GDAL 2.0 overview

Even Rouault

SPATIALYS

July 15h 2015

Plan

- Introduction to GDAL/OGR
- Community
- GDAL 2.0 features
- GDAL 2.1 preview
- Potential future directions
- Discussion

About me

- Contributor to the GDAL/OGR project since 2007 chair of the PSC since 2015.
- Contributor to MapServer
- Co-maintainer: libtiff, libgeotiff, PROJ.4
- Funder of Spatialys, consulting company around geospatial open source technologies, with a strong focus on GDAL and MapServer

GDAL/OGR Introduction

- Geospatial Data Abstraction Library
- Raster (GDAL) and Vector (OGR)
- Read/write access to more than 200 geospatial formats and protocols
- Widely used (FOSS & proprietary): GRASS, MapServer, Mapnik, QGIS, PostGIS, OTB, SAGA, FME, ArcGIS, Google Earth…

  (93 referenced at http://trac.osgeo.org/gdal/wiki/SoftwareUsingGdal)

- Started in 1998 by Frank Warmerdam
- A project of OSGeo since 2008
- MIT/X Open Source license (permissive)
- 1.2 M physical lines of code (175 kLOC from external libraries) for library, utilities and language bindings
- 166 kLOC for Python autotest suite. 62% line coverage

Main features

- Support datasets of arbitrary size with limited resources
- C++ library with C API
- Multi OS: Linux, Windows, MacOSX/iOS, Android, ...
- Language bindings: Python, Perl, C#, Java,...
- OGC WKT coordinate systems, and PROJ.4 as reprojection engine
- Format support through drivers implemented a common interface
- Utilities for translation,reprojection, subsetting, mosaicing, tiling…
- Can file with local, remote (/vsicurl), compressed (/vsizip/, /vsigzip/, /vsitar), in-memory (/vsimem) files

Raster Features

- Efficient support for large images (tiling, overviews)
- Block caching
- Warping engine
- Various algorithms: rasterization, vectorization (polygonization / contour), interpolation of void cells, sieve filter
- Utilities for translation & subsetting (gdal_translate), warping/reprojection (gdalwarp), pyramid building (gdaladdo), mosaicing (gdalwarp/gdalbuildvrt), indexing (gdaltindex), tiling (gdal_retile, gdal2tiles), DEM processing (gdaldem), grid interpolation, ...

Raster Formats

- Plain Raster (images): JPEG, PNG, GIF, WebP, BPG ...
- Geospatial Files: GeoTIFF, .img, NITF, ...
- Wavelet: JPEG 2000, ECW, MrSID, ...
- RDBMS: Oracle Raster, PostGIS Raster, Rasdaman
- Portable database: Rasterlite, MBTiles, GeoPackage NEW
- Web Service: WMS, WCS
- Radar: CEOS, Envisat
- Elevation: DTED, USGS DEM, SRTM HGT
- Containers: HDF4, HDF5, NetCDF
- Other: Geospatial PDF
- GDAL specific: in-memory, VRT

⇒ 137 Formats

Vector Features

- OGC/ISO geometry model
- Commandline utilities for translation (ogr2ogr), query (ogrinfo), indexing (ogrtindex)
- Spatial filter
- GEOS for geometry operations
- SQL:
  - OGR SQL or SQLite engine for all formats
  - Passthrough SQL for RDBMS

Vector Formats

- GIS: Shapefile, MapInfo, ESRI Personal/File Geodatabase
- CAD: DXF, DWG, DGN (pre-V8)
- RDBMS: PostGIS, Oracle, MySQL, Ingres, MSSQL, ODBC
- Portable databases: SQLite/Spatialite, GeoPackage
- Exchange: KML (*GDAL 1.11:reference implementation*) GML, GeoJSON

CERTIFIED OGC® COMPLIANT RI KML 2.2.0

- Web Service: WFS, Fusion Tables, CartoDB, CouchDB, NEW Cloudant, GME
- National: SDTS, NAS, NTF, TIGER/Line, Interlis, VFK, Edigeo, SOSI, SXF, MTK GML, RUIAN GML, INSPIRE Cadastral GML
- Non spatial : CSV, XLS, XLSX, ODS
- GDAL specific: in-memory, VRT

⇒ 81 formats

Community

- 57 SVN committers registered
  - 17 active in last 12 months + 42 non-committer contributors
  - https://www.openhub.net/p/gdal
- 2200 gdal-dev list members. 2900 messages in last 12 months, 759 discussion threads, 74% with at least one answer
- About 500 tickets filed in last 12 months (5850 over history). ~580 opened, after ~200 closed as a winter cleanup (special thanks to Jukka Rahkonen!)

GDAL/OGR 2.0 Release

- Released in June 2015 (1.11.0 in April 2014)
- 10 RFCs implemented over GDAL 2.0 cycle
  - Trade-off: new features vs backward compatibility
  - Application code and out-of-tree drivers will need updates, mostly on the vector side. See MIGRATION_GUIDE.TXT
- 11 new drivers :
  - BPG (Better Portable Graphics)
  - GeoPackage Raster
  - KEA
  - PLMosaic + PLScenes (Planet Labs mosaics and scenes)
  - ROI_PAC (NASA JPL format)
  - VICAR
  - Cloudant
  - JML (OpenJUMP)
  - Selaphin (Telemac hydraulic model)
  - CSW (Catalog Service for the Web)

RFC 46: GDAL/OGR unification

- A long-time dream come true!
- Raster and vector drivers unified :
  - handled by the same registrar/manager
  - vector drivers can have metadata
  - efficient driver identification
- Raster and vector datasets unified :
  - a dataset can have both raster bands and vector layers.
  - vector datasources&layers can have metadata
- Unified drivers: PCIDSK, PDF, GeoPackage

