Services for Web-Publishing and Extraction of Fragments of Big Raster Data-sets

Alexei E. Hmelnov[1]

[1]*Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences (IDSTU SB RAS), 134 Lermontov St, Irkutsk, 664033, Russian Federation*

Abstract

To be able to conveniently work with large raster data-sets we have developed the `MRG` file format. In the article we consider the software tools developed to create and process the `MRG` files and the global DEMs and the other global raster data that we have collected and converted to `MRG`. To make the data available for non local users we have developed the WMS and WPS services for publishing `MRG` data. In the article we discuss some details of their implementation and usage.

Keywords

data format, big raster data, loss-less compression, global DEM, WMS service, WPS service

1. Introduction

Since the time, when in the beginning of 2000$^{\text{th}}$ the 1$^{\text{st}}$ version of Shuttle Radar Topography Mission (SRTM) digital elevation model (DEM) had been published, the progress of remote sensing technologies has led to public availability of several global digital elevation models (GDEM) and now we can choose which of them to use for a particular project. The quality of the models may strongly vary, for example, while the ASTER GDEM is usually reported to be more precise than SRTM, our experience shows that in the area of Irkutsk it has prohibitively low quality, so that we don't even consider this GDEM in our current research.

The best resolution of the current publicly available GDEMs is $1''$ (arc-second) $\approx 30m$ per pixel. All the GDEMs are published as a collection of tiles with the sizes from $1^\circ \times 1^\circ$ to $5^\circ \times 5^\circ$ in GeoTIFF or JPEG 2000 file formats. So all the tiles of typical GDEM take several dozens of gigabytes. To obtain the tiles it is usually required to register on the corresponding site, and it will take rather long time to find and download the data for an area of interest, and the user sometimes may be banned for trying to download the data too fast. Because the end-user is usually interested in exploring a particular area of GDEM, after downloading the data it will usually be required to combine the tiles into a single raster file and to cut the area of interest from it. As a result it takes quite a lot of time to start the actual usage of the data to solve the problem under consideration.

6^{th} International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2024), July 01–05, 2024, Irkutsk, Russia

hmelnov@icc.ru (A. E. Hmelnov)

0000-0002-0125-1130 (A. E. Hmelnov)

ICCS-DE 2024 Workshop Proceedings (iccs-de.icc.ru)
DOI: 10.47350/ICCS-DE.2024.07

After we found that the SRTM data become publicly available we have developed the file format MRG (MultiResolution Grid), which is intended for compact storage of big raster images with the capability to quickly extract any fragments of the image with any resolution [1]. When represented in MRG, DEM data usually take less space, than the original tiles, even compressed, but, in contrast to the original formats, in MRG we have the random access to any DEM fragments.

In contrast to the MrSID, ECW or JPEG 2000 file formats, which also have some kind or resolution pyramids, but use very complex and time consuming transforms, the MRG file format is based upon construction of a resolution pyramid and upon loss-less compression of the differences between the interpolated and the actual data for the next resolution level using a family of very simple compression algorithms. The complexity of these algorithms is always asymmetric: while the compression phase searches for the optimal split of the sequence of delta values into the intervals of constant bit depths using the dynamic programming method, the decompression phase is extremely fast, because it simply reads and decodes the intervals [2].

Representation of a DEM in MRG makes it seamless and gives the user the capability to forget about the tiles and to quickly pan and zoom into any DEM part and to extract any its fragment with the required level of details. So, the local user can quickly find the data of interest and start working with it.

Recently with the support of the big project "Fundamentals, methods and technologies for digital monitoring and forecasting of the environmental situation on the Baikal natural territory" we have acquired a data storage system of the size of several hundred terabytes. Using the data storage it becomes possible to collect several GDEMs and to convert them into the corresponding MRG files. We had to advance somehow the MRG software to be able to work with the data as big as never before.

To provide convenient access to the big MRG data-sets for the remote users we have developed a couple of services: the WMS service for viewing the data, and the WPS service for extraction of their fragments.

In the article we'll consider some details of the implementation of the services and their usage and some statistics on the examples of the GDEM and optical remote sensing data represented in MRG.

