Locally rendered high-resolution Land Use/Land Cover Bitmaps from OpenStreetMap Data for geospecific 3D Simulation

Dirk Frommholz

DLR Institute of Optical Sensor Systems, Berlin, Germany - dirk.frommholz@dlr.de

Keywords: Land Use/Land Cover, OpenStreetMap, Rendering, CORINE, Open Source, VBS4.

Abstract

This paper describes a method for creating land use/land cover (LULC) bitmaps from OpenStreetMap data for virtually any part of the world at any resolution on a standard personal computer. The automatic workflow utilizes free software and open formats, does not require web services nor graphical GIS applications and can be run by novice cartographers. In the process, offline vector data sets from the collaborative mapping project are prepared into self-contained transferable database files that arrange relevant geometric features and attributes as spatially indexed tables for fast access. The database gets rendered into a georeferenced bitmap within minutes by the same engine as for the original OpenStreetMap tiles. In contrast to digital maps, the appearance of the output is altered with XML-based style sheets implementing filter rules to picture land cover classes according to various nomenclatures such as CORINE and Virtual Battlespace 4 (VBS4). For linear features like roads and railway tracks lacking a second dimension, the width is scaled to match real-world proportions. The scale factor passed to the style sheets is determined from the extent of the region to be rendered, the requested ground sampling distance, database hints and the latitude of the conformal target projection. Results for scenes around the world are visually analyzed, and a comparison to validated LULC data is conducted. As a showcase targeting security agencies, highly detailed land cover is integrated into a scene for the VBS4 3D simulator to model geospecific landscapes in real time.

1. Introduction

Land use and land cover (LULC) maps constitute a classic product obtained using remote sensing. As thematic layers, they combine information about the physical material of the Earth's surface and a contextual interpretation on its utilization by people (Fisher et al., 2005). Well-known downstream applications of LULC data include urban planning, land zoning and the modeling of natural hazards such as floods and wildfires. Land cover information thus can help improve the quality of life in urban and rural areas, adequately dimension infrastructure, precisely allocate resources to emergency services and determine insurability. When carried out over time, LULC mapping enables environmental monitoring to detect and quantify deforestation, loss of arable land and soil sealing. Large-scale temporal data analysis could be used to validate and predict the positive and negative effects of global warming. Because of the socioeconomic relevance of the subjects listed, land use and land cover layers are considered one of the Global Fundamental Geospatial Data Themes by the United Nations Committee of Experts on Global Geospatial Information Management (UN-GGIM) (UN-GGIM, 2019). Less prominently, LULC maps are also used by current 3D graphics engines for entertaining and serious computer games like the X-Plane flight simulator (Laminar Research, 2024) and Virtual Battlespace 4 (VBS4) (Bohemia Interactive Simulations, 2022). In this domain, they provide a semantic layer to auto-generate geospecific vegetation and other surface elements in real time, i.e., landscapes with a realistic look and feel specific to the visualized geographic area.

LULC data typically comes as two-dimensional geographically referenced ortho-projected rasters. This is because it is frequently derived from spatially congruent multi- or hyperspectral orthophotos generated from aerial or satellite bitmaps on which (semi-)manual tagging or automatic image analysis are performed. However, LULC data may also be stored as scalable semantically annotated 2D or 3D vector shapes. In both cases, unique colors or labels are used for the underlying primitives such as pixels or polygons to represent different classes of ground cover like grass, concrete and water or the related uses like meadows, buildings or harbors. There are no generally accepted conventions on the number of classes, their definition and labeling, which greatly vary among applications that utilize LULC layers. Standardization has been attempted for instance by the European Commission with the CORINE land cover (CLC) nomenclature (5, 15 and 44 classes in levels 1, 2 and 3) (Kosztra et al., 2019), by the Food and Agriculture Organization (FAO) of the United Nations with its Land Cover Classification System (LCCS) (Di Gregorio and Jansen, 1998) and many other institutions. In this regard, the LCCS is special since it defines 8 dichotomous land cover categories which can be combined with hierarchically organized environmental and technical attributes to obtain a flexible set of domain-specific LULC class designations.

2. Related Work and Motivation

There is a huge number of publications on the creation of thematic LULC maps from remote sensing data due to their wide range of applications (Wang et al., 2023). An example for a very recent large-scale land cover data set published under an open data license is WorldCover from the European Space Agency (ESA) (Zanaga et al., 2022). WorldCover was primarily derived from multichannel orthophotos of the Sentinel-1/2 satellites of the Copernicus Earth observation program initiated by the European Union. Data processing builds on supervised learning with a gradient boosting decision tree algorithm using training data that comprises reflectance values, vegetation indices, elevation data and meteorological features. The generated raster images with 11 classes according to the LCCS cover the entire earth at a ground sampling distance (GSD) of

10 meters per pixel (m/pix) (Kerchove et al., 2022). Because of these properties, WorldCover data may already be sufficient to dynamically generate the landscape in flight simulators given the distant perspective of the pilot. On a smaller scale, the work of (Zhang et al., 2022) demonstrates fine-detail land cover extraction in urban environments from aerial photos to produce the openly accessible UrbanWatch database covering 22 North American cities. Semi-supervised machine learning with two neural networks in a teacher-student setup is performed to specifically address perspective view angles, class variation and other challenges coming with very high resolution input data. The resulting model was trained on labeled images from the U.S. National Agriculture Imagery Program (NAIP). Inference yielded LULC bitmaps with 9 different categories at a GSD of 1 m/pix and high positional accuracy, which in practice will be adequate for ground-based serious gaming applications like VBS4. However, the small number of classes limits the variety of landscape types that can be rendered. Moreover, for other target areas, the availability of high-detail input imagery will be a decisive factor for the usability of the proposed method.

