

CAD-Dokumentation zu GIS mit SpatiaLite migrieren

Christoph Rinne

05. August 2022

Inhaltsverzeichnis

Vorwort	2
1 Einführung	2
1.1 Verwendete Software & Informationen	2
1.2 Originaldaten	2
1.2.1 Ausgrabung	2
1.2.2 Digitalisierung in AutoCAD	3
1.3 Vorbereitung in AutoCAD	3
1.3.1 Export in DXF	3
1.3.2 Einheiten	4
1.3.3 Schraffuren zerlegen	4
1.3.4 Datenextraktion	5
2 SpatiaLite GUI	5
2.1 Datenkontrolle	5
2.1.1 line_layer_2d	5
2.1.2 poly_layer_2d	6
2.2 Linien zu Polygone	7
Literatur	8

Vorwort

Ziel ist die Überführung von Ausgrabungsplänen aus CAD-Dateien in ein GIS. Ausgang ist die Retrodigitalisierung (2D) einer Papierdokumentation einer über vier Jahre erfolgten Ausgrabung des Kollektivgrabes Odagsen 1, Stadt Einbeck, Ldkr. Northeim. Hierbei geht es nicht um einen schönen, interaktiven Plan in einem GIS am Ende, sondern um die Nachnutzung möglichst vieler Daten für eine räumliche Statistik.

Anmerkungen

- Menüpfade oder Abfolgen von Fenstern werden mit schlichten Pfeilen dargestellt: “Datei > Speichern”.
- Tastaturkürzel, die ich gerne nutze, stehen in Spitzklammern je Taste: <strg> + <c>.
- Schalter auf Formularen werden in `[]` gesetzt: [OK]
- Zur Darstellung von Befehlen im Text nutze ich die in Markdown übliche Darstellung von Code oder eben Anweisungen an den Computer: **anweisung**.
- Der Text enthält viele Links die auf Papier nicht funktionieren. Sparen Sie bitte Papier und verzichten Sie auf den Ausdruck.

```
DROP TABLE IF EXISTS 'block_line_2d';
DROP TABLE IF EXISTS 'insline_layer_2d';
DROP VIEW IF EXISTS 'insline_layer_2d_view';
DROP TABLE IF EXISTS 'line_layer_2d';
DROP TABLE IF EXISTS 'polyg_layer_2d';
DROP TABLE IF EXISTS 'text_layer_2d';
```

1 Einführung

1.1 Verwendete Software & Informationen

- OS Windows 10
- QGIS 3.22.4-Białowieża Quelle: [<https://qgis.org>]
- SpatiaLite SpatiaLite GUI 2.1.0 beta1, SpatiaLite 5.0.0, SQLite 3.33.0, Quelle [<http://www.gaia-gis.it>]
- AutoCAD 2010, Quelle für aktuelle *kostenlose* Schulversionen: [<https://www.autodesk.de/education/edu-software/overview>]
- SpatiaLite Cookbook html [<http://www.gaia-gis.it/gaia-sins/spatialite-cookbook/index.html>]
- SpatiaLite Funktionen [<http://www.gaia-gis.it/gaia-sins/spatialite-sql-5.0.0.html>]

AutoCAD ist eine sehr komplexe Software und Ausgrabungen können eine komplexe Struktur annehmen, die es zu dokumentieren gilt. Erwarten Sie nicht, dass die notwendige Kompetenz beim Erstellen der digitalen Daten stets vorhanden war, auch der Autor (Chr. Rinne) ist hier nur Autodidakt.

Rechnen Sie mit Fehlern im originalen Datenbestand und einer ggf. nicht optimalen Struktur oder erwarten Sie nicht die von Ihnen bevorzugte Struktur. Korrektur von Fehler und Anpassungen der Struktur erfolgen sicher am besten im originalen Arbeitsumfeld, also CAD.