2. The MRG software

To work with the MRG files we have developed a number of software modules that allow us to create the files from the original GeoTIFF, JPEG 2000 and the other file formats, view the resulting raster images, and retrieve from the MRG files various kinds of information, like image fragments with desired resolution, contour lines or cross-sections:

MRGView is an interactive (GUI) program for processing DEMs;

MRGImgV is an interactive (GUI) program for processing multichannel images;

MRGLib is a dynamic library, that makes it possible to develop external programs, which work with MRG;

MRGCmd is a command line utility, that can perform in batch mode almost all the operations, that we can do in `MRGView` and `MRGImgV`;

MRGWebSrv is a standalone application that works as a Web-server and implements the WMS-service and WPS-service protocols for publishing the registered MRG files;

MRGWebSrvSvc is a `Windows` service that works the same way as `MRGWebSrv`.

All the `MRG` software use the common code base. We use `MRGWebSrv` mostly for debugging purposes, because it is much more easier to debug a standalone application, than the `Windows` service. And the main advantage of `Windows` service is that after its registration it works in an unattended mode and will be automatically restarted, when the computer restarts.

The `MRGCmd` application plays crucial role for construction of big GDEM files, because, in contrast to the GUI applications, it allows us e.g. to rerun the same task several times with exactly the same parameters, when something goes wrong, or to change some parameters and process exactly the same data again to compare the results of the two runs. The `MRGCmd` utility can perform a sequence of the following commands

- **Set** - Set MRG creation options
- **Open** - Open MRG file
- **NewImage** - Create MRG file from images
- **NewDEM** - Create MRG file from DEM files
- **Describe** - Describe MRG
- **GetIsolines** - Get isolines from MRG image
- **GetProfiles** - Get profiles from MRG image
- **Export** - Export fragment to GeoTIFF
- **Defragment** - Create defragmented version of MRG
- **Compare** - Compare current MRG to another one
- **Exec** - Execute commands from file

which can be passed in the application command line separated by semicolons or in a separate text file using the `Exec` command.

Let's consider a couple of examples. The following command compares two files: the old and already checked file `testWok.mrg` and the new file `testW.mrg`. The comparison will stop after finding 4 errors (mismatches).

```
MRGCmd.exe Open testWok.mrg ; Compare -e4 testW.mrg
```

And the following `bat`-file generates `tif` previews for a series of `MRG` files, that were created for the Copernicus DEM bands of different horizontal resolution.

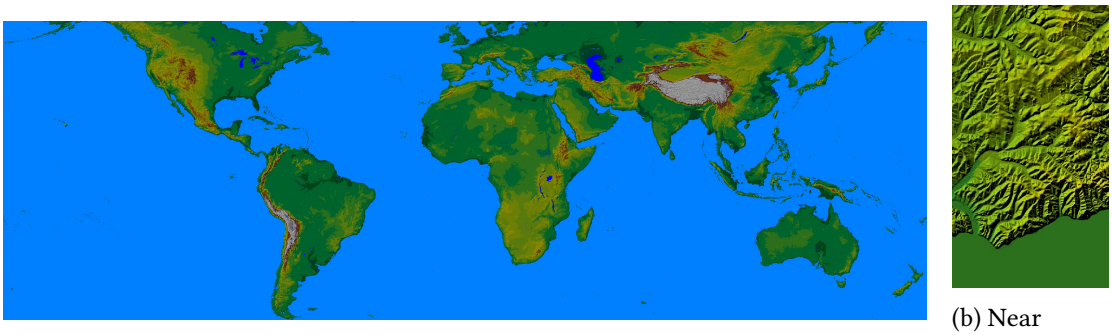
```
for %%f in (all_*.mrg) do (
  echo %%f
  MRGCmd Set -W ; Open %%f ; Export -f -l5 TIFF\%%~nf.tif
)
```

The `-l5` flag tells the `Export` command to use the 5^{th} level of details, i.e. to use the resolution which is $2^5 = 32$ times lower than the original very large image.

