ST_LUCAS reference data for online automated land cover mapping

Martin Landa, Tomáš Bouček, Lukáš Brodský, Ondřej Pešek, Lena Halounová
Department of Geomatics
Faculty of Civil Engineering, Czech Technical University in Prague

FOSS4G
Prizren, 2023

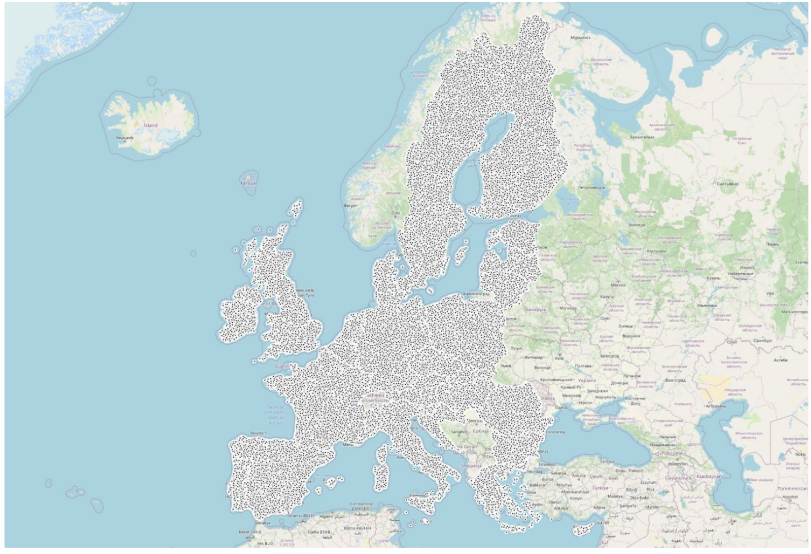
Content

- What is LUCAS?
- What is ST_LUCAS system?
- ST_LUCAS demonstration
- Conclusions

What is LUCAS?

- **Land Use and Coverage Area frame Survey**
- Managed by **Eurostat**
- Accessible as plain CSV files, https://ec.europa.eu/eurostat/web/lucas
- Started in 2000 to identify changes in land cover and land use
- Sampling density given by 2x2 km grid

Year	EU countries	Points
2006	11	168 402
2009	23	234 623
2012	27	270 272
2015	28	339 696
2018	28	337 854



FOSS4G Prizren, 2023

What is LUCAS?

- Examine land cover (76 classes) and land use (41 classes)
- Structural elements in the landscape
- Evaluate agro-environmental information
- Take a 500-gram topsoil sample at one out of 10 points
- Collect photos (facing + 4 directions)

Land cover			
A00	ARTIFICIAL LAND	A10	Roofed built-up areas
		A20	Artificial non-built up areas
		A30	Other artificial areas
B00	CROPLAND	B10	Cereals
		B20	Root crops
		B30	Non-permanent industrial crops
		B40	Dry pulses, vegetables and flowers
		B50	Fodder crops
		B70	Permanent crops: fruit trees
		B80	Other permanent crops
C00	WOODLAND	C10	Broadleaved woodland
		C20	Coniferous woodland
		C30	Mixed woodland
D00	SHRUBLAND	D10	Shrubland with sparse tree cover
		D20	Shrubland without tree cover
E00	GRASSLAND	E10	Grassland with sparse tree/shrub cover
		E20	Grassland without tree/shrub cover
		E30	Spontaneously re-vegetated surfaces
F00	BARE LAND AND LICHENS/MOSS	F10	Rocks and stones
		F20	Sand
		F30	Lichens and moss
		F40	Other bare soil
G00	WATER AREAS	G10	Inland water bodies
		G20	Inland running water
		G30	Transitional water bodies
		G40	Sea and ocean
		G50	Glaciers, permanent snow
H00	WETLANDS	H10	Inland wetlands
		H20	Coastal wetlands

What is LUCAS?