Neben AutoCAD gibt es teils kostengünstigere Alternativen, u.a.:

- BricsCAD [<https://www.bricsys.com>]
- MegaCAD [<https://www.megacad.de/>]

1.2 Originaldaten

1.2.1 Ausgrabung

Die Daten stammen von der Ausgrabung und Auswertung des spätneolithischen Kollektivgrabes [Ogasen I](#), Stadt Einbeck, Ldkr. Northeim in Niedersachsen. Die Ausgrabung erfolgte in vier Kampagnen von 1981 bis 1984 als Forschungs- und Lehrgrabung des Institutes für Ur- und Frühgeschichte der Georg-August-Universität in Niedersachsen ([Rinne \(2003\)](#); [Heege und Heege \(1989\)](#)). In diesen Kampagnen

wurden zahlreiche Schnitte und eine wechselnde Anzahl von Plana angelegt als auch die dazwischen ursprünglich belassenen Profilstege sukzessive abgebaut und auf insgesamt 154, meist einzelnen und neu gerichteten Din A3-Blättern im M 1:20 dokumentiert. Zu den jeweiligen Planblättern wurden Überlieger auf Transparentpapier mit Nivellierwerte und weiteren Angaben angefertigt. Die Einmessung erfolgte mit Theodolit, Nivelliergerät und Maßband.

1.2.2 Digitalisierung in AutoCAD

Die Digitalisierung der Planzeichnungen erfolgte im Mai und Juni 1997 in AutoCAD Ver. 12 (DOS) und Ver. 13 (Windows 3.1) auf einem Din-A3-Grafiktablett und mit Referenzierung anhand der Koordinatenangaben auf den Blättern. Jede Datei erhielt eine stringent vergeben Namen ODS(chnitt) P(lanum) , z.B. ODS1P102. Für jedes Blatt wurde die Planumsangabe der Zeichnung, die Planumsangabe mit Bezug auf die Angabe der Ausgräberin, die Bearbeitungszeit, die mittlere Angabe der Nivellierwerte zum Fixpunkt der Oberfläche erfasst. Ergänzt wurden nachträglich die gängigen Metadaten der resultierenden Dateien.

datei	pl_lokal	pl_heege	zeit_h	niv_oberfl	byte	datum	uhrzeit	nr
ODS1P101	1	1	0.9	50	47219	29.04.97	11:37	1
ODS1P102	1	1	3.1	38	195786	29.04.97	9:54	2
ODS1P103	1	1	1.6	34	142651	29.04.97	10:06	3
ODS1P106	1	1	0.4	38	11904	27.04.97	7:19	6
ODS1P107	1	1	2.0	40	64278	29.04.97	11:53	7
ODS1P204	1	1	1.6	48	204959	29.04.97	11:36	4
ODS1P205	1	1	3.8	50	484742	29.04.97	11:46	5
ODS1P308	1	2	0.5	58	320690	29.04.97	11:59	8
ODS1P309	1	2	2.8	60	231455	29.04.97	12:06	9

Die Dateien sind einfach strukturiert. Folgende Layer wurden für Informationseinheiten verwendet: BEFUND, BEFUND_UNSICHER, BEFUNDSCHRAFF, FEUER, GRABUNGSGRENZE, GRENZE, KNOCHEN, KNOCHENSCHRAFF, STEINE, STEINESCHRAFF, TPROFIL. Alle Linien wurden als 2D-Polygone digitalisiert, allerdings wurden die Polygone nicht geschlossen, sondern bei der oft dichten Lage von Steinen und Knochen nur sauber an den gemeinsamen Punkten gefangen.

Als Störungen klassifizierte Befunde haben eine horizontale Linienschraffur, Sandsteine erhielten eine Punktschraffur und gebrannte Steine eine diagonale Schraffur auf dem Layer "FEUER", um diese Information zu vermitteln. Knochen wurden ausschließlich für den optischen Effekt stets schraffiert. Die Schraffuren wurden nicht je Objekt, sondern meist für zahlreiche Objekte angelegt, wodurch diese Schraffuren als ein Objekt über mehrere Steine oder Knochen laufen und der Mittelpunkt dieser Schraffur räumlich nicht mit einem Objekt zusammenhängt. Dies trifft vor allem auf Knochen zu, bei den eher singulären Sandsteinen oder im Verbund gebrannten Steinen ist ein räumlicher Kontext eher gegeben.