3. The Global DEMs and the other raster images represented by MRG files

As it was already mentioned above, the very development of the MRG file format was inspired by the open publishing of STRM data. The 1st version of SRTM data-set had resolution of $3'' \approx 90m$ per pixel, and it takes approximately from 8 to 10 GB to represent the data in MRG file format (the difference in the file sizes is explained by the choice of the interval header packing algorithm: the default one gives about 10 GB, and the algorithm based upon the interval statistics for these data gives better results, but to obtain the statistics it is anyway required first to compress the data with the default parameters).

Initially we were not trying to create the GDEMs, and were completely satisfied by the fact that in the MRG file it was possible to store the whole DEM for the territory of Russia on a single CD and to quickly view the image immediately from this CD. When creating the MRG file for the whole SRTM GDEM it was first required to upgrade the MRG file format, because the 32-bit file offsets of the data blocks, which were initially used in the files, had become no longer enough for the 10 GB files. To solve the problem we have added to the file header a field about the data block offset alignment: now the default alignment of 16 bytes enabled us to create the files with the size up to 64GB. But to represent the modern global DEMs and the other raster images this limitation may become too restrictive. Increasing the block alignment further would mean wasting much more space, so we decided to switch to the 64-bit offsets for the very large files of this kind. The modern SRTM GDEM of the $1''$ per pixel resolution [3] takes up 56.3 GB in MRG (Fig. 1), while the source data were published in 14 297 HGT files (pure 2D arrays of 16-bit integer values) compressed by ZIP of the total size 97.6 GB.



(a) The whole GDEM

(b) Near Listvyanka

Figure 1: The hill shading of SRTM GDEM from MRG

According to some modern research papers [4], which compare different GDEMs, the best of them, which is far ahead of the others, is the Copernicus DEM. In fact the Copernicus DEM was produced from WorldDEM – a commercial GDEM with the initial resolution of 10 m per pixel, that was then improved in its next version WorldDEM Neo to 5 m per pixel. So, while, say, the SRTM DEM is a result of the attempts to up-sample the initial DEM of lower resolution using every trick in the book, the Copernicus DEM is a result of down-sampling of the much more precise WorldDEM data.

The Copernicus DEM data for 2022 as it was published [5],[6] take 1.74 TB and contain 26 450 TAR files. Each of the uncompressed TAR archives contains single DEM tile in an uncompressed GeoTIFF file and the other uncompressed auxiliary, preview, and license files. Among the auxiliary files the largest is the Height Error Mask GeoTIFF, which always has exactly the same file size as the DEM GeoTIFF, because it is also uncompressed and has the same pixel size and format. As a result of adding all the auxiliary files the DEM data take up about 47% of the TAR file size. We believe that it would be much easier to download and distribute the data if the tiles were compressed somehow.

The pixel values of the Copernicus DEM GeoTIFFs are the 32-bit floats. Unfortunately, the compression algorithm in the MRG file format is designed for integer differences only, and by now it can contain only the 16-bit integer values, so we have to round the float values to 0.25 m steps to be able to fit the height of Qomolangma into the unsigned 16-bit range.

Another peculiarity of the Copernicus DEM data is that the horizontal resolution of the tiles varies with latitude. All the tiles cover the area $1^\circ \times 1^\circ$, but their horizontal (by longitude) resolution varies from the same as the vertical (by latitude) resolution of 3600 pixels per degree in the stripe of latitudes from -50° to 49° to 360 pixels per degree for the Antarctic tiles with the latitudes below or equal to -86° (see Table 1). The latitudes in the names of the tiles are the latitudes of their lower (south) edges, so the north and south stripes are in fact symmetric (mirror each other). As a result we have to split the data into 10 stripes of the consecutive tiles having the same horizontal resolution. The total size of the Copernicus DEM data in MRG is 96.5 GB.