RFC 46: GDAL/OGR unification

- Open options (to avoid global configuration options)
- C API backward compatibility preserved.
- C++ API impacted. e.g. OGRSFDriverRegistrar::Open() replaced by GDALOpenEx()

More at https://trac.osgeo.org/gdal/wiki/rfc46_gdal_ogr_unification

RFC 46: GDAL/OGR unification

```
$ ogrinfo --format postgresql
Format Details:
  Short Name: PostgreSQL
  Long Name: PostgreSQL/PostGIS
  Supports: Vector
  Help Topic: drv_pg.html
  Supports: Open() - Open existing dataset.
  Supports: Create() - Create writeable dataset.
  Creation Field Datatypes: Integer Integer64 Real String Date DateTime Time IntegerList Integer64List RealList StringList Binary
  Supports: Creating fields with NOT NULL constraint.
  Supports: Creating fields with DEFAULT values.
  Supports: Creating geometry fields with NOT NULL constraint.

<CreationOptionList />


<OpenOptionList>
  <Option name="LIST_ALL_TABLES" type="boolean" description="Whether all tables, including non-spatial ones, should be listed" default="NO" />
</OpenOptionList>
```

RFC 46: GDAL/OGR unification

```
<LayerCreationOptionList>
  <Option name="GEOM_TYPE" type="string-select" description="Format of geometry columns" default="geometry">
            <Value>geometry</Value>
            <Value>geography</Value>
  </Option>
  <Option name="OVERWRITE" type="boolean" description="Whether to overwrite an existing table with the layer name to be created" default="NO" />
  <Option name="LAUNDER" type="boolean" description="Whether layer and field names will be laundered" default="YES" />
  <Option name="PRECISION" type="boolean" description="Whether fields created should keep the width and precision" default="YES" />
  <Option name="DIM" type="integer" description="Set to 2 to force the geometries to be 2D, or 3 to be 2.5D" />
  <Option name="GEOMETRY_NAME" type="string" description="Name of geometry column. Defaults to wkb_geometry for GEOM_TYPE=geometry or the_geog for GEOM_TYPE=geography" />
  <Option name="SCHEMA" type="string" description="Name of schema into which to create the new table" />
  <Option name="SPATIAL_INDEX" type="boolean" description="Whether to create a spatial index" default="YES" />
  <Option name="TEMPORARY" type="boolean" description="Whether to a temporary table instead of a permanent one" default="NO" />
  <Option name="NONE_AS_UNKNOWN" type="boolean" description="Whether to force non-spatial layers to be created as spatial tables" default="NO" />
  <Option name="FID" type="string" description="Name of the FID column to create" default="ogc_fid" />
  <Option name="FID64" type="boolean" description="Whether to create the FID column with BIGSERIAL type to handle 64bit wide ids" default="NO" />
  <Option name="EXTRACT_SCHEMA_FROM_LAYER_NAME" type="boolean" description="Whether a dot in a layer name should be considered as the separator for the schema and table name" default="YES" />
</LayerCreationOptionList>
```

RFC 46: GDAL/OGR unification

```
$ gdalinfo --format gpkg
Format Details:
  Short Name: GPKG
  Long Name: GeoPackage
  Supports: Raster
  Supports: Vector
  Extension: gpkg
  Help Topic: drv_geopackage.html
  Supports: Subdatasets
  Supports: Open() - Open existing dataset.
  Supports: Create() - Create writeable dataset.
  Supports: CreateCopy() - Create dataset by copying another.
  Supports: Virtual IO - eg. /vsimem/
  Creation Datatypes: Byte
  Creation Field Datatypes: Integer Integer64 Real String Date DateTime Binary
  Supports: Creating fields with NOT NULL constraint.
  Supports: Creating fields with DEFAULT values.
  Supports: Creating geometry fields with NOT NULL constraint.
```

RFC 46: GDAL/OGR unification

```
<CreationOptionList>
  <Option name="RASTER_IDENTIFIER" type="string" description="Human-readable identifier (e.g. short name)" />
  <Option name="RASTER_DESCRIPTION" type="string" description="Human-readable description" />
  <Option name="BLOCKSIZE" type="int" description="Block size in pixels" default="256" max="4096" />
  <Option name="TILE_FORMAT" type="string-select" description="Format to use to create tiles" default="PNG_JPEG">
            <Value>PNG_JPEG</Value>
            <Value>PNG</Value>
            <Value>PNG8</Value>
            <Value>JPEG</Value>
            <Value>WEBP</Value>
[...]
  </Option>


<LayerCreationOptionList>
  <Option name="GEOMETRY_NAME" type="string" description="Name of geometry column." default="geom" deprecated_alias="GEOMETRY_COLUMN" />
  <Option name="GEOMETRY_NULLABLE" type="boolean" description="Whether the values of the geometry column can be NULL" default="YES" />
  <Option name="PRECISION" type="boolean" description="Whether text fields created should keep the width" default="YES" />
  <Option name="SPATIAL_INDEX" type="boolean" description="Whether to create a spatial index" default="YES" />
[...]
</LayerCreationOptionList>
```