Symbole für Holzkohle, Rotlehm und verbrannte Knochen wurden als grafische Blöcke mit den Namen HK, RL, LB eingefügt. Diese können mit dem jeweiligen Datei-, Layer- und Blocknamen als auch den Koordinaten aus allen Zeichnungen eines Ordners in eine Tabelle exportiert werden (s.u. Vorbereitung in AutoCAD).

1.3 Vorbereitung in AutoCAD

1.3.1 Export in DXF

Für den Export aller DWG-Datei in DXF kann ein Script geschrieben und als Startoption an AutoCAD innerhalb eines Batch-Scriptes übergeben werden. Die Batch-Datei zum Starten von AutoCAD wird im Ordner der DWG-Dateien aufgerufen, wodurch das Arbeitsverzeichnis hier liegt und die Pfadangaben im Skript (*.scr) entfallen können.

```
REM Command to start ACAD with the script to convert all DWG files to DXF
"c:\Program Files\Autodesk\AutoCAD 2014\acad.exe" /b od_convert_dwg2dxf.scr
```

Das Script für AutoCAD wiederholt die Befehle für jede DWG-Datei und muss mit einer **Leerzeile** **enden**. Sollten Sie in der DWG Änderungen vornehmen (s.u.) und wollen diese auch speichern ergänzen Sie die den Befehl “*_qsave*”.

```
;; Script file for AutoCAD
;; Start AutoCAD on the command line with option: /b script-file.scr"
_open
ODS1P101.DWG
_saveas
dxf
16
ODS1P101.DXF
_close
_open
ODS1P102.DWG
...
<blank line>
```

1.3.2 Einheiten

AutoCAD kennt Einheiten (inch, mm, m etc) und rechnet diese automatisch ineinander um. Dies wird leider oft ignoriert, so dass DWG-Dateien in der Archäologie zwar in Metern gemessen sind, die Angabe zur Einheit aber auf dem Standard “Millimeter” steht oder sogar eventuell auf Inch (Britisch). Dies kann im Export-Script gleich mit angepasst werden, um die automatische Skalierung um den Faktor 1000 bei einem heterogenem Datenbestand zu vermeiden. Dazu im vorangehenden Code nach dem Öffnen der DWG-Datei und vor dem Speichern (*_saveas*) den folgenden Code einfügen. Hierbei steht die 6 für “Meter”, 5 für “Zentimeter” und 4 für “Millimeter”.

```
INSUNITS
6
```

Sollten die Zeichnungen darüber hinaus tatsächlich falsch skaliert sein, kann dies in einem Zug mit folgender Befehlsfolge im Script erledigt werden. Die Leerzeile nach “all” beendet die Objektwahl und “.01” ist durch den notwendigen Faktor zu ersetzen. Für die Optik können Sie noch ein “Zoom” “G” (Grenzen) ergänzen.

```
_scale
all

0,0,0
.01
```

1.3.3 Schraffuren zerlegen

Schraffuren kodieren oft Informationen, sind aber schlecht in ein GIS zu überführen. Werden Schraffuren in die zugehörigen Elemente, z.B. einzelne Linien zerlegt, handelt es sich um den Import einer schlichten Geometrie. In einem GIS kann dann mit eine räumliche Verbindung (*spatial join*) zwischen den unterschiedlichen Objekten hergestellt werden. Im vorliegenden Fall könnten dann alle Steine mit mindestens einem Linienmittelpunkt vom Layer “FEUER” als gebrannt markiert werden. Dazu muss vor dem Speichern (*_saveas*) folgender Code eingefügt werden.

```
(setq SS (ssget "x" '((0 . "hatch") (8 . "FEUER"))))
(if SS
  (progn
    (setq CNT 0)
    (repeat (sslength SS)
      (vl-cmdf "._explode" (ssname SS CNT))
      (setq CNT (1+ CNT))
    )
  )
)
```

<blank line>

Da dies nicht ganz selbsterklärend ist, eine knappe Erläuterung: Die erste Zeile definiert die Variable “ss” und weist dieser mit `ssget` aus der gesamten Datei “x” die Objekte zu, die der folgende Liste an Parametern entsprechen (*dotted pairs*, d.h. Attributkennziffer . Wert). Wenn die Variable “ss” Inhalt hat wird eine Abfolge (*progn*) von Anweisungen durchgeführt: 1. ein Zähler mit dem Startwert “0” definiert und dann auf alle Elemente der Auswahl “ss” der Befehl “*explode*” ausgeführt, wobei der jeweilige Objektname anhand des Zählers ermittelt wird.