- LUCAS attributes evolution
  - 5 removed
  - 77 added
  - 24 renamed
  - 30 coding changed - example (LC1):
    - C21 - Other broadleaved tree land (2006)
    - C21 - Spruce dominated coniferous woodland (2012 - 2018)
- Conclusions
  - **Real and unique ground-true data** for land products validation and new models calibration
  - Not analysis-ready dataset for temporal change analysis

Year	Number of attributes
2006	20
2009	44
2012	46
2015	59
2018	97

What is ST_LUCAS system?

What is ST_LUCAS system?

- Provides **harmonized** (fully automated) LUCAS data
- Provides harmonized **space-time** (ST) aggregated LUCAS data
- Configurable & Extensible system
- Standardized data interface (OGC OWS)
- Python API for geospatial developers and scientists
- QGIS plugin for wider audience
- Translation method to provide LUCAS land cover data in other nomenclatures
- Allow user-defined analytics as e.g. the legend aggregation

Order Article Reprints

Open Access Article

Open Geospatial System for LUCAS In Situ Data Harmonization and Distribution

by Martin Landa *,†, Lukáš Brodský †, Lena Halounová, Tomáš Bouček and Ondřej Pešek

Department of Geomatics, Faculty of Civil Engineering, Czech Technical University in Prague, 166 29 Prague, Czech Republic

* Author to whom correspondence should be addressed.

† These authors contributed equally to this work.

ISPRS Int. J. Geo-Inf. **2022**, *11*(7), 361; **https://doi.org/10.3390/ijgi11070361**

Received: 6 May 2022 / Revised: 10 June 2022 / Accepted: 20 June 2022 / Published: 23 June 2022

Download | Browse Figures | Versions Notes

Abstract

The use of in situ references in Earth observation monitoring is a fundamental need. LUCAS (Land Use and Coverage Area frame Survey) is an activity that has performed repeated in situ surveys over Europe every three years since 2006. The dataset is unique in many aspects; however it is currently not available through a standardized interface, machine-to-machine. Moreover, the evolution of the surveys limits the performance of change analysis using the dataset. Our objective was to develop an open-source system to fill these gaps. This paper presents a developed system solution for the LUCAS in situ data harmonization and distribution. We have designed a multi-layer client-server system that may be integrated into end-to-end workflows. It provides data through an OGC (Open Geospatial Consortium) compliant interface. Moreover, a geospatial user may integrate the data through a Python API (Application Programming Interface) to ease the use in workflows with spatial, temporal, attribute, and thematic filters. Furthermore, we have implemented a QGIS plugin to retrieve the spatial and temporal subsets of the data interactively. In addition, the Python API includes methods for managing thematic information. The system provides enhanced functionality which is demonstrated in two use cases.

Keywords: LUCAS; in situ; data harmonization; data distribution; web services; QGIS plugin