In addition to classifying remote sensing data directly, researchers have also investigated how LULC maps could be derived from the large amount of geographic information collected by volunteers around the world. Abolishing fixed ground sampling distances, (Schultz et al., 2017) created global LULC information mainly from classic vector maps from the OpenStreetMap (OSM) project (OpenStreetMap contributors, 2024a). The OSM project aims at providing a comprehensive freely available world map through open collaboration. For their website and public Web Map Service (WMS), the authors have relabeled geometric primitives stored in the OSM database to distinguish between 14 surface classes resembling CLC level 2. Gaps in the data got filled mostly for Europe from Sentinel-2 imagery at a GSD of 10 m/pix to which a machine learning model was applied for classification. In these areas, resampling artifacts remain present due to the used raster vectorization and composition techniques. If OSM geometries are available, the result can be scaled to any target resolution. At present, there is no integrated download option, and land cover must be stored locally using WMS-enabled GIS tools. Likewise, in (Fonte et al., 2017), LULC vector shapes are extracted from OpenStreetMap data according to the CLC level 2 and Urban Atlas nomenclatures. The conversion process considers user-defined restrictions such as area thresholds, topological relations and priorities to handle region overlaps when the semantic tags of the input geometries are assigned target classes. Linear OSM elements like for infrastructure are turned into polygons depending on their type, category and reasonable assumptions on their dimensions. The collaboratively collected width attributes do not get exploited. Since the workflow does not handle unmapped regions, it is limited to urban areas with a good input data coverage. The used toolchain comprises open source software only except for the controlling Python code. However, the replication of the conversion, its adaptation to different class encodings and the ordered rasterization of the vector output may require experienced staff due to the tight integration with GIS and database systems. In (Johnson et al., 2022), to semantically annotate Sentinel-2 imagery for training AI-based land cover extraction algorithms, it is suggested to initially obtain LULC maps from OpenStreetMap data and store them as three-channel bitmaps. For each bitmap pixel, a manually generated ontology maps the OSM feature tags to labels representing one of 15 custom land cover categories. Labels are split among the channels separating mutually exclusive classes to model spatial inclusion relationships, and the ontology provides a mechanism to prioritize overlapping geometries. The chosen output data layout enables the conversion into images of different LULC nomenclatures using pixel math. However, due to the small number of classes and the selected GSD of 10 m/pix like for the satellite data, the coarse multichannel bitmaps will impact realism in first-person 3D simulation applications.

This paper picks up the idea of resolution independence and describes an automatic workflow that turns OpenStreetMap vector maps into geographically referenced LULC rasters within minutes. Coverage is global, and different nomenclatures as well as arbitrary target GSDs are supported. The process is controlled through scripts which invoke unmodified free software tools mainly from the Geospatial Data Abstraction Library (GDAL) (GDAL/OGR contributors, 2024). Any intermediate results and the final product are stored using open data formats. Unlike in previous work, no web services, complex database installations nor graphical GIS applications will be involved that might impose a hurdle to inexperienced cartographers and students. Instead, all data is kept in files and converted locally so that inputs and outputs can be easily copied to another computer even when there is no network connection.

In the workflow, freely available serializations from the OpenStreetMap data set are initially prepared into a self-contained geographic database stored as an OGC GeoPackage (GPKG) (Open Geospatial Consortium, 2024). The database file is used to preselect and reorganize the originally unordered geometric primitives and their semantic attributes into spatially indexed tables for faster access. Gaps in the data are supplemented by a morphologically filtered, reduced and vectorized version of the current WorldCover LULC product. The WorldCover polygons form a coarse but scalable base layer for land areas including those outside Europe. Missing large water bodies like oceans are added from the dedicated OSM water polygon shape file (OpenStreetMap contributors, 2025). The combined all-vector database gets rendered into a georeferenced bitmap by the locally installed Mapnik toolkit (Pavlenko, 2024) that is also used to generate standard OSM tiles. Unlike GIS, Mapnik has a small memory footprint, exposes detailed rasterization settings and supports expressions evaluated at runtime. Rendering is controlled via the extent setting passed as a longitude/latitude pair following OSM data set conventions and the desired GSD given in meters per pixel. The appearance of the output can be adapted with XML-based style sheets (Ray, 2001) which currently support most of CLC level 3 and the VBS4 land cover categories. Also, the stroke width of linear elements like for infrastructure gets scaled to fit their real-world proportions when the respective scale factor is passed to the style sheets. This option will be particularly relevant for 3D simulation and is demonstrated for the conformal Web Mercator target projection.

For the qualitative evaluation of the produced thematic maps, test samples from various locations are inspected visually and checked pixel-wise against spatially aligned references of existing validated high-resolution LULC data after label harmonization. As a showcase, VBS4-encoded land cover is integrated into a virtual 3D scene to prove its suitability for the simulation software used by security agencies.

3. Workflow Description

Local production of georeferenced land use/land cover bitmaps is subdivided into three stages as shown in figure 1. The initial

https://doi.org/10.5194/isprs-annals-X-G-2025-269-2025 |

area database combination stage fuses an OpenStreetMap serialization that contains the scene to be rendered with the coarse WorldCover base layer and the global OSM water polygons into a GeoPackage using the GDAL ogr2ogr vector processor. In the LULC render stage, the result is passed to the mapnik-render command line utility from the Mapnik toolkit. This program needs to be configured with an XML-based style sheet and the geographic extent of the desired area. The style sheet defines the mapping of OSM primitives to land cover classes and is passed the GSD, the scale factor for line features and the target coordinate reference system (CRS). The output is a PNG image to be turned into a GeoTIFF bitmap (or any other location-aware raster format) in the georeferencing stage. Injection of the geospatial information passed to Mapnik is accomplished with the gdal_translate tool from the GDAL software suite.

Because the output GeoPackage from the first workflow stage contains all the geometric and semantic information needed to render land cover bitmaps, it can be copied to other computers to repeat the process given the availability of the involved GDAL and Mapnik tools. If the combined database covers a large area, e.g., a country or continent, it can be reused to locally render different target scenes of the contained cities, counties or states. Runtime and storage requirements however scale with the amount of input. File sizes can be reduced by enabling seek-optimized Deflate compression in the GDAL GPKG driver. To facilitate the deployment of the workflow, its stages have been poured into Python scripts available for download [1]. The repository also hosts the XML style sheets for the CLC level 3 and VBS4 nomenclatures, Mapnik binaries and static vector inputs.