RFC 51: RasterIO improvements

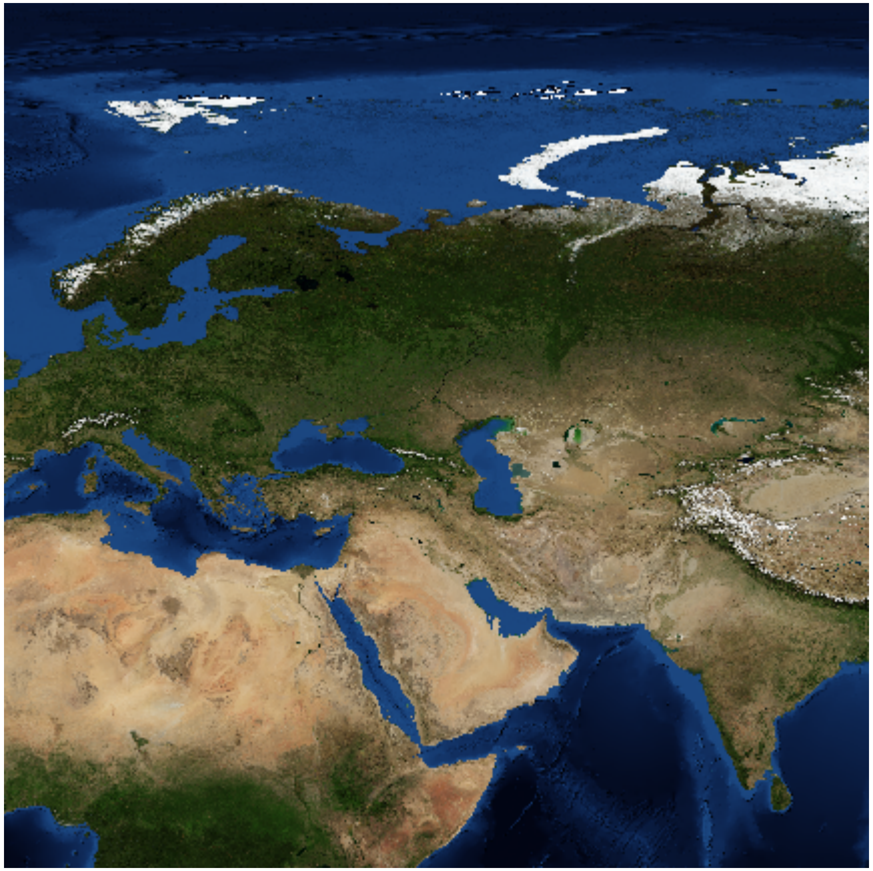
- GDAL raster band or dataset RasterIO methods can now accept a resampling method:
    - Nearest neighbour (default, fastest but not pretty)
    - Bilinear
    - Bicubic (Catmull Rom)
    - Cubic-spline
    - Lanczos
    - mode (majority)
    - Gauss

  gdal_translate -r, GDALRasterIOEx(), GDALDatasetRasterIOEx()
- Intel SSE-2 acceleration of most of those methods available
- Progress/cancel callback available
- 64-bit spacings in user buffer possible

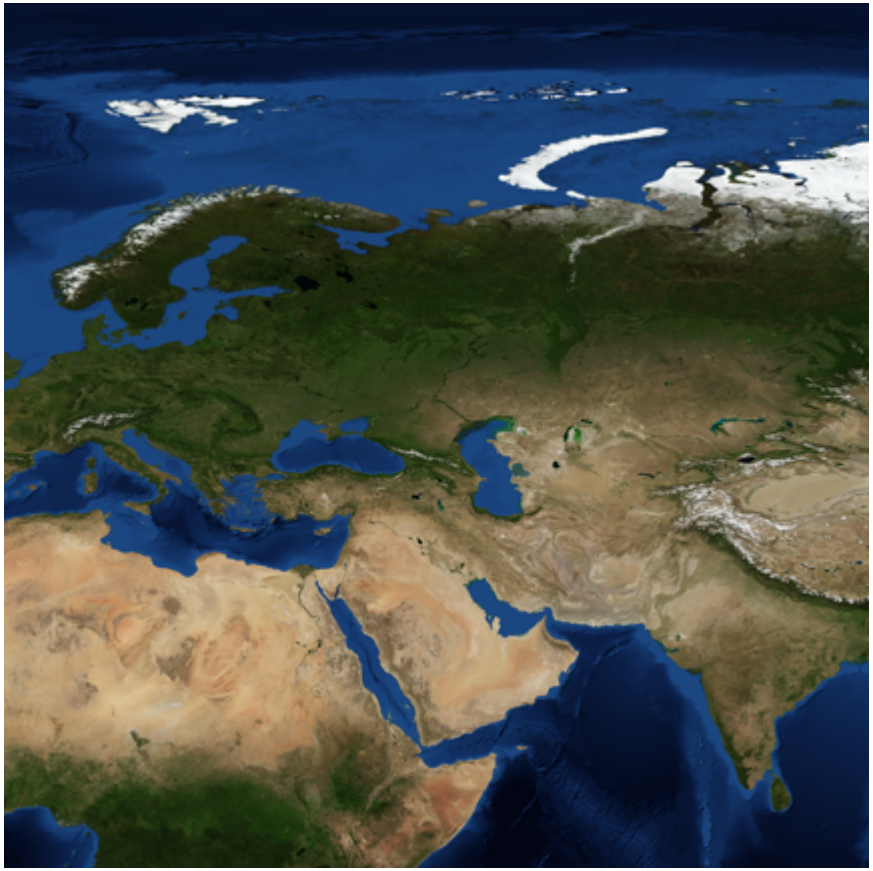
More at https://trac.osgeo.org/gdal/wiki/rfc51_rasterio_resampling_progress