FOSS4G Prizren, 2023

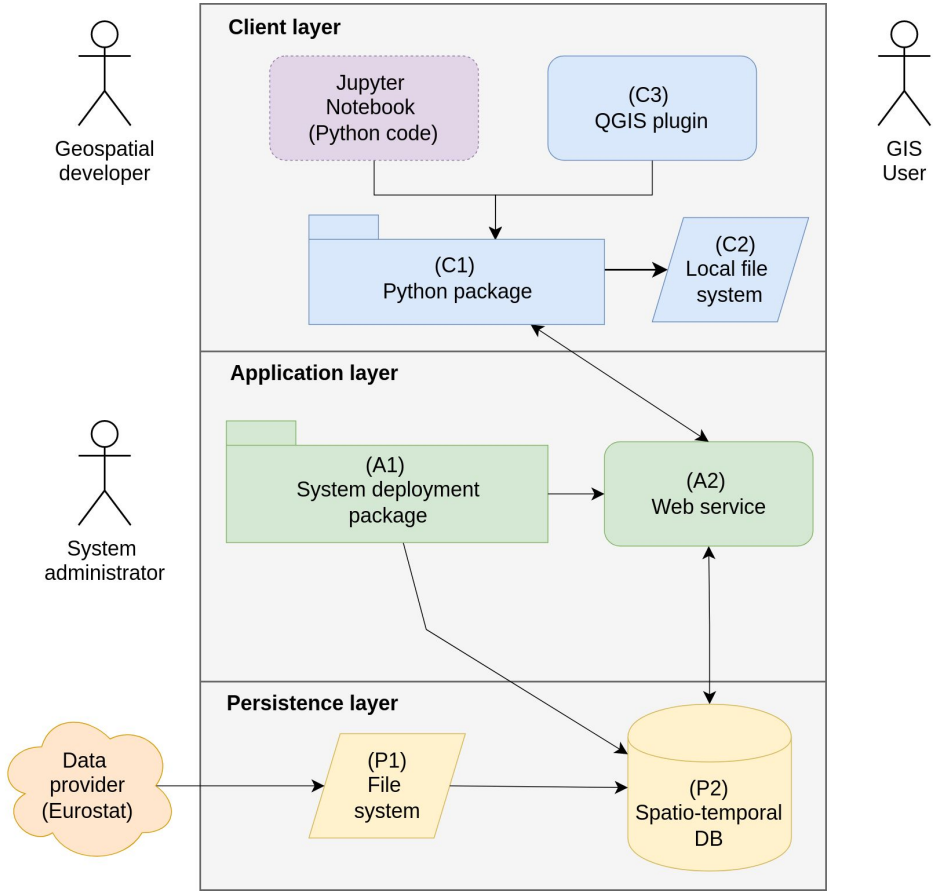
ST_LUCAS architecture

Persistent data storage (P1, P2)

Automation of the harmonization process & space-time aggregation (A1)

Standardized (OGC) web service (A2)

Client Python API (C1) & QGIS plugin (C3)



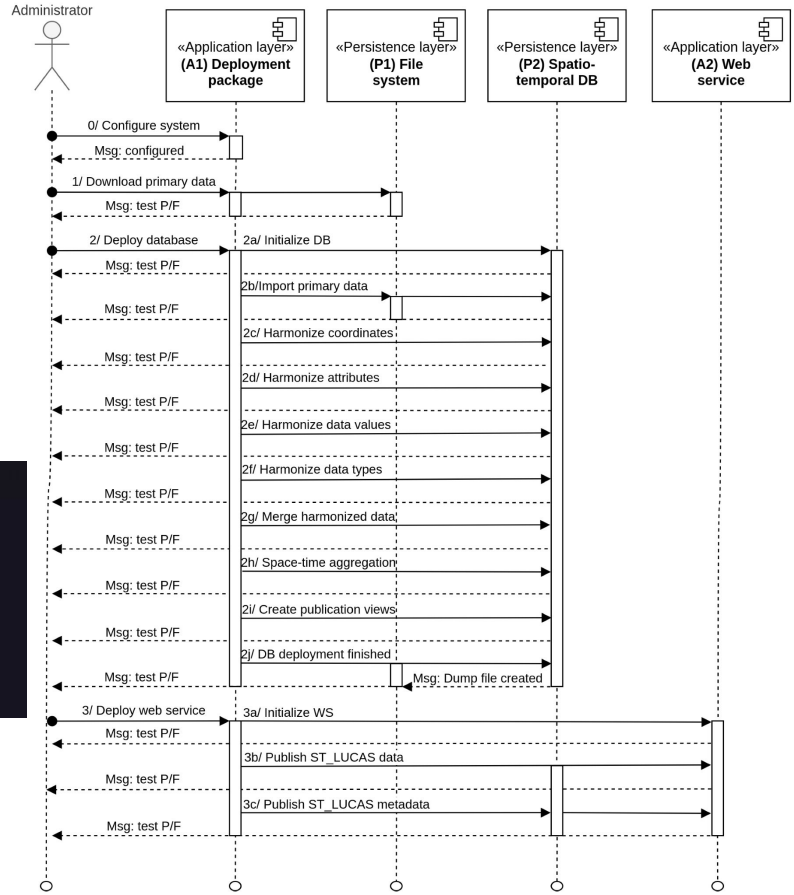
ST_LUCAS deployment

1. Configure system (step 0)

```
POSTGRES_PASSWORD=very_secret_pw0
POSTGRES_DB=lucas
POSTGRES_SCHEMA=data
MAPSERVR_SCHEMA=ms
```