3.1 Area Database Combination

The purpose of the area database combination stage of the workflow is to produce a self-contained spatially indexed all-vector database file whose geometric primitives can then be used to render LULC bitmaps at any scale. It condenses, fuses and orders semantically annotated vector features from the OSM serialization, base layer and global water map for fast access.

High-resolution data for the land cover to be rendered comes from OpenStreetMap serializations, which can be retrieved online for different areas of the world (Geofabrik GmbH, 2024). These files in OSM Protocolbuffer Binary Format (.osm.pbf), OSM XML and other geographically located representations effectively are snapshots of a collection of attributed nodes, relations and ways (OpenStreetMap contributors, 2024b). Because the original motivation behind OpenStreetMap is a freely available comprehensive world map, only a fraction of the contained information will be needed for LULC layers. Thus, the OSM serialization is filtered and turned into attributed geometric primitives like points, lines and multipolygons (collective term for single polygons, polygons with holes and disjoint polygon sets) with the GDAL ogr2ogr utility. The subset of attributes to be kept for each class of primitives is passed via a text-based configuration file named osmconf_lulc.ini. Relevant OSM keys listed in this file include landuse, natural, surface and amenity and more specialized fields like highway, building and leaf_cycle to further subdivide the main classes of fine-grained LULC nomenclatures such as CLC. To approximate the real-world proportions of linear features in the render stage, their width and lane count (for roads) attributes will be extracted if present. These values may appear in the raw OSM data in a variety of notations. To homogenize width entries with and without units of measure, decimal numbers and feet/inch representations, text fields containing non-numeric characters will be discarded. This is achieved with Structured Query Language (SQL) code in the configuration file that builds on a case statement and type casts. Discouraged but present composite lane counts like 2;1 or 2+1 for same-direction and oncoming traffic likewise are truncated to integers or removed. As output, the ogr2ogr tool is instructed to write a GeoPackage file. For each kind of primitives, the self-describing preliminary container stores their geometries and attributes into dedicated tables of a relational SQLite database. The GeoPackage is assigned the OSM geodetic coordinate system (WGS84 longitude/latitude, EPSG code 4326 (IOGP, 2024)) and a spatial index to accelerate queries on subregions.

OpenStreetMap data does not directly map extended water surfaces, and the shore information present is often incomplete. Land boundaries therefore are taken from the dedicated OSM water polygon shape file that contains oceans and seas derived from the repaired OSM coastlines. Also, land-side gaps in areas not mapped yet by the project volunteers get filled from a specifically prepared version of the ESA WorldCover LULC raster layer from 2021. Data preparation starts with downsampling the 7.46 terapixel bitmap to 8.3 gigapixels changing its GSD from roughly 10 m/pix to 280 m/pix. Pixels classified as sea or urban areas are set to zero as these regions will be modeled in more detail by the water polygons and OSM land cover overlay. The resulting voids are interpolated with the gdal_fillnodata Python tool from GDAL using the nearest neighbor filter to preserve the original LULC categories. The maximum search distance for the interpolator is 1000 pixels. This setting will grow land areas into the sea to compensate inaccuracies in the water polygon shapes. Small regions are eliminated with an 11 x 11 median filter to reduce the polygon count during subsequent vectorization. Vectorization is performed with a tiny Python program that calls functions from the free Shapely and GeoPandas libraries (Gillies, 2024)(GeoPandas developers, 2024). Figure 2 exemplifies the water polygons and prepared WorldCover layer.

Both the water and base layer shapes are added to the preliminary GeoPackage with ogr2ogr as separate tables of multipolygons. Their columns store landuse and natural attributes according to the OSM labeling conventions. The final outcome of the first workflow stage is an all-vector database file that allows rendering the high-detail OSM land cover, low-resolution base layer and water polygons together using almost identical style sheet rules. Nevertheless, individual tables permit selective data analysis and visualization depending on the target application.

3.2 LULC Map Rendering

The second workflow stage turns the created GeoPackage into a LULC bitmap of a specific nomenclature. Rendering is done with the unmodified demo program mapnik-render from the Mapnik toolkit. The program expects a style sheet, the output image size and desired scene extent in geodetic coordinates as input to produce a PNG raster without localization. To facilitate the invocation of mapnik-render for cartographers, a Python script developed to control rendering substitutes the PNG size parameters with a metric GSD value. Width and height of the output bitmap are computed from the Cartesian coordinates of the extent in the target Web Mercator projection divided by the ground resolution aimed at. Sparse transformation of the geographic bounding box corners is carried out with the cs2cs

[1] https://www.github.com/DLR-OS/lrlulc

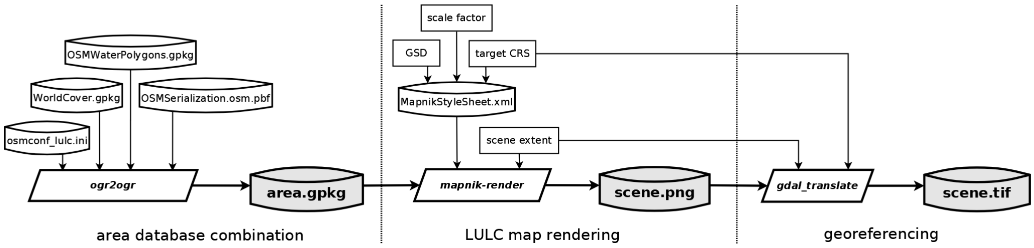
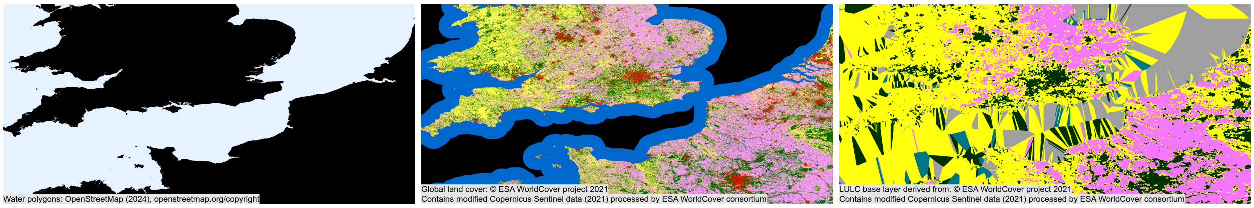


Figure 1. Local file-based LULC production workflow.