1.3.4 Datenextraktion

In AutoCAD können aus einzelnen oder auch vielen Zeichnungen eines Ordners diverse Elemente mit deren Attributen als Liste exportiert werden (`_dataextraction`). Die Befehlsführung ist weitgehend intuitiv. In den einzelnen Fenstern kann die Auswahl an Elemente und Attribute durch entsprechende Anzeigeeoptionen bzw. Filter gesteuert werden. Im Beispiel Odagsen “Nur Blöcke Anzeigen” für die Auswahl von “HK”, “LB” und “RL”. Dann den Kategorienfilter auf “Allgemein”, “Geometrie” und “Zeichnung” setzen um dann nur die Attribute “Dateiname”, “Layer”, “Position x”, “Position Y” und “Position Z” zu wählen. **Wichtig:** der Export muss wegen der Punkt-Komma-Problematik als CSV-Datei gespeichert werden.

In diesem Fall erkennt der DXF-Import sowohl die Blockdefinitionen als auch die Einfügapunkte und listet diese korrekt (s.u.). Eine Datenextraktion ist deshalb nicht notwendig.

2 SpatiaLite GUI

Starten Sie die SpatiaLite GUI und erstellen Sie eine neue, leere Datenbank. In diesem Fall werden die vielen DXF-Dateien nicht einzeln, sondern der gesamte Ordner importiert: “Menu > Advanced > Import DXF drawings”. Wählen Sie dann nur eine DXF-Datei aus und ändern Sie im Importfenster dann die Angabe auf “(x) Import any DXF drawing file from selected folder”. Da ein lokales Koordinatensystem verwendet wurde belassen Sie SRID auf “-1”. Weitere Angaben: “(x) automatic 2D/3D”, “(x) mixed layers (distinct by type)”, “(X) none” für das *Ring handling* also das erkennen von sog. Donuts. Nach einer kurzen Wartezeit wurden folgende Tabellen und Sichten erstellt:

- `block_line_2d`: die Linien der grafischen Blockdefinitionen in jeder Datei.
- `inline_layer_2d`: Eine Liste der eingefügten Blöcke in jeder Datei, u.a. mit Datei-, Layer und Blocknamen als auch x, y und z-Koordinate des Einfügapunktes.
- `inline_layer_2d_view`: Die Kombination der beiden vorgenannten Dateien in einer Sicht, die im vorliegenden Fall zwar offensichtlich korrekte Geometrien enthält, in QGIS im Kartenfenster aber dennoch nicht dargestellt wird.
- `line_layer_2d`: sehr viele Linien der diversen Objekte (Steine, Knochen etc.) mit jeweiligem Datei- und Layernamen.
- `polyg_layer_2d`: Deutlich weniger Polygone mit jeweiligem Datei- und Layernamen.
- `text_layer_2d`: Die Texte in den DXF-Dateien, z.B. Befund und Profilnummern, mit dem zugehörigen Einfügapunkt, Datei- und Layernamen.

2.1 Datenkontrolle

Es folgt eine Datenkontrolle mit Überarbeitung, die vor allem die Geometrien betrifft. Hier sind teils durch das unsaubere Digitalisieren doppelte Knoten vorhanden oder einige Polygone überschneiden sich selbst.

2.1.1 line_layer_2d

Zahlreiche **fehlerhafte Geometrien** können über das Kontextmenü repariert werden: Spalte “geometry” Kontextmenü > “Malformed geometries”. Liefert 501 fehlerhafte Geometrien: 1. überwiegend wiederholter Knoten (*repeated vertex*), 2. fehlerhafte Geometrie durch zu wenig Punkte. Bestätigen Sie [Repair], um das erste Problem direkt zu lösen. Sollten Sie die Daten prüfen und von Hand korrigieren wollen, dann führen Sie folgende Befehle nacheinander aus.

```
-- Eine Geometrie mit Fehler zur Kontrolle ansehen
Select astext(geometry) from line_layer_2d
  where feature_id = 32;
-- Die Korrektur ansehen
Select astext(sanitizegeometry(geometry)) from line_layer_2d
  where feature_id = 32;
-- Die Korrektur ausführen
Update line_layer_2d
  set geometry = sanitizegeometry(geometry)
  where isvalid(geometry) = 0;
```