RFC 51: RasterIO improvements

- 2% reduction of a 21600x21600 uncompressed tiled GeoTIFF BMNG



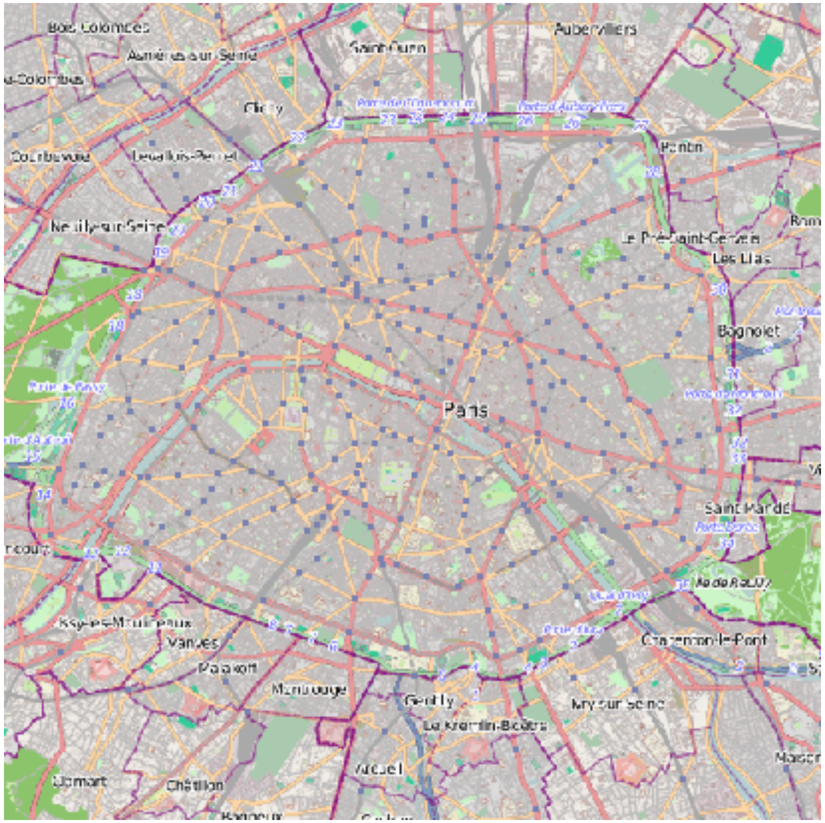
Nearest neighbour, 45s total (5s CPU)



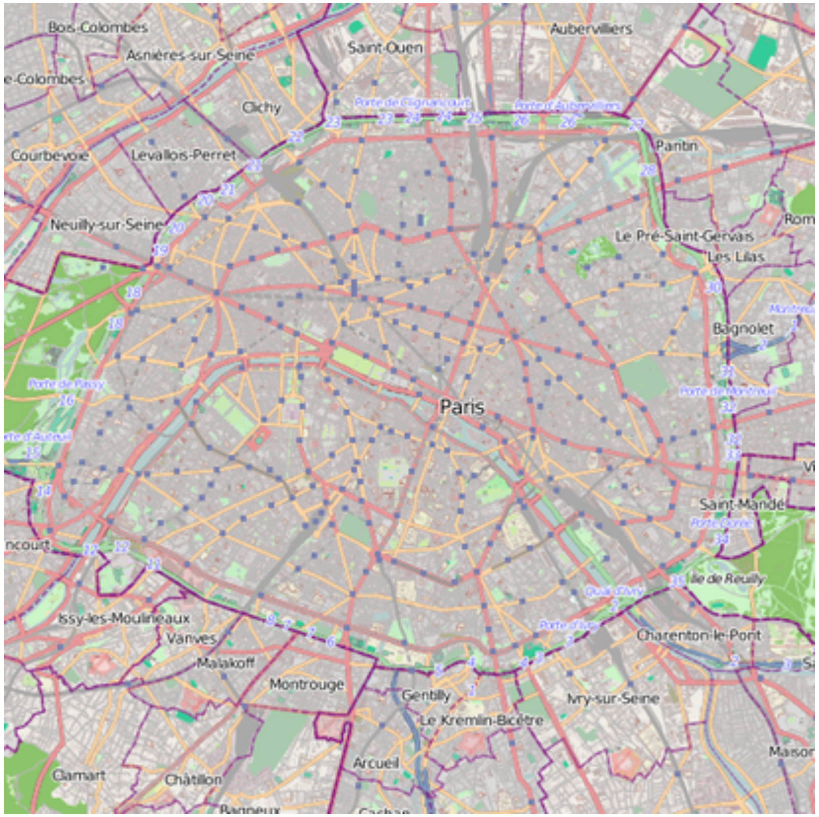
Bicubic, 1m5s total (15s CPU)