Table 1: Horizontal resolution of the Copernicus DEM tiles

Latitude	**Pixels/°**	**MRG size**
$-90^\circ .. -86^\circ$	360	0, 98 GB
$-85^\circ .. -81^\circ$	720	2.05 GB
$-80^\circ .. -71^\circ$	1200	5.25 GB
$-70^\circ .. -61^\circ$	1800	1.20 GB
$-60^\circ .. -51^\circ$	2400	151 MB
$-50^\circ..49^\circ$	3600	66.7 GB
$50^\circ..59^\circ$	2400	9.64 GB
$60^\circ..69^\circ$	1800	8.26 GB
$70^\circ..79^\circ$	1200	2.02 GB
$80^\circ..84^\circ$	720	267 MB

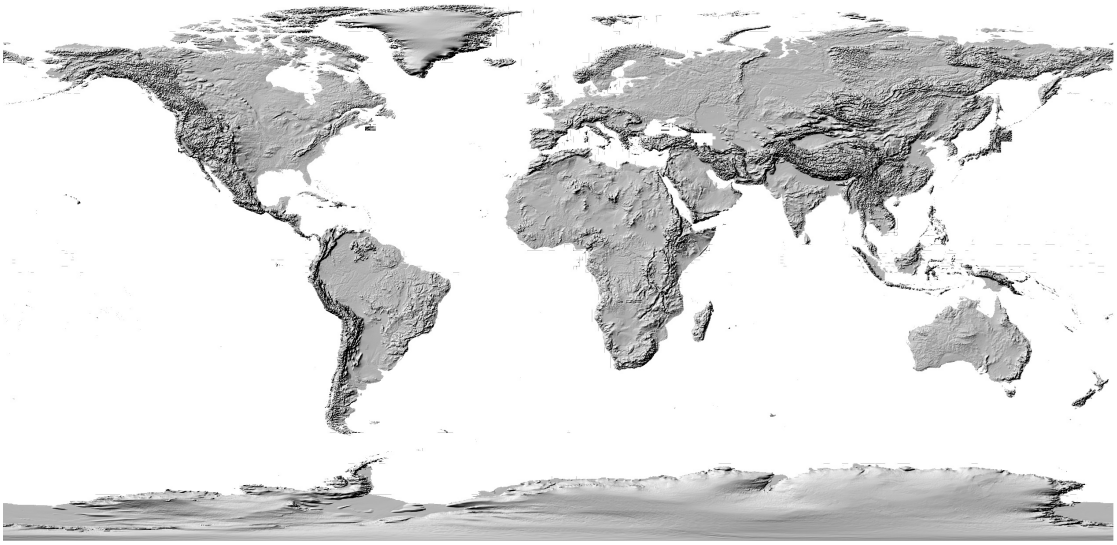
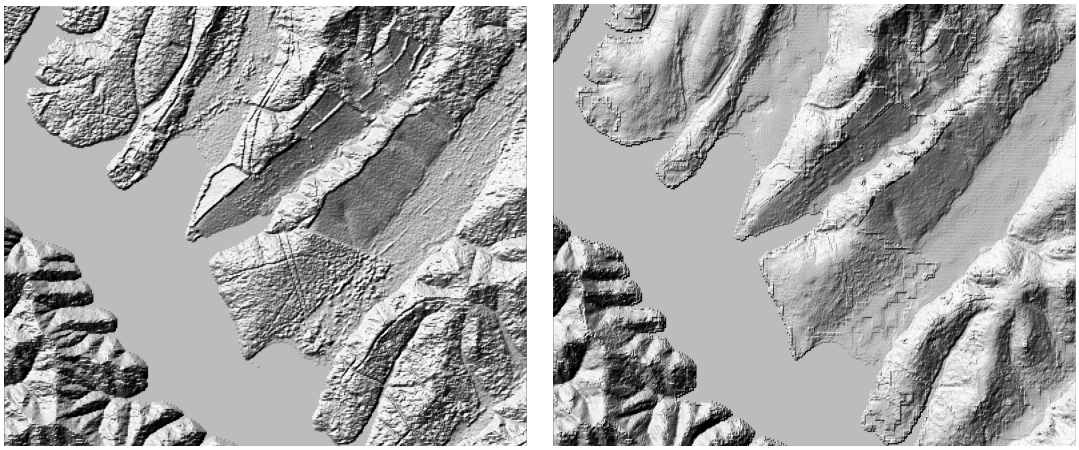