(a) (b) (c)

Figure 2. Base layer for the English Channel (a) OSM water polygons, (b) original WorldCover raster, (c) processed and vectorized WorldCover data to be masked with the water polygons in the render stage.

program from the PROJ library, a GDAL dependency (PROJ contributors, 2024).

The required style sheet is written in Mapnik XML. At the top level, among other global properties, it contains a map element with the target coordinate system to reproject any geometries into. The map element hosts a set of styles that shall be applied to different output image layers. The layers are linked to source vector data as provided by the tables of the previously produced GeoPackage. During rendering, geometric primitives are taken from the tables assigned to the layer and rasterized into pixels colored according to the corresponding style. This process follows the painter’s algorithm, i.e., layers declared first will be overwritten by layers declared later. It therefore becomes possible to decorate large polygonal landscape areas such as forests or grassland with specialized elements like roads or single buildings.

Figure 3 has sample XML code of a Mapnik style sheet that shows how the “continuous urban fabric” class of the CLC nomenclature could be rendered. The corresponding layer considers all multipolygon geometries from the respective table stored inside the GeoPackage file. Their coordinates are assumed to be geodetic as denoted by the assigned srs attribute. Read access shall happen through Mapnik’s OGR I/O plugin. The style contains a filter rule that defines the OSM keys it is in charge for (landuse, building, highway). If the filter rules match the tagging of the current multipolygon, the symbolizer code is applied to decorate the inside of that shape using the given hexadecimal RGB color triplet (fill) and anti-aliasing level (gamma). Edge smoothing must be disabled for LULC output to not blend together class labels, i.e., gamma is always zero. Decoration instructions may contain expressions composed of OSM attributes, constant values and arithmetic operations to be evaluated at runtime.

For flexibility, class colors, coordinate system choices, layer data sources and other settings are encapsulated in separate files as XML entity definitions. These definitions are referenced by the static Mapnik style sheet. The style sheet currently implemented holds rules that filter OSM attributes according to the most detailed CLC categorization. Alternative decorations can be rapidly realized by changing the definitions of the CORINE colors taken from (European Environment Agency, 2024) to the RGB triplets of the target classification scheme in the dedicated include file. However, altering the human-readable filter rules remains possible as an advanced option, e.g., when a particular land cover nomenclature exposes a level of granularity deviating from CLC. Category mapping has been exercised for the VBS4 encoding used in section 4.3, and both styling approaches were applied for statistical quality assessment in section 4.2.

The shown XML snippet contains render code for multipolygons. Their dimensions in reality will be correctly reflected when Mapnik reprojects the 2D shapes from the OSM reference frame into the target coordinate system of the output LULC map and fills the interior according to the style setting. However, for layers that address lines as they are predominantly used for infrastructure by the OpenStreetMap project, scaling of one-dimensional elements to their real size in the world with respect to the target projection happens only along the path but not across. This is acceptable for standard maps that Mapnik has been optimized for since roads, railway tracks and water courses are typically depicted as fixed-size curves depending on their importance or capacity. The line symbolizer provided

```
<Map srs="epsg:3857" background-color="#000000">
  <Style name="LULC_Corine_ContinuousUrbanFabric">
    <Rule>
      <Filter>
        [landuse]='residential' or [building]='appartments'
        or [highway]='pedestrian' or [highway]='footway'
      </Filter>
      <PolygonSymbolizer gamma="0" fill="#e6004d" />
    </Rule>
  </Style>
  ...
  <Layer name="layer1" srs="epsg:4326">
    <StyleName>LULC_Corine_ContinuousUrbanFabric</StyleName>
    <Datasource>
      <!-- GeoPackage input read with Mapnik ogr plugin -->
      <Parameter name="file">NewYork.gpkg</Parameter>
      <Parameter name="type">ogr</Parameter>
      <!-- Take multipolygons for this output image layer -->
      <Parameter name="layer">multipolygons</Parameter>
    </Datasource>
  </Layer>
  ...
</Map>
```

Figure 3. Mapnik XML snippet of CLC level 3 style sheet.

by the toolkit hence takes the stroke width in pixels as a parameter. However, to display not just the correct length but also the true width of infrastructure elements in land cover bitmaps, the second dimension for linear objects must be determined and translated into a pixel count. Calculation of the stroke setting is shared between the render stage script and XML style sheet.

The Python script for the render stage provides the GSD and a scale factor that models the distortion of the target projection. Since the orientation of roads, rails and other infrastructure is unpredictable for the scene to be rendered, the direction of the width vector to be scaled is also unknown. This limits the set of target projections for the LULC image as specified in the outmost tag of the Mapnik style sheet to conformal mappings. Conformal mappings expose identical distortion factors in longitude and latitude directions that will be almost constant for small regions. Suitable projections for land cover bitmaps include Spherical/Web Mercator (EPSG code 3857) as in the code excerpt, WGS84 Mercator (EPSG code 3395) and Peirce quincuncial (Snyder and Voxland, 1989). All these mappings of the globe to a plane cover almost any location worldwide, do not require additional parameters like zones or grids and completely fill rectangular raster images. However, the computational effort to obtain the distortion scale $s(\lambda, \phi)$ from longitude λ and latitude ϕ ranges from $s = 1/cos(\phi)$ for the Web Mercator projection to the numerical integration of elliptic curves for Peirce quincuncial (Eisele, 1963). Thus, the Python script in charge for rendering currently fixes the LULC map coordinate system to Web Mercator.