2. Perform deployment (steps 1-3)

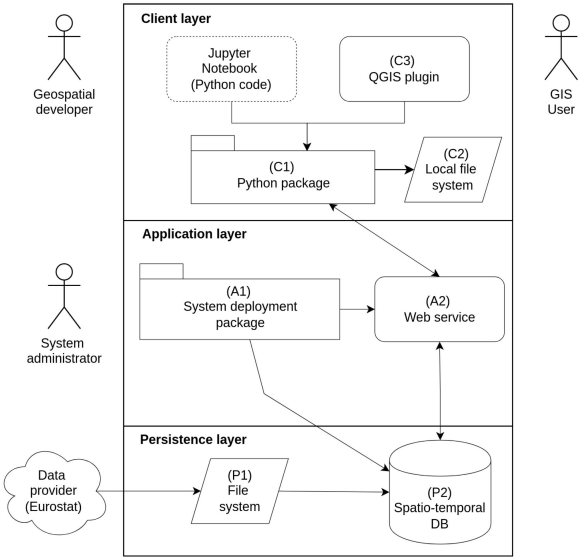
```
landa@lincalc-02:/opt/st_lucas$ docker compose up
```



FOSS4G
Prizren, 2023

ST_LUCAS automated tests

- Unit tests

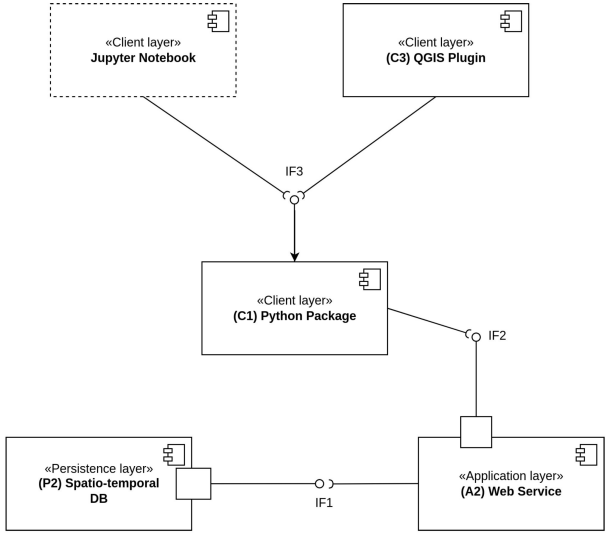


Component ID	Test IDs	Description
A1, P1	1_001	Primary data are downloaded according to the system configuration.
A1, P2	2a_001	DB is initialized according to the system configuration.
	2b_001-003	Primary data are imported according to the system configuration.
	2c_001-002	Coordinates are harmonized according to the system configuration.
	2d_001	Attributes are harmonized according to the system configuration.
	2e_001-002	Data values are harmonized according to the system configuration.
	2f_001	Data types are harmonized according to the system configuration.
	2g_001-004	Harmonized data are merged according to the system configuration.
	2h_001-003	Data are space–time aggregated according to the system configuration.
	2i_001-004	Publication views are created according to the system configuration.
	2j_001	DB recovery file is created according to the system configuration.
A1, A2	3a_001-003	Test case consists of checking OGC WFS operations: GetCapabilities, DescribeFeatureType and GetFeature.
	3b_001-003	ST_LUCAS dataset available via WFS.
	3c_001-003	The test cases consist of checking that ST_LUCAS metadata are published according to the deployed database.
C1, C2	001-007	Test cases consist of checking `LucasRequest` and `LucasIO` classes methods to build a request, download a LUCAS subset, store retrieved data on the local file system, and access associated photos.