RFC 51: RasterIO improvements

- 75% reduction of a OpenStreetMap extract



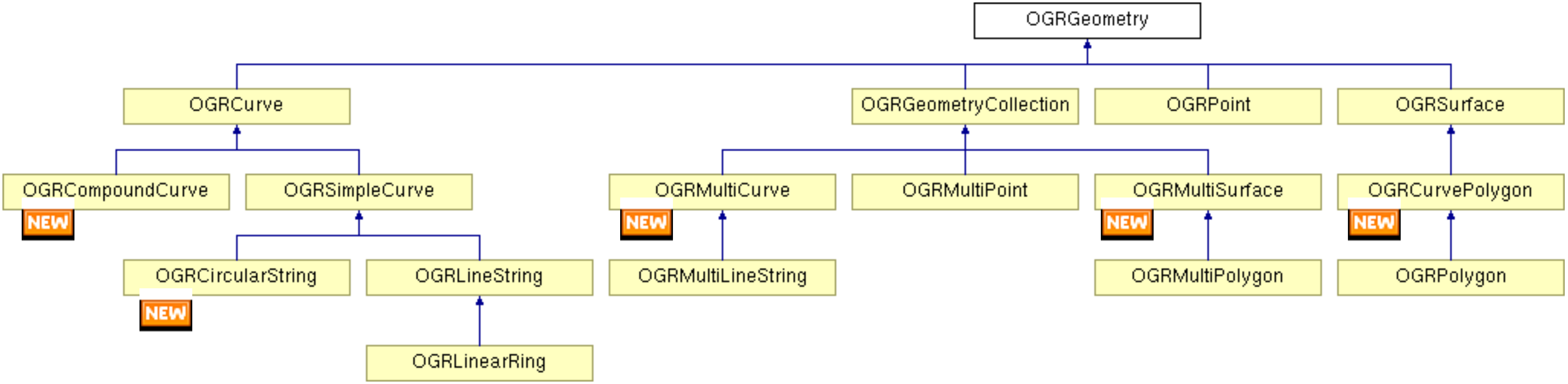
Nearest neighbour, ~100ms



Lanczos, ~100ms

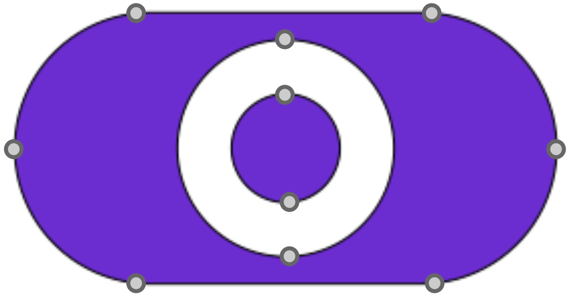
RFC 49: Curve geometries

- GDAL 1.X : OGC Simple Feature Access Part 1 v1.1.0
- GDAL 2.0: ISO SQL/MM Part 3 / OGC SFA v1.2.1



⇒ 5 new geometry types : CircularString, CompoundCurve, MultiCurve, CurvePolygon, MultiSurface

RFC 49: Curve geometries



Exemple:

```
MULTISURFACE(
        CURVEPOLYGON(
                COMPOUNDCURVE(  CIRCULARSTRING(5 0,0 5,5 10),
                                (5 10,15 10),
                                CIRCULARSTRING(15 10,20 5,15 0),
                                (15 0, 5 0)),
                CIRCULARSTRING(10 1,10 9,10 1)),
        CURVEPOLYGON(CIRCULARSTRING(10 3,10 7,10 3)))
```

RFC 49: Curve geometries

- What about spatial operations ?

```
g1 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0
0,1 1,2 0),(2 0,0 0)))')
g2 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0
0,1 -1,2 0),(2 0,0 0)))')
g3 = g1.Union(g2)
```

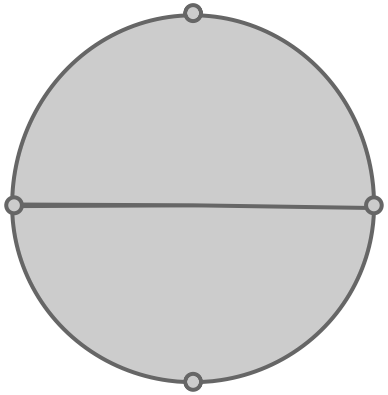
g3 = ???????????????

RFC 49: Curve geometries

- What about spatial operations ?

```
g1 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0 0,1 1,2 0),(2 0,0 0)))')
g2 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0 0,1 -1,2 0),(2 0,0 0)))')
g3 = g1.Union(g2)
```

g3 = ???????????????

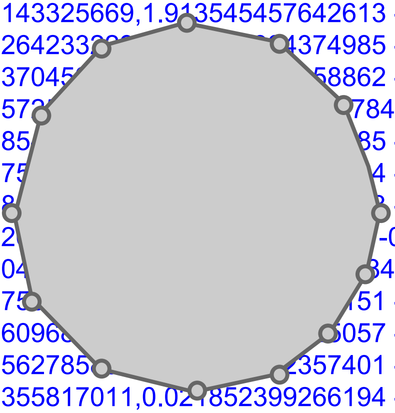


RFC 49: Curve geometries

```
POLYGON ((0 0,0.002435949740176 0.069756473744125,0.009731931258429 0.13917310096061,0.021852399266194
0.20791169081776,0.038738304061681 0.275637355817011,0.060307379214091 0.342020143325669,0.086454542357401
0.406736643075803,0.117052407141074 0.469471562785898,0.151951903843575 0.529919264233229,0.190983005625057
0.587785252292491,0.233955556881021 0.642787609686564,0.280660199661355 0.694658370459024,0.330869393641151
0.743144825477401,0.384338524674348 0.788010753606727,0.440807096529255 0.829037572555052,0.5
0.866025403784448,0.561628853210948 0.898794046299173,0.625393406584095 0.927183854566806,0.690983005625071
0.951056516295154,0.75807810440034 0.970295726276021,0.826351822333095 0.984807753012234,0.895471536732373
0.99452189536828,0.965100503297521 0.999390827019113,1.034899496702508 0.999390827019113,1.104528463267684
0.99452189536828,1.173648177666962 0.984807753012234,1.241921895599716 0.970295726276021,1.309016994374985
0.951056516295182,1.374606593415933 0.927183854566806,1.438371146789109 0.898794046299173,1.5
0.866025403784448,1.559192903470773 0.829037572555052,1.61566147532568 0.788010753606727,1.669130606358862
0.743144825477401,1.719339800338673 0.694658370459024,1.766044443119028 0.642787609686564,1.809016994374985
0.587785252292491,1.848048096156446 0.529919264233229,1.882947592858954 0.469471562785898,1.913545457642613
0.406736643075803,1.939692620785934 0.342020143325669,1.961261695938333 0.275637355817011,1.978147600733848
0.20791169081776,1.990268068741613 0.139173100960068,1.997564050259825 0.069756473744128,2 0,1.997564050259825
-0.069756473744128,1.990268068741613 -0.139173100960068,1.978147600733848 -0.20791169081776,1.961261695938333
-0.275637355817011,1.939692620785934 -0.342020143325669,1.913545457642613 -0.406736643075803,1.882947592858954
-0.469471562785898,1.848048096156446 -0.529919264233229,1.809016994374985 -0.587785252292491,1.766044443119028
-0.642787609686564,1.719339800338673 -0.694658370459024,1.669130606358862 -0.743144825477401,1.61566147532568
-0.788010753606727,1.559192903470773 -0.829037572555052,1.5 -0.866025403784448,1.438371146789109
-0.898794046299173,1.374606593415933 -0.927183854566806,1.309016994374985 -0.951056516295182,1.241921895599716
-0.970295726276021,1.173648177666962 -0.984807753012234,1.104528463267684 -0.99452189536828,1.034899496702508
-0.999390827019113,0.965100503297521 -0.999390827019113,0.895471536732373 -0.99452189536828,0.826351822333095
-0.984807753012234,0.75807810440034 -0.970295726276021,0.690983005625071 -0.951056516295154,0.625393406584095
-0.927183854566806,0.561628853210948 -0.898794046299173,0.5 -0.866025403784448,0.440807096529255
-0.829037572555052,0.384338524674348 -0.788010753606727,0.330869393641151 -0.743144825477401,0.280660199661355
-0.694658370459024,0.233955556881021 -0.642787609686564,0.190983005625057 -0.587785252292491,0.151951903843575
-0.529919264233229,0.117052407141074 -0.469471562785898,0.086454542357401 -0.406736643075803,0.060307379214091
-0.342020143325669,0.038738304061681 -0.275637355817011,0.021852399266194 -0.20791169081776,0.009731931258429
-0.13917310096061,0.002435949740176 -0.069756473744125,0 0))
```