The XML style sheet evaluates the metric width attribute from the source GeoPackage. If the attribute is not present, as a rough estimate at least for highways, it takes the lane count instead with a constant lane width of 3.5 m. On a missing number of lanes, their count will be set to two. Since Mapnik evaluates arithmetic terms in line symbolizer settings, the stroke width is derived in the style sheet as the product of the width in reality, the inverse GSD (in pix/m) and the distortion scale. When the lane count is available only or has to be guessed, the brush is configured from the number of lanes multiplied by their assumed average width, the inverse GSD and distortion scale.

In the shared calculation, the actual scale to be passed from the Python script to the style sheet is determined for the center coordinates of the render extent. For large areas running primarily North to South, this will introduce inaccuracies in the width of vertical line objects depicted as the distortion value changes over the LULC map. A more accurate scaling would be accomplished when the GeoPackage tables for lines are enhanced with their geographic coordinates. This however requires to dissect the geometry blobs of the primitives and compute a representative coordinate tuple from the path vertices in the area database combination stage. The brush setting calculation then needs to be put entirely into the XML sheet, which impacts render time.

Point features like single trees currently do not get processed into LULC items. For such objects, projection distortions would have to be considered for both the width and height as mandated by the Mapnik point symbolizer.

3.3 Georeferencing

The last workflow stage adds the extent passed to mapnik-render to the LULC PNG bitmap and assigns it the target coordinate system, i.e., Web Mercator. This task is performed with the gdal_translate utility. The localized output raster is stored in a data format that supports geographic references, e.g., GeoTIFF. Due to the limited set of colors needed to encode land cover classes, image file sizes likely will benefit from lossless compression if supported. The output from the stage can be transformed into any other CRS, for instance, with the gdalwarp tool from GDAL, or loaded into GIS software for further analysis.

4. Results

The quality of LULC images created from OpenStreetMap data is visually inspected for sample renderings of places around the globe. Spatial accuracy and correctness of selected thematic maps are compared statistically to validated references. As a practical application, sub-meter land cover gets loaded into the VBS4 simulator to model the environment of a 3D scene.

4.1 Visual Analysis

Four areas have been chosen to visually assess various aspects of LULC rendering, which took place on a 2021 workstation-class PC with rotating disk drives. As a high-resolution example, a thematic map of the city center of Berlin/Germany comprising 9462 x 4573 pixels was computed in 11 seconds at a target GSD of 0.2 m/pix. The compressed input GeoPackage with relevant OpenStreetMap entries for the entire Berlin-Brandenburg metro region covers 30546 km^2 and consumes 706 MiB of mass storage. For a larger low-resolution area, attributed OSM geometries of Australia and Oceania were clipped and prepared into a 222 MiB deflated GPKG file of New Guinea during the initial workflow stage. Its output was subsequently used to generate a land cover image of 10853 x 5594 pixels for the sparsely populated island. Processing of the target comprising 786000 km^2 took 334 seconds including I/O operations for the chosen GSD of 200 m/pix. Figure 4 opposes the LULC layers to congruent satellite orthophotos and standard OSM tiles.

Due to the lossless scalability of OSM vector data, any LULC elements expose straight edges in both renderings. This is in contrast to image-based labeling algorithms that operate on pixel sets. In the Berlin land cover image, the shapes of single buildings, roads, green spaces and water bodies can be distinguished to a level suitable for 3D simulation. The interior of New Guinea has large forest areas not mapped yet by the OpenStreetMap contributors, which got filled from WorldCover multipolygons. Block artifacts remain visible because the vector layer originates from a subsampled raster image. This issue could be resolved by tracing a high-resolution WorldCover bitmap. However, since storage requirements scale with the number and complexity of the base layer shapes, any database file transfers would potentially be slowed down or become prohibitive. The water polygons stored as a separate table inside the GeoPackage of the area tightly encircle the main land mass and its surrounding islands. Inaccuracies in the coast lines are masked by the spatially extended WorldCover base layer.

Scaling of linear features is illustrated for data sets of Honululu/USA near the Tropic of Cancer and Tromsø/Norway at the Arctic Circle. The equidirectional distortion factors for the Web Mercator projection that depend on the latitude only are $s_H = 1.07$ and $s_T = 2.9$ respectively. Figure 5 shows how different kinds of width information were extracted from the OSM data and multiplied with the inverse target GSD and s values to realistically display roads in the land cover images. The stroke sizes used to decorate these elements approximate their appearance in the orthophoto. In contrast, road visualization in the OSM maps seems to be solely guided by highway categories.

https://doi.org/10.5194/isprs-annals-X-G-2025-269-2025 |

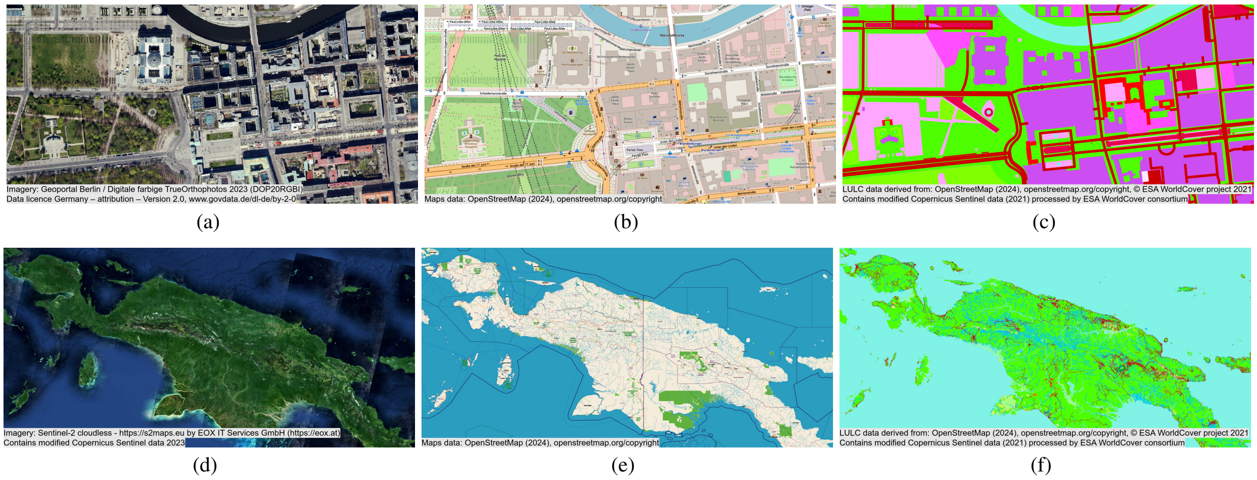


Figure 4. Comparison of orthophotos, native OSM tiles and the produced CLC level 3 images at different GSDs (a-c) Berlin/Germany, 0.2 m/pix (d-f) New Guinea, 200 m/pix, with gaps filled from WorldCover data and OSM water polygons applied.