FOSS4G Prizren, 2023

ST_LUCAS automated tests

- Integration tests (daily)



Interface ID	Test IDs	Description
IF1, IF2	001–004	Test cases consist of checking WFS responses retrieved by the Python package (IF2) covering various combinations of spatial, attribute, thematic, and temporal filters. The responses are compared with the subsets retrieved from spatio-temporal DB via SQL statements (IF1). Test cases pass only if there is no difference between the WFS responses and the subsets retrieved from DB.

```
============================= INTEGRATION TESTS =============================
============================= test session starts ==============================
platform linux -- Python 3.9.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0 -- /usr/local/bin/python3
cachedir: /tmp/pytest_cache_dir
rootdir: /opt
collecting ... collected 7 items

test_st_lucas.py::TestST_LUCAS::test_001 PASSED
test_st_lucas.py::TestST_LUCAS::test_002 2023-06-23 13:07:27,040 - root - WARNING - __init__.getSRS - Requested srsName
'EPSG:3035' is not declared as being allowed for requested typename 'lucas:lucas_st_points'. Options are: 'urn:x-
ogc:def:crs:EPSG:3035'.
2023-06-23 13:07:27,127 - LUCAS - INFO - io.download - Download process successfuly finished. Size of downloaded data:
14kb
PASSED
test_st_lucas.py::TestST_LUCAS::test_003 2023-06-23 13:07:27,450 - root - WARNING - __init__.getSRS - Requested srsName
'EPSG:3035' is not declared as being allowed for requested typename 'lucas:lucas_st_points'. Options are: 'urn:x-
ogc:def:crs:EPSG:3035'.
2023-06-23 13:07:27,540 - LUCAS - INFO - io.download - Download process successfuly finished. Size of downloaded data:
14kb
PASSED
test_st_lucas.py::TestST_LUCAS::test_004 2023-06-23 13:07:27,793 - root - WARNING - __init__.getSRS - Requested srsName
'EPSG:3035' is not declared as being allowed for requested typename 'lucas:lucas_st_points'. Options are: 'urn:x-
ogc:def:crs:EPSG:3035'.
2023-06-23 13:07:27,875 - LUCAS - INFO - io.download - Download process successfuly finished. Size of downloaded data:
14kb
PASSED
test_st_lucas.py::TestST_LUCAS::test_005 2023-06-23 13:07:28,138 - root - WARNING - __init__.getSRS - Requested srsName
'EPSG:3035' is not declared as being allowed for requested typename 'lucas:lucas_points'. Options are: 'urn:x-
ogc:def:crs:EPSG:3035'.
2023-06-23 13:07:28,163 - LUCAS - INFO - io.download - Download process successfuly finished. Size of downloaded data:
13kb
PASSED
test_st_lucas.py::TestST_LUCAS::test_006 2023-06-23 13:07:28,349 - root - WARNING - __init__.getSRS - Requested srsName
'EPSG:3035' is not declared as being allowed for requested typename 'lucas:lucas_points'. Options are: 'urn:x-
ogc:def:crs:EPSG:3035'.
2023-06-23 13:07:28,371 - LUCAS - INFO - io.download - Download process successfuly finished. Size of downloaded data:
0kb
PASSED
test_st_lucas.py::TestST_LUCAS::test_007 2023-06-23 13:07:28,564 - root - WARNING - __init__.getSRS - Requested srsName
'EPSG:3035' is not declared as being allowed for requested typename 'lucas:lucas_points'. Options are: 'urn:x-
ogc:def:crs:EPSG:3035'.
2023-06-23 13:08:03,607 - LUCAS - INFO - io.download - Download process successfuly finished. Size of downloaded data:
643628kb
PASSED

============================== 7 passed in 207.68s (0:03:27) ===============================
```

FOSS4G
Prizren, 2023

ST_LUCAS Python package

Functionality:

- Access data provided by ST_LUCAS system
- Access photos provided by GISCO service
- Perform class aggregation
- Perform nomenclature translation

Install:

```
pip install st_lucas
```

ST_LUCAS Python package

Spatial filters

```
from st_lucas import LucasRequest, LucasIO

# define request

request = LucasRequest()

# using bbox

request.bbox = (4504276, 3020369, 4689608, 3105290)

# or by countries

request.countries = ['CZ', 'SK']

# or by user-defined polygon (GML)

request.aoi_polygon = …
```

ST_LUCAS Python package

Additional filters

```
from st_lucas import LucasRequest, LucasIO

# define request

request = LucasRequest()

# by years

request.years = [2006, 2009]

# by thematic groups (subset of attributes

request.group = 'CO'

# by attributes

request.propertyname = 'LC1_H'

from owslib.fes import PropertyIsEqualTo

request.operator = PropertyIsEqualTo

request.literal = ‘A30’
```

LC1_H

code	name
-1	Not relevant
A11	Buildings with one to three floors
A12	Buildings with more than three floors
A13	Greenhouses
A21	Non built-up area features
A22	Non built-up linear features
A30	Other artificial areas

ST_LUCAS QGIS Plugin

QGIS Python Plugins Repository

Download latest

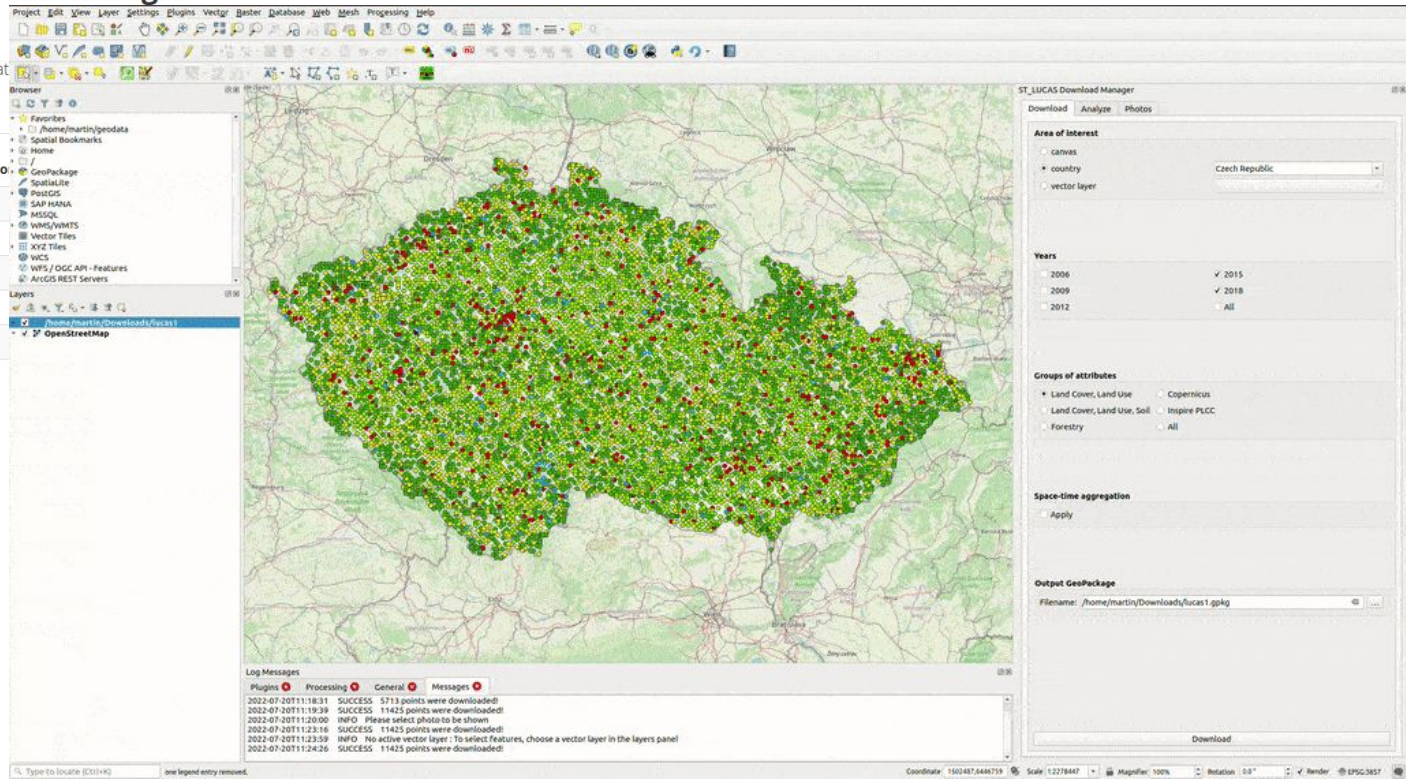
ST_LUCAS Download Manager

(0) votes

QGIS plugin for accessing harmonized space-time aggregat

About | Details | Versions

Version	Experimental	Minimum QGIS versio
1.1.0	yes	3.0.0
1.0.4	no	3.0.0
1.0.3	no	3.0.0
1.0.2	no	3.0.0
1.0.1	yes	3.0.0
1.0.0	yes	3.0.0



FOSS4G Prizren, 2023

Demonstration

Demonstration

- Import st_lucas package
- Define request

U411	Abandoned industrial areas
U412	Abandoned commercial areas
U413	Abandoned transport areas
U414	Abandoned residential areas
U415	Other abandoned areas
U420	Semi-natural and natural areas not in use

In [1]:
```
from st_lucas import LucasRequest, LucasIO
from owslib.fes import PropertyIsLike
```

In [2]:
```
request = LucasRequest()
request.countries = ['CZ']
