industrial zone | Parks | UA (%) |
|---|---|---|---|---|---|
| Commercial zone | 26 | 3 | 3 | 0 | 81.3 |
| Residential zone | 3 | 87 | 0 | 0 | 96.7 |
| Industrial zone | 0 | 2 | 44 | 2 | 91.7 |
| Park | 0 | 0 | 2 | 14 | 87.5 |
| PA (%) | 89.7 | 94.6 | 89.8 | 87.5 | 91.9 (OA) |

**Table S11 Confusion matrix of UFZ classification using KNN algorithm.**

| Category | Commercial zone | Residential zone | Industrial zone | Park | UA (%) |
|---|---|---|---|---|---|
| Commercial zone | 21 | 11 | 2 | 0 | 61.8 |
| Residential zone | 4 | 73 | 8 | 1 | 84.9 |
| Industrial zone | 4 | 7 | 36 | 4 | 70.6 |
| Park | 0 | 1 | 3 | 11 | 73.3 |
| PA (%) | 72.4 | 79.3 | 73.5 | 68.8 | 75.8 (OA) |

**Table S12 Confusion matrix of UFZ classification using LDA algorithm.**

| Category | Commercial zone | Residential zone | Industrial zone | Park | UA (%) |
|---|---|---|---|---|---|
| Commercial zone | 22 | 7 | 1 | 0 | 73.3 |
| Residential zone | 5 | 82 | 4 | 1 | 89.1 |
| Industrial zone | 2 | 3 | 42 | 3 | 84.0 |
| Park | 0 | 0 | 2 | 12 | 85.7 |
| PA (%) | 75.9 | 89.1 | 85.7 | 75.0 | 84.9 (OA) |

**Table S13 UFZs classification accuracy of different variables combination.**

| Category | Exp. a (%) | Exp. b (%) | Exp. c (%) | Exp. d (%) | Exp. e (%) | Exp. f (%) | Exp. g (%) |
|---|---|---|---|---|---|---|---|
| Commercial zone | 63.6 | 54.5 | 45.5 | 72.7 | 81.8 | 54.5 | 89.7 |
| Residential zone | 86.5 | 78.8 | 73.1 | 94.2 | 90.4 | 84.6 | 94.6 |
| Industrial zone | 77.8 | 70.4 | 70.4 | 85.2 | 85.2 | 70.4 | 89.8 |
| Park zone | 88.9 | 66.7 | 55.6 | 77.8 | 66.7 | 55.6 | 87.5 |
| OA | 81.8 | 72.7 | 67.7 | 87.9 | 85.9 | 74.7 | 91.9 |

## Software codes (Python platform)

```
from sklearn.ensemble import RandomForestClassifier
import osgeo
from osgeo import gdal
from osgeo import osr
from osgeo import ogr

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # a library for plotting
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_boston
from sklearn.ensemble import RandomForestRegressor
from sklearn.inspection import permutation_importance

def readTif(fileName):
  dataset = gdal.Open(fileName)
  if dataset == None:
    print(fileName+"File cannot be opened")
    return
  im_width = dataset.RasterXSize
  im_height = dataset.RasterYSize
  im_bands = dataset.RasterCount
  im_data = dataset.ReadAsArray(0,0,im_width,im_height)
  im_geotrans = dataset.GetGeoTransform()
  im_proj = dataset.GetProjection()
  return im_width, im_height, im_bands,im_data,im_geotrans,im_proj


def writeTiff(im_data,im_width,im_height,im_bands,im_geotrans,im_proj,path):
    if 'int8' in im_data.dtype.name:
        datatype = gdal.GDT_Byte
    elif 'int16' in im_data.dtype.name:
        datatype = gdal.GDT_UInt16
    else:
        datatype = gdal.GDT_Float32

    if len(im_data.shape) == 3:
        im_bands, im_height, im_width = im_data.shape
    elif len(im_data.shape) == 2:
        im_data = np.array([im_data])
    else:
        im_bands, (im_height, im_width) = 1,im_data.shape
            driver = gdal.GetDriverByName("GTiff")
    dataset = driver.Create(path, im_width, im_height, im_bands, datatype)
    if(dataset!= None):
        dataset.SetGeoTransform(im_geotrans)
```

```
        dataset.SetProjection(im_proj)
    for i in range(im_bands):
        dataset.GetRasterBand(i+1).WriteArray(im_data[i])
    del dataset



work_path = 'D:/Geo/all_feature'          # Define your work path
sample_path = work_path + '/' + 'features.csv'

sample = np.array(pd.read_csv(sample_path, dtype='float32'))
X_sample = sample[:,1:44]
Y_sample = sample[:,0]

x_train,x_test,y_train,y_test=train_test_split(X_sample,Y_sample,test_size=0.2,random_state=40)

# Parameters
num_trees = 100
max_depth = None        # If none, nodes are expanded until all leaves are pure.
max_features = 'sqrt'      # then max_features=sqrt(n_features)

rf = RandomForestClassifier(n_estimators=num_trees, max_depth=max_depth,
max_features=max_features)

# Fit the model
rf.fit(x_train, y_train)

# Check the importance
print(rf.feature_importances_)
sorted_idx = rf.feature_importances_.argsort()
plt.barh(range(x_train.shape[1]),rf.feature_importances_[sorted_idx])
plt.xlabel("Random Forest Feature Importance")
plt.show()

# Prediction of calibration dataset
y_pred_cali = rf.predict(x_train)
CM = confusion_matrix(y_pred_cali,y_train)
acc = np.trace(CM) / len(y_train)
print(CM, 'overall accuracy for cali: ', acc)

# Prediction of valibration dataset
y_pred_vali = rf.predict(x_test)
CM = confusion_matrix(y_pred_vali,y_test)
acc = np.trace(CM) / len(y_test)
print(CM, 'overall accuracy for vali: ', acc)
print("complete")
```