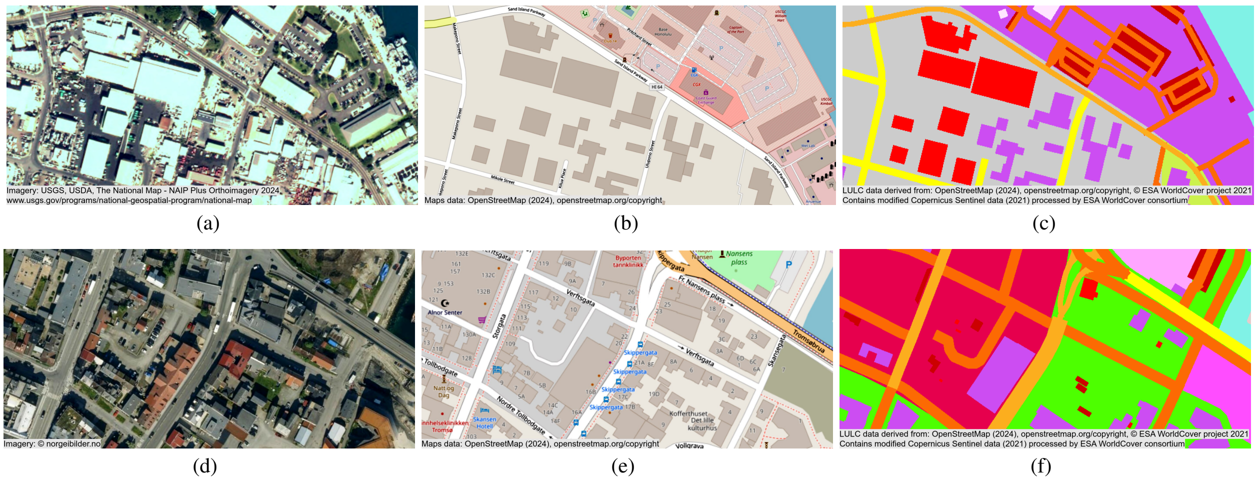


Figure 5. Roads in orthophotos, OSM maps and CLC level 3 images with real dimensions estimated from metric data (yellow), lane count/constant lane width (bright orange) and as fixed width (dark orange) (a-c) Honululu/USA, 21.3°N (d-f) Tromsø/Norway, 69.7°N.

4.2 Statistical Evaluation

Due to the lack of freely available validated high-detail CLC level 3 maps, the statistical evaluation of results from the proposed workflow is performed on a subset of data from the cited UrbanWatch project. Its land cover bitmaps picture different urban and semi-urban places in the United States that are also covered by OpenStreetMap. The original resolution of the reference is 1 m/pix with a claimed overall accuracy of 91.52%. Since the UrbanWatch segmentation is neither based on OSM nor WorldCover data, an unbiased comparison is ensured.

For the comparison, the reference images were reprojected into the Web Mercator coordinate system, which slightly degraded their GSD to between 1.2 to 1.68 m/pix. Congruent LULC rasters based on OSM data from the states the chosen targets are a part of got rendered. To translate the CORINE into UrbanWatch land cover categories, both the plain color mapping and advanced XML filter rule/color encoding adaptation approaches were examined. Table 1 contains the overall and per-class similarities between the ground truth and render output after a pixel by pixel comparison. Figure 6 illustrates LULC details for the Seattle area in Washington State.

As the readings indicate, the overall correlation between the reference and OSM LULC images ranges from 43.5% to 60.4% when the CLC level 3 rule set is kept and only the color map gets adjusted. Adapting both the XML style sheet and color map to the UrbanWatch nomenclature increases the match rate by 1.6% to 36%. Congruences are strong for the road and water categories with little fluctuations when the customized style sheet comes into play. As these features constitute basic building blocks for the classic maps that OpenStreetMap focuses on, they seem to appear accurately modeled by the community.

Significant overlap between the reference and render results also exists for buildings and canopy. However, when the style sheet gets adapted in addition to the color encoding, a reciprocal effect is revealed, i.e., the building percentages go down while the canopy readings increase. This partly is due to the unmodified XML style sheet treating densely packed buildings block-wise rather than individually according to the CLC nomenclature. Hence, any trees in the front spaces and backyards of dwellings will be omitted. In contrast, the style sheet specifically tailored to UrbanWatch does not consider OSM residential, commercial or industrial landuse zones but single structures. When zoning is the only information present in the vector data, matches for the building class will decrease, and canopy from the WorldCover base layer will prevail. There currently is no way to conditionally render spatially overlapping OSM objects with the style sheet alone. Hints to e.g. reproduce individual structures instead of subjacent landuse polygons or ras-