Figure 2: The shadow relief of Copernicus DEM in 10 MRG files shown together

Another interesting GDEM, that was produced from the Copernicus DEM using machine learning is FABDEM (Forest And Buildings removed Copernicus DEM) [7],[8]. This data-set is a result of an attempt to make some approximation of global DTM from Copernicus DEM by subtracting the heights of trees in the area of forests and the heights of buildings in the areas of cities, which are estimated using the random forest algorithm trained on a series of sample areas, for which the heights were measured by the means of more precise instruments. By visual inspection of the FABDEM we have found some artifacts in the data, like the numerous stepped boundaries which are clearly visible on the Figure 3(b) (which indicate us that the FABDEM generation algorithm usually 'deforest' the Copernicus DEM using some rectangular areas), but the overall quality of the data-set is reported to be very high.

The whole FABDEM data-set is distributed in 356 zip archives, each of them covers an area of $10^\circ \times 10^\circ$ and may contain up to 100 files for the $1^\circ \times 1^\circ$ tiles in the area. The pixel values of the FABDEM GeoTIFFs are also the 32-bit floats. But in contrast to the Copernicus DEM data the horizontal resolution of all the FABDEM tiles was risen to 3600 pixels per degree. The total size of the FABDEM archives is 296 GB, and the size of the resulting MRG file with the default header compression parameters is 85.3 GB, but here we also have rounded its values to the same 0.25 m steps.

Another useful global data-set is the high-resolution canopy height model of the Earth (the Global Canopy) [9], [10]. The vegetation height model has been obtained from the Sentinel images using an ensemble of convolutional neural networks (CNNs) and has the corresponding global resolution of 10 m/pixel, which is 3 times higher than all the models considered hereinbefore. The Global Canopy data are published in 2650 GeoTIFF files with 8-bit integer pixel values, representing the estimated tree heights in meters. Almost all the heights are in the range from 0 to 45 m and in the areas without vegetation like water objects or deserts the heights has the 'no data' values. The GeoTIFF files use the LZW compression and their total size is 349 GB. The dimensions of the resulting MRG file (Figure 4) are 4320000×1728000 pixels, and its size is 331 GB when using the default interval header compression algorithm, which in fact



(a) Copernicus DEM (b) FABDEM

Figure 3: Copernicus DEM vs FABDEM (the hill shading of the same area)

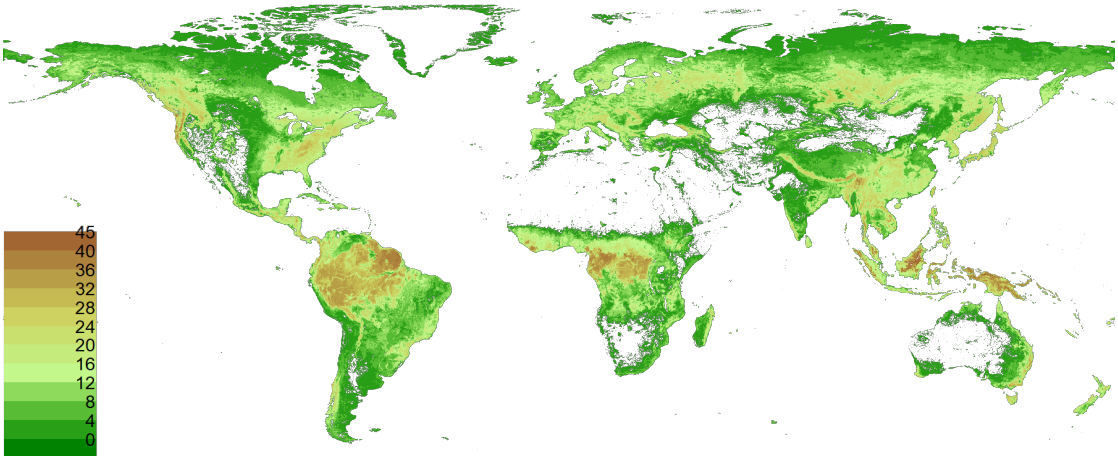


Figure 4: The Global Canopy data-set. Its pixel values are the estimated tree heights in meters

is more suitable for the 16-bit values. After using a more specific to the data interval header packing, which is based upon the statistics of the data, the resulting `MRG` file size further goes down to 323 GB.