Es bleiben 78 fehlerhafte Geometrien mit “Repeated vertex. Too few points in geometry ...”. Die Kontrolle der Geometrie von Feature 412 zeigt drei identische Punkte. Veralgemeinernd können wir mit folgender Abfrage diese fehlerhaften erst finden und dann auch löschen:

```
-- Fehlerhafte Linien aus identischen Punkten finden.
select * from line_layer_2d
where length(geometry) = 0 and isvalid(geometry) = 0;
-- Löschen dieser Linien
Delete from line_layer_2d
where length (geometry) = 0 and isvalid(geometry) = 0;
```

Weitere Merkmale von möglicherweise fehlerhafter Linien können gesucht und ggf. gelöscht werden.

Eine Linie besteht nur aus drei Punkten, wobei Start und Endpunkt identisch sind.

```
Select feature_id, layer, numpoints(geometry)
from line_layer_2d
where StartPoint(geometry) = EndPoint(geometry) and numpoints(geometry) = 3;
```

Je nach Kontext könnten Linien mit zwei Punkten oder besonders kurze Linien auch weniger plausibel sein.

Abschließend ein Überblick über den Datenbestand:

```
SELECT Count(*), layer, GeometryType("geometry")
FROM "line_layer_2d"
GROUP BY 2, 3;
```

2.1.2 poly_layer_2d

Auch hier zuerst **fehlerhafte Geometrien** über das Kontextmenü reparieren: Spalte “geometry” Kontextmenü > “Malformed geometries”: Wiederholte Knoten (*repeated vertex*) reparieren. Ringe mit weniger als 4 Punkten entsprechen den Linien mit identischem Start- und Endpunkt. fehlerhafte Geometrie durch zu wenig Punkte. Bestätigen Sie [Repair], um das erste Problem direkt zu lösen. Die zugehörige Funktion ist “ST_MakeValid(geom)”. Sollten Sie die weiteren fehlerhaften Daten prüfen und von Hand korrigieren wollen, dann führen Sie folgende Befehle nacheinander aus.

```
select feature_id, layer, astext(geometry)
from polyg_layer_2d
where isvalid(geometry) = 0 and st_npoints(geometry) < 4;
-- Löschen mit Delete from ...
```

Es bleiben noch die sich überschneidenden Polygone, zur visuellen Kontrolle

```
select feature_id, layer, st_npoints(geometry), astext(geometry)
from polyg_layer_2d
where isvalid(geometry) = 0
```

Die Polygone können mit der Funktion “ST_RingsCutAtNodes(geom)” in ein Multilinienobjekt zerteilt werden. Etwas einfacher und eventuell auch erfolgreich ist die Anwendung der Funktion “MakeValid(geometry)”, die ein Multipolygon zurückgibt. Um keinen gemischten Geometrien zu erhalten sollte die Geometrie von polyg_layer_2d angepasst werden.

```
update polyg_layer_2d
set geometry = CastToMulti(MakeValid(geometry));
```

Zur Kontrolle und falls nur einfache Polygone gewünscht sein sollten:

```
SELECT *, st_numgeometries(geometry)
FROM "polyg_layer_2d"
where st_numgeometries(geometry) > 1;
```

Die Fälle mit genau zwei Subpolygonen können getrennt werden.

```
with recursive cnt(x) as
(select 1
 union all
 select x+1 from cnt limit 2)
select filename, layer, CastToMulti(ST_GeometryN(geometry, x)) as geometry
from cnt, polyg_layer_2d as b
where ST_NumGeometries(b.geometry) = 2;
```