request.years = [2018]
request.propertyname = 'LU1_H'
request.operator = PropertyIsLike
request.literal = 'U41%' # Abandoned areas
```

Demonstration

- Download data
- Investigate data using GeoPandas

```
In [3]: lucasio = LucasIO()
        lucasio.download(request)
        print("Number of LUCAS points:", lucasio.count())

        2023-06-29 12:03:51,994 - LUCAS - INFO - io.download -
        successfuly finished. Size of downloaded data: 144kb
        Number of LUCAS points: 30

In [4]: df = lucasio.to_geopandas()
        df.groupby(['lu1_h'])['lu1_h'].count()

Out[4]: lu1_h
        U411     4
        U412     1
        U414     2
        U415    23
        Name: lu1_h, dtype: int64
```

U411	Abandoned industrial areas
U412	Abandoned commercial areas
U413	Abandoned transport areas
U414	Abandoned residential areas
U415	Other abandoned areas
U420	Semi-natural and natural areas not in use

Demonstration

- Display photos for selected LUCAS points

In [7]:
```
import requests
from IPython.display import Image, display

for point in df[["point_id", "lu1_h"]].values[1:4]:
    images = lucasio.get_images(2018, point[0])
    r = requests.get(images["P"])
    print("ID:", point[0])
    print("LU1:", point[1])
    display(Image(r.content, width=400))
```

```
ID: 48262870
LU1: U415
```

U411	Abandoned industrial areas
U412	Abandoned commercial areas
U413	Abandoned transport areas
U414	Abandoned residential areas
U415	Other abandoned areas
U420	Semi-natural and natural areas not in use

Conclusions

Conclusions

- https://geoforall.fsv.cvut.cz/st_lucas/
- Harmonized (and space-time aggregated) LUCAS data easily accessible via
  - Python API or
  - QGIS plugin
- Real and unique ground-true **analysis ready** data
- Open source (MIT and GNU GPL): https://gitlab.com/geoharmonizer_inea/st_lucas
- Co-financed under Grant Agreement Connecting Europe Facility (CEF) Telecom project 2018-EU-IA-0095 by the European Union

Thank you for your attention!
Questions?