area	share and match percentages - reference ↦ rendered OpenStreetMap LULC map																							
	overall		building			road			parking			water			canopy			grass			barren			agriculture		
	m	f/m	gt	m	f/m	gt	m	f/m	gt	m	f/m	gt	m	f/m	gt	m	f/m	gt	m	f/m	gt	m	f/m	gt	m	f/m
Charlotte, NC (37.2 km^2)	60.4	61.4	9.0	76.2	74.3	14.5	55.7	55.4	9.6	0.0	5.7	0.3	41.4	41.4	48.0	87.1	88.4	18.3	19.3	19.3	0.3	10.4	10.4	0.0	0.0	0.0
Seattle, WA (123.6 km^2)	43.6	59.3	14.9	80.4	64.8	17.8	57.2	56.3	4.0	0.0	18.6	7.9	99.3	99.3	42.9	24.2	64.9	11.9	26.1	26.2	0.1	4.7	4.7	0.0	0.0	0.0
Denton County, TX (170.3 km^2)	50.0	56.0	12.6	92.0	68.9	15.9	60.8	60.5	6.0	0.0	6.2	4.0	84.0	84.0	28.2	33.3	50.5	30.9	49.2	61.5	1.8	39.4	39.4	0.3	0.0	0.0
New York City, NY (181.0 km^2)	53.6	54.6	31.8	59.9	59.2	23.5	48.7	47.7	4.4	0.0	7.8	0.8	96.7	96.7	30.9	56.2	59.8	8.2	59.4	59.8	0.1	24.3	24.3	0.0	0.0	0.0
Riverside, CA (29.8 km^2)	43.5	48.4	16.0	81.2	64.8	24.2	41.8	41.4	9.0	0.0	3.9	0.0	0.1	0.1	22.2	43.3	73.8	18.6	15.1	17.5	0.4	4.2	4.2	9.0	87.8	87.8

Table 1. Similarity between the UrbanWatch reference and output LULC images (gt) category share in ground truth, (m) similarity with adapted color map only, (f/m) similarity with adapted filter rules plus color map.

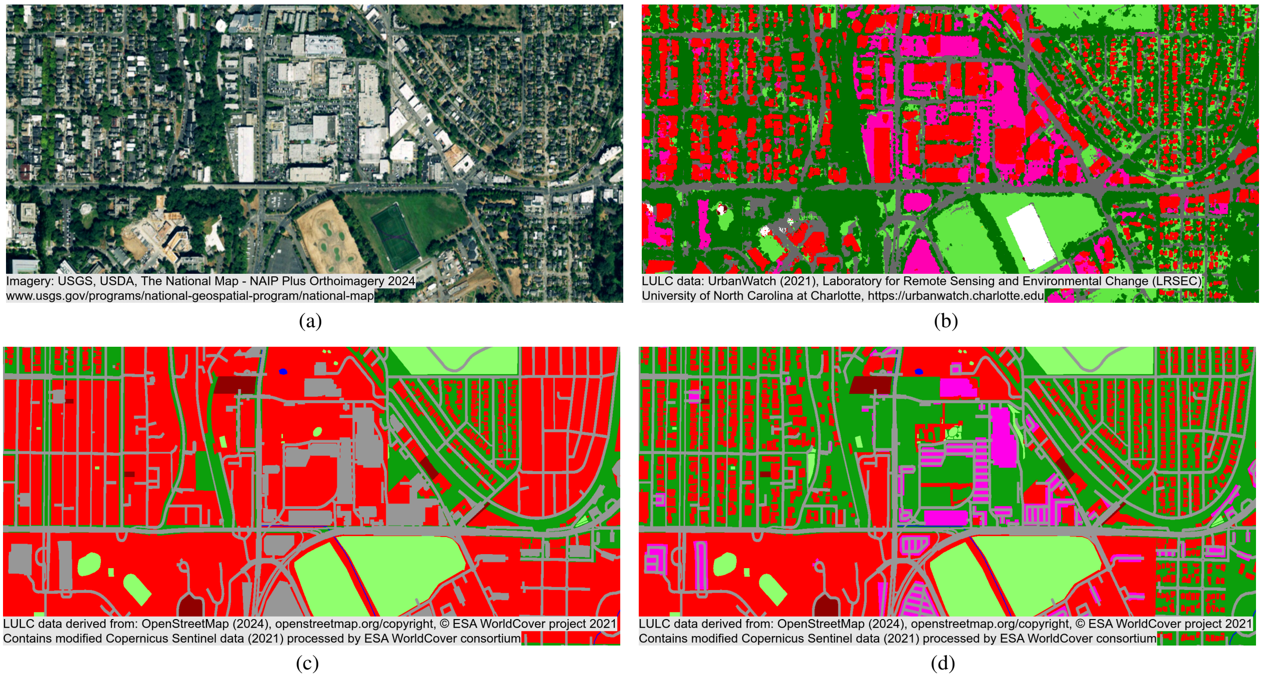


Figure 6. LULC for Seattle/USA (a) ortho view, (b) UrbanWatch reference, (c) render result with CORINE color remapping only absorbing individual structures, (d) result with adapted style sheet and color remapping showing single buildings and parking lots.

terize polygons not containing buildings however could be realized with custom attributes in the GeoPackage database tables. More precisely, in the area database combination stage of the workflow, one would have to determine whether a landuse polygon contains any detailed objects, and if so, mark it as occupied in an extra GeoPackage table column named e.g. “occupation”. The XML file implementing the decoration rule for landuse polygons could then evaluate the “occupation” key in its filter part. When the landuse polygon was flagged as occupied, rendering might be skipped in favor of the contained objects whose separate layer declaration must appear later in the style sheet due to the painter’s algorithm used by Mapnik.

Little similarities between the ground truth and OSM LULC images could be observed for grass, barren and agricultural lands as well as for parking areas with only minor changes when the custom rule set gets applied. For the first three categories, this is mostly because these regions remain unmapped in OpenStreetMap and have to be rendered from the coarse base layer. In the absence of detailed thematic data, matches for urban scenes will fully depend on the interpolation correctness when the built-up areas removed from WorldCover are labeled from the surrounding landscape. While car parks may have been modeled in-depth in OpenStreetMap, they will be subsumed under the traffic category and not represented explicitly using the CLC style sheet. However, with specialized UrbanWatch filter rules, they exhibit non-zero percentages in the table, but still suffer from undersegmentation in the ground truth.

4.3 Application Example - VBS4

As a use case for the proposed workflow, a 0.5 m/pix LULC raster of Heligoland/Germany was created for the Virtual Battlespace 4 simulator and warped from Web Mercator to the geodetic reference frame. The CLC level 3 color map got replaced by the VBS4 graylevel map encoding 39 surface classes. The environment data could then be directly loaded through the Geo Mode interface of the software. Figure 7 shows the land cover layer the 3D engine utilized to plant distinguishable patches of geospecific vegetation (mostly grass, trimmed grass and scrub) in real-time precisely taking the stadium area into account. This is only possible with a highly detailed surface description.