RFC 49: Curve geometries

```
POLYGON ((0 0,0.002435949740176 0.069756473744125,0.009731931258429 0.139173100960061,0.021852399266194
0.20791169081776,0.038738304061681 0.275637355817011,0.060307379214091 0.342020143325669,0.086454542357401
0.406736643075803,0.117052407141074 0.469471562785898,0.151951903843575 0.529919264233229,0.190983005625057
0.587785252292491,0.233955556881021 0.642787609686564,0.280660199661355 0.694658370459024,0.330869393641151
0.743144825477401,0.384338524674348 0.788010753606727,0.440807096529255 0.829037572555052,0.5
0.866025403784448,0.561628853210948 0.898794046299173,0.625393406584095 0.927183854566806,0.690983005625071
0.951056516295154,0.75807810440034 0.970295726276021,0.826351822333095 0.984807753012234,0.895471536732373
0.99452189536828,0.965100503297521 0.999390827019113,1.034899496702508 0.999390827019113,1.104528463267684
0.99452189536828,1.173648177666962 0.984807753012234,1.241921895599716 0.970295726276021,1.309016994374985
0.951056516295182,1.374606593415933 0.927183854566806,1.438371146789109 0.898794046299173,1.5
0.866025403784448,1.559192903470773 0.829037572555052,1.61566147532568 0.788010753606727,1.669130606358862
0.743144825477401,1.719339800338673 0.694658370459024,1.766044443119028 0.642787609686564,1.809016994374985
0.587785252292491,1.848048096156446 0.529919264233229,1.882947592858954 0.469471562785898,1.913545457642613
0.406736643075803,1.939692620785934 0.342020143325669,1.961261695938333 0.275637355817011,1.978147600733848
0.20791169081776,1.990268068741613 0.139173100960068,1.997564050259825 0.069756473744128,2 0,1.997564050259825
-0.069756473744128,1.990268068741613 -0.139173100960068,1.978147600733848 -0.20791169081776,1.961261695938333
-0.275637355817011,1.939692620785934 -0.342020143325669,1.913545457642613 -0.406736643075803,1.882947592858954
-0.469471562785898,1.848048096156446 -0.529919264233                    4374985 -0.587785252292491,1.766044443119028
-0.642787609686564,1.719339800338673 -0.69465837045                     58862 -0.743144825477401,1.61566147532568
-0.788010753606727,1.559192903470773 -0.829037572                     784448,1.438371146789109
-0.898794046299173,1.374606593415933 -0.92718385                      85 -0.951056516295182,1.241921895599716
-0.970295726276021,1.173648177666962 -0.98480775                      4 -0.99452189536828,1.034899496702508
-0.999390827019113,0.965100503297521 -0.9993908                         -0.99452189536828,0.826351822333095
-0.984807753012234,0.75807810440034 -0.97029572                         -0.951056516295154,0.625393406584095
-0.927183854566806,0.561628853210948 -0.89879404                      34448,0.440807096529255
-0.829037572555052,0.384338524674348 -0.78801075                      151 -0.743144825477401,0.280660199661355
-0.694658370459024,0.233955556881021 -0.64278760966                   5057 -0.587785252292491,0.151951903843575
-0.529919264233229,0.117052407141074 -0.46947156278                   2357401 -0.406736643075803,0.060307379214091
-0.342020143325669,0.038738304061681 -0.275637355817011,0.021852399266194 -0.20791169081776,0.009731931258429
-0.139173100960061,0.002435949740176 -0.069756473744125,0 0))
```

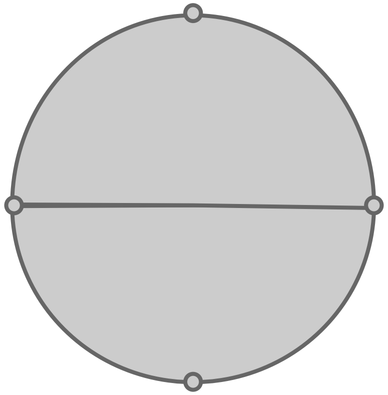


RFC 49: Curve geometries

- What about spatial operations ?

```
g1 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0 0,1 1,2 0),(2 0,0 0)))')
g2 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0 0,1 -1,2 0),(2 0,0 0)))')
g3 = g1.Union(g2)
```

g3 = ???????????????