Um diese Multipolygone mit genau zwei Subpolygonen getrennt anzufügen muss eine *“insert into”*-Anweisung eingefügt werden. Danach noch die ursprünglichen Polygone löschen und die Tabellen der Datenbank aktualisieren.

```
with recursive cnt(x) as
(select 1
 union all
 select x+1 from cnt limit 2)
insert into polyg_layer_2d (filename, layer, geometry)
select filename, layer, CastToMulti(ST_GeometryN(geometry, x)) as geometry
from cnt, polyg_layer_2d as b
where ST_NumGeometries(b.geometry) = 2;
```

```
Delete from polyg_layer_2d
where ST_NumGeometries(geometry) = 2;
```

```
Select UpdateLayerStatistics();
```

Zum Abschluss der Prüfung eine Übersicht zum Datenbestand:

```
SELECT Count(*), layer, GeometryType("geometry")
FROM "polyg_layer_2d"
GROUP BY 2, 3;
```

2.2 Linien zu Polygone

Sehr viele Steine und Knochen wurden nicht als Polygone erkannt.

inien Folgendes durch st_buildarea()

```
Select 'POLYGON(' || substr(astext(geometry),11)||')'
from line_layer_2d
where feature_id = 28

INSERT INTO polyg_layer_2d (filename, layer, geometry)
Select filename, layer,
ST_GeomFromText('POLYGON(' || substr(astext(geometry),11)||')
', -1)
from line_layer_2d
where StartPoint(geometry) = EndPoint(geometry)

SELECT UpdateLayerStatistics('polyg_layer_2d', 'geometry')

DELETE from line_layer_2d
```

```
where StartPoint(geometry) = EndPoint(geometry)
SELECT UpdateLayerStatistics('line_layer_2d', 'geometry')
```

Kontrolle der verbleibenden Linien

```
select count(feature_id), layer
from line_layer_2d
group by 2
```

Befunde, Grabungsgrenze und Profile hier erstmal ignorieren, Ziel ist räumliche Statistik. Deshalb für Knochen und Steine möglichst viele Polygone schaffen:

```
select feature_id, numpoints(geometry), st_length(geometry)
from line_layer_2d
where layer = 'KNOCHEN'
order by 2, 3
```

Viele Linien mit nur 2 Punkten, maximal 13 cm lang und drei Punkten bis 18 cm lang. Visuelle Kontrolle der Objekte zeigt vielfach an einer kurzen Kante offene Objekte, die einfach geschlossen werden kann. Dazu oft auch langgestreckte Objekte, die nicht geschlossen werden können, da z.B. eine Überschneidung in sich erfolgen würde.

```
create table polyg_tmp as
select feature_id, filename, layer,
st_GeomFromText('POLYGON(' || substr(astext(geometry),11,length(astext(geometry))-11)
|| ', ' || substr(astext(StartPoint(geometry)), 7) || ')', -1) as geometry
from line_layer_2d;

-- recover geometry column
-- prüfen der Geometrie
begin transaction;
SELECT Count(*), GeometryType("geometry"), Srid("geometry"), CoordDimension("geometry")
FROM "polyg_tmp"
GROUP BY 2, 3, 4;
-- indexliste für später erstellen
create temp table polyg_null_tmp as
SELECT feature_id
FROM polyg_tmp
where geometry is null;
-- nuller löschen
DELETE FROM polyg_tmp
where geometry is null;
-- alles löschen, was nicht knochen oder Stein ist
DELETE FROM polyg_tmp
where layer not in ('KNOCHEN', 'STEINE');
commit
```

```
create temp table poly_invalidgeom_tmp as
select feature_id
from polyg_tmp
where isvalid(geometry) = 0;
```

```
delete from polyg_tmp
where isvalid(geometry) = 0;
```

Sichtkontrolle zeigt weiterhin einige seltsame, sehr langschmale Objekte.

Liefert am Ende erfolgreiche und fehlerhafte Geometrien. -> visuelle Kontrolle

Literatur

Heege, E., Heege, A., 1989. Die Häuser Der Toten. Jungsteinzeitliche Kollektivgräber Im Ldkr. Northeim, Wegweiser Zur Vor- Und Frühgeschichte Niedersachsens. Hildesheim.

Rinne, C., 2003. Odagsen Und Großenrode, Ldkr. Northeim. Jungsteinzeitliche Kollektivgräber Im Südlichen Leinetal, Beiträge Zur Archäologie in Niedersachsen. Marie Leidorf, Rahden/Westf.