5. Conclusion

This paper presented a file-based workflow hinged on free software that reinterprets OpenStreetMap data to create land use/ land cover bitmaps for almost any part in the world at arbitrary

https://doi.org/10.5194/isprs-annals-X-G-2025-269-2025 | © Author(s) 2025. CC BY 4.0 License.

(a) (b)

Figure 7. Heligoland/Germany in VBS4 (a) graylevel OSM LULC layer, (b) auto-generated geospecific vegetation patches.

resolutions on a standard personal computer. The achievable thematic and positional accuracy of the output is directly inherited from the attributed input geometries, vectorized WorldCover base layer and OSM water polygon mask. Line features get scaled to approximate their real-world proportions. To further improve the quality of the results, conditional rendering of overlapping objects as briefly outlined remains to be investigated in-depth. LULC encodings for specific applications can be obtained by editing the original human-readable CLC color table or OpenStreetMap filter rules. This was demonstrated for the VBS4 simulator.

References

Bohemia Interactive Simulations, 2022. VBS4 22.1.0 Manuals. Bohemia Interactive Simulations, Prague, Czech Republic.

Di Gregorio, A., Jansen, L. J., 1998. Land Cover Classification System (LCCS): Classification Concepts and User Manual. Food and Agriculture Organization, Rome, Italy.

Eisele, C., 1963. Charles S. Peirce and the Problem of Map-Projection. *Proceedings of the American Philosophical Society*, 107(4), 299-307.

European Environment Agency, 2024. CORINE Land Cover - Band information. https://github.com/sentinel-hub/collections/tree/main/collections/corine-land-cover (12 September 2024).

Fisher, P., Comber, A., Wadsworth, R., 2005. Land use and Land cover: Contradiction or Complement. P. Fisher, D. J. Unwin (eds), *Re-presenting GIS*, John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, 85–98.

Fonte, C. C., Patriarca, J. A., Minghini, M., Antoniou, V., See, L., Brovelli, M. A., 2017. Using OpenStreetMap to Create Land Use and Land Cover Maps. *Volunteered Geographic Information and the Future of Geospatial Data*, IGI Global, 113–137.

GDAL/OGR contributors, 2024. GDAL/OGR Geospatial Data Abstraction Library. Open Source Geospatial Foundation. https://gdal.org (9 June 2024).

Geofabrik GmbH, 2024. OpenStreetMap Data Extracts. Download server. https://download.geofabrik.de (10 July 2024).

GeoPandas developers, 2024. GeoPandas Documentation. https://geopandas.org/en/stable/docs.html (11 August 2024).

Gillies, S., 2024. The Shapely User Manual. https://shapely.readthedocs.io/en/stable/manual.html (30 May 2024).

IOGP, 2024. EPSG Geodetic Parameter Dataset. International Association of Oil & Gas Producers Geomatics Committee. https://epsg.org (11 August 2024).

Johnson, N., Treible, W., Crispell, D., 2022. OpenSentinelMap: A Large-Scale Land Use Dataset using OpenStreetMap and Sentinel-2 Imagery. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1332–1340.

Kerchove, R. V. D., Zanaga, D., Xu, P., Tsendbazar, N., Lesiv, M., 2022. WorldCover Product User Manual. European Space Agency.

Kosztra, B., Büttner, G., Hazeu, G., Arnold, S., 2019. *Updated CLC illustrated nomenclature guidelines*. Environment Agency Austria, Vienna, Austria.

Laminar Research, 2024. MeshTool 3.0 Instructions and Release Notes. X-Plane Developer MeshTool website. https://developer.x-plane.com/tools/meshtool (19 August 2024).

Open Geospatial Consortium, 2024. OGC GeoPackage Encoding Standard version 1.4. OGC standards website. https://www.geopackage.org/spec140/index.html (10 September 2024).

OpenStreetMap contributors, 2024a. Open Street Map website. https://www.openstreetmap.org (10 September 2024).

OpenStreetMap contributors, 2024b. OpenStreetMap Elements. https://wiki.openstreetmap.org/wiki/Elements (31 July 2024).

OpenStreetMap contributors, 2025. Data Derived from OpenStreetMap for Download - Water polygons. https://osmdata.openstreetmap.de/data/water-polygons.html (1 February 2025).

Pavlenko, A., 2024. Mapnik toolkit for map applications. Mapnik GitHub repository. https://github.com/mapnik/mapnik (10 September 2024).

PROJ contributors, 2024. PROJ coordinate transformation software library. Open Source Geospatial Foundation. https://proj.org (9 June 2024).

Ray, E. T., 2001. *Learning XML*. O'Reilly Media, Sebastopol, CA, USA.

Schultz, M., Voss, J., Auer, M., Carter, S., Zipf, A., 2017. Open land cover from OpenStreetMap and remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 63, 206-213.

Snyder, J. P., Voxland, P. M., 1989. An album of map projections. Technical report, U.S. Geological Survey, Alexandria, VA, USA.

UN-GGIM, 2019. The Global Fundamental Geospatial Data Themes. Technical report, United Nations Committee of Experts on Global Geospatial Information Management, New York City, NY, USA.

Wang, Y., Sun, Y., Cao, X., Wang, Y., Zhang, W., Cheng, X., 2023. A review of regional and Global scale Land Use/Land Cover (LULC) mapping products generated from satellite remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 206, 311-334.

Zanaga, D., Van De Kerchove, R., Daems, D., Keersmaecker, D. et al., 2022. ESA WorldCover 10m 2021 v200. https://doi.org/10.5281/zenodo.7254221 (17 September 2024).

Zhang, Y., Chen, G., Myint, S. W., Zhou, Y., Hay, G. J., Vukomanovic, J., Meentemeyer, R. K., 2022. UrbanWatch: A 1-meter resolution land cover and land use database for 22 major cities in the United States. *Remote Sensing of Environment*, 278.