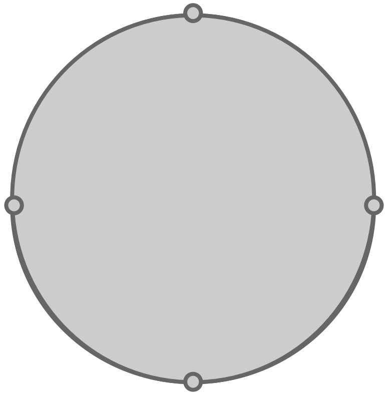


Try again….

RFC 49: Curve geometries

- Conversion methods between non-curve and curve geometry types, explicit or implicit (e.g. GEOS based operations)

```
g1 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0 0,1 1,2 0),(2 0,0 0)))')
g2 = ogr.CreateGeometryFromWkt('CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING (0 0,1 -1,2 0),(2 0,0 0)))')
g3 = g1.Union(g2)
```



CURVEPOLYGON (CIRCULARSTRING (0 0,1 1,2 0,1 -1,0 0))

RFC 49: Curve geometries

- Driver support: GML, PostGIS, Interlis, GeoPackage, SQLite, MEM, VRT
- Methods to import/export ISO WKB/WKT for 2.5D
- Application code must be ready to deal with the new types
- Still no M-dimension support though or TIN / POLYHEDRALSURFACE

More at https://trac.osgeo.org/gdal/wiki/rfc49_curve_geometries

See also: https://github.com/mhugent/QGIS-Enhancement-Proposals/blob/master/QEP-8-geometry_redesign.rst

RFC 31: 64-bit feature IDs and fields

- Fix issues with big integer values being truncated / overflowed to 32bit values
- GetFID() / SetFID() now return a 64bit integer on all platforms
- GetFeature(), DeleteFeature(), SetNextByIndex() now take a 64bit integer argument.
- GetFeatureCount() now returns a 64bit integer.
- OFTInteger64 and OFTInteger64List field data types
- GetFieldAsInteger64(), GetFieldAsInteger64List(), SetField (xx, 64bit integer)
- Driver support: Shapefile, PostGIS, GeoJSON, CSV, SQLite, GeoPackage, MySQL, Oracle, MEM, VRT, GML, WFS, OSM
- Application code must be ready to deal with the new types and the interface changes

More at https://trac.osgeo.org/gdal/wiki/rfc31_ogr_64

RFC 50: OGR field subtypes

- A subtype is a hint that over-specifies a base field type
- Boolean, 16 bit integers or 32 bit floating point values
- Supported in GeoJSON, GML, CSV, PostgreSQL, GeoPackage, SQLite, FileGDB/OpenFileGDB, VRT
- OGR SQL dialect extended:
  - The field list of a SELECT can now accept boolean expressions, such as "SELECT x IS NULL, x >= 5 FROM foo"
  - The WHERE clause of a SELECT can now accept boolean fields, such as "SELECT * FROM foo WHERE a_boolean_field"
- Transparent for application code that doesn’t care about it

More at https://trac.osgeo.org/gdal/wiki/rfc50_ogr_field_subtype

RFC 52: Stricter OGR SQL quoting

- Conformance with SQL 92 standard
- String literals must be quoted with single(‘) quote characters
- Identifiers may be quoted, with double(“) quote characters
- Example :
  - OK: SELECT “a_column’, another_one FROM “a_table” WHERE “a_column” = ‘some_value’
  - ~~Wrong: SELECT ‘a_column’ FROM a_table WHERE ‘a_column’= “some_value”~~
- Application code must be audited. Some requests that used to work with GDAL 1.X might fail, or return different results

More at https://trac.osgeo.org/gdal/wiki/rfc52_strict_sql_quoting

RFC 53: Not-null constraints and default values

- Get/set methods for not-null constraints and default values of OGR fields
- Example:

```
Geometry Column 1 NOT NULL = WKT
 Geometry Column 2 NOT NULL = geom2
 id: Integer (0.0) NOT NULL DEFAULT 1234567
 dbl: Real (0.0) NOT NULL DEFAULT 1.456
 str: String (0.0) NOT NULL DEFAULT 'a'
 d: Date (0.0) NOT NULL DEFAULT CURRENT_DATE
 t: Time (0.0) NOT NULL DEFAULT CURRENT_TIME
 dt: DateTime (0.0) NOT NULL DEFAULT CURRENT_TIMESTAMP
 dt2: DateTime (0.0) NOT NULL DEFAULT '2013/12/11 01:23:45'
```

- Automatic propagation with ogr2ogr
- Driver support: GML, PostGIS, CartoDB, GeoPackage, SQLite, MEM, VRT, Oracle, MySQL, WFS, FileGDB/OpenFileGDB

More at https://trac.osgeo.org/gdal/wiki/rfc53_ogr_notnull_default

RFC 54: Dataset transactions

- RDBMS support BEGIN/COMMIT/ROLLBACK per connection
- GDAL 1.X: available at OGRLayer level ⇒ not the appropriate scope
- GDAL 2.0: available as GDALDataset methods
- Driver support: PostGIS, GeoPackage, SQLite
- Generic support for emulated transaction for drivers not naturally transactional. Implemented in FileGDB for now

More at https://trac.osgeo.org/gdal/wiki/rfc54_dataset_transactions

RFC 55: refined SetFeature and DeleteFeature semantics

- Reminder: SetFeature (OGR) ~= UPDATE (SQL) and DeleteFeature ~= DELETE (SQL)
- Previously when updating or deleting a row that didn’t exist, an error or a success could be reported depending on drivers
- Now GPKG, Shape, MySQL, OCI, SQLite, FileGDB, PG, CartoDB, MITAB and MSSQL will return OGRERR_NON_EXISTING_FEATURE

More at https://trac.osgeo.org/gdal/wiki/rfc55_refined_setfeature_deletefeature_semantics

RFC 56: OFTTime/OFTDateTime millisecond accuracy

- Fields of time and date/time can now accept up to millisecond accuracy (integral second before).
- Upgraded drivers: GeoJSON, CSV, PG, CartoDB, GPKG, MapInfo, LIBKML, ODS, XLSX, GeoRSS, GPX

More at https://trac.osgeo.org/gdal/wiki/rfc56_millisecond_precision

RFC 57: 64bit histogram bucket count

- Functions and methods that relate to histogram computation now count number of pixel cells on 64bit integer. Needed to avoid issues on big rasters of more than 2 billion pixels.

More at https://trac.osgeo.org/gdal/wiki/rfc57_histogram_64bit_count

GeoPackage support

- SQLite based container, OGC standard since 2014.
- Implements core spec in read/write/update + most standardized extensions
- Vector:
  - Spatial index (R-Tree)
  - Curve geometries
  - A-spatial tables (GDAL specific extension)
- Raster:
  - PNG, JPEG, WEBP, PNG+JPEG, PNG8 tiles
  - Tiling schemes: dataset specific or global tiling schemes (GoogleMercator, GlobalGeodetic)
  - Overviews/pyramids: power of 2 or arbitrary values
- Metadata (flattened)

More at http://gdal.org/drv_geopackage.html and http://gdal.org/drv_geopackage_raster.html

Other raster enhancements

- Warping:
  - can use overviews of source dataset
  - correctness fixes when subsampling with cubic, …
  - speed improvements for most resampling methods (apply to gdaladdo as well)
- Better scalability of raster block cache due to reduced lock contention. Ex. with i7-4700 HQ (8 vCPUs):
  - GDAL 1.11: time testblockcache: **19 s**
  - GDAL 2.0: time testblockcache: **7.6 s**
  - GDAL 2.0: with GDAL_RB_LOCK_TYPE=SPIN: **2.7s**
- Memory driver: most RasterIO() operations no longer involve the block cache at all

Other raster enhancements

- GeoTIFF driver:
  - Default: time ./testblockcache -ondisk: **7.5s**
  - GTIFF_DIRECT_IO=YES: short circuit the block cache&libtiff for most RasterIO() operations (restricted to uncompressed stripped GeoTIFF). time ./testblockcache -ondisk: **2s**
  - GTIFF_VIRTUAL_MEM_IO=YES: same as above, with tiled GeoTIFF as well. Uses memory-mapped file access. Linux only for now, 64bit recommended (could be extended to other platforms possible). time ./testblockcache -ondisk: **0.3s**
- New KEA driver :
  - Developed by Landcare Research/P.Bunting/S.Gillingham
  - Based on HDF5 container
  - Designed to support all GDAL concepts
  - Full read/write/update capabilities
- Improvements in JPEG 2000 OpenJPEG driver. Conformity with INSPIRE Orthoimagery technical guidelines. GMLJP2 v2 support

Other vector enhancements

- PostGIS:
  - Bulk insertions in COPY mode by default
  - Defered table creation
  - Curve geometries support
  - Dataset transactions
  - Interleaved layer reading
  - 64-bit integer identifiers and fields values
- MapInfo .TAB: update support
- CSV: autodetection of field types

GDAL 2.1 preview

- GDAL WMTS (OGC Web Map Tile Service) driver
- Support for very large rasters (WMS, WMTS) at high zoom levels (up to zoom level 22 for GoogleMaps tiling scheme with world coverage)
- CreateFeature() with arbitrary FID in FileGDB driver
- Linear interpolation in gdal_grid (fast and good looking results)
- OGR MongoDB read/write driver
- Improved OGR ElasticSearch driver
- On-the-fly pansharpening through VRT
- Integration of existing command line utilities (gdal_translate, gdalwarp, ogr2ogr, etc…) as C callable functions (and available through Python too) (GSoC 2015 project)

Potential project directions

- CMake build system
  - Aim: out-of-tree builds, correct header dependency.
  - When completed and stable, unified build system for Unix&Windows
  - Ongoing effort lead by Dmitriy Baryshnikov: https://github.com/nextgis/gdal_svn/tree/rsmd-reader-dev

- Per-dataset block cache:
  - To reach zero lock contention
  - https://trac.osgeo.org/gdal/wiki/rfc47_dataset_caching (Blake Thompson)

Potential project directions

- Integration of Geography Network Model:
    - Creating network topology over spatial data manually or automatically
    - Adding/removing spatial layers/features to the network
    - Defining business logic of the networks (way of assigning graph costs)
    - Several methods of network analysis.
    - Generic support for OGR formats with write capabilities
    - Potentially link with dedicated interfaces (PgRouting, Spatialite network, GML...)
    - Developed by Mikhail Gusev for GSoC 2014 https://trac.osgeo.org/gdal/wiki/rfc48_geographical_networks_support

Potential project directions

- Planar topology:
  - New abstraction based on related ISO SQL/MM Part3 modeling
  - Topology primitives: nodes, edges, faces
  - TopoGeometry build on primitives / hierarchical TopoGeometry
  - Building of topology from geometries
  - Geometry ←→ TopoGeometry conversions
  - Interface with PostGIS, GRASS, Oracle, GML, Spatialite, TopoJSON
  - Conversion : topo2topo

Potential project directions

- OpenFileGDB write support
- SRS handling improvements: guessing EPSG codes, proposing appropriate datum shifts according to location, …
- Alternative geometry library: Boost::Geometry
- Complex features???
- More drivers, performance improvements, ...

Questions?

See Also:

- http://www.gdal.org/
- https://trac.osgeo.org/gdal/wiki/RfcList

Contact: even.rouault@spatialys.com