The last but not least in our collection of the global data-sets is a by-product of the project known as the Hansen forest [11][12]. The main goal of the project is to observe the global tree cover changes using the Landsat images. As a result of the project its authors have produced such data-sets as 'Forest loss year' or 'Tree canopy cover for year 2000' but to obtain the data they have also produced the global composite images for a series of years including the last of them 'Circa year 2022 Landsat cloud-free image composite'. This composite image has its own independent value which is of interest for us: it can be used as a base layer for various maps. Unfortunately for us the composite image doesn't include all the true-color channels, it contains

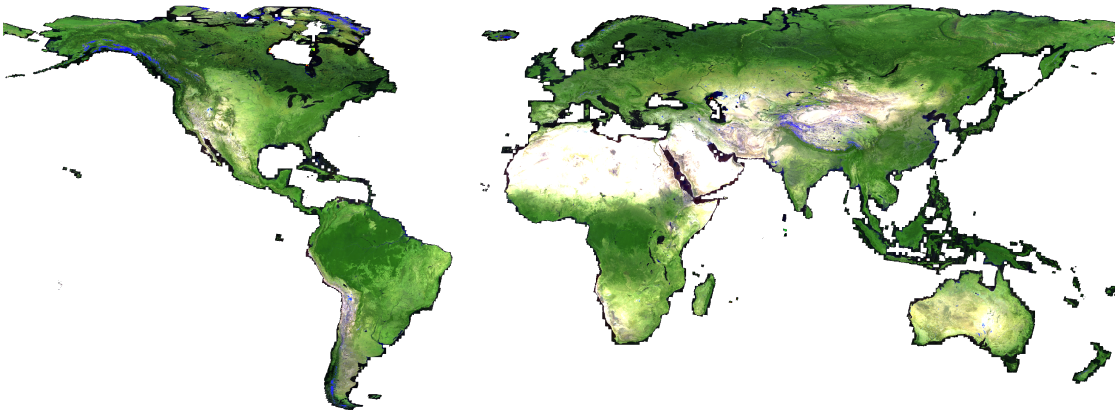


Figure 5: The Hansen forest circa year 2022 Landsat cloud-free image composite in pseudo colors: red=SWIR2, green=SWIR1, blue=Red

only the following 4 Landsat bands, which are needed for the forest analysis: Red ($0.66\,\mu m$), NIR ($0.86\,\mu m$), SWIR1 ($1.6\,\mu m$), SWIR2 ($2.2\,\mu m$), but some combinations of the bands may look rather natural (Figure 5).

In contrast to the other `MRG` files the Hansen forest composite image has more than one (four) channels, and it uses the color decorrelation algorithm to improve the compression ratio [13]. The global Hansen forest composite image data are published as 504 GeoTIFF tiles with LZW compression, which have four 8-bit channels (Red, NIR, SWIR1, SWIR2). The total size of the files is 755 GB, and the size of the corresponding `MRG` file is 446 GB.

4. Web service for publishing `MRG` files

It would be impossible to immediately use the raster images of the sizes as it was described in the previous chapter in any modern GIS. So we have developed the program `MRGWebSrvSvc`, which implements two OGC standards for accessing the registered big `MRG` files: WMS and WPS. Initially we have supported the WMS (Web Map Service) standard, that allows users to view the images as a map layers and combine the layers with any other raster or vector data-sets. To be published by the service the `MRG` files should be registered in its configuration file. Besides from the `MRG` file path the configuration should specify its display name and display parameters.

To be able to work with the fragments of big `MRG` files in GIS programs we have additionally implemented the WPS (Web Processing Service) processes for requesting the data. To obtain the the list of processes of a WPS service the `GetCapabilities` request can be used. And the `DescribeProcess` request provides the information about the inputs and outputs of the process. To run the process the `Execute` request is used. The WPS standard imposes certain restrictions on the ways of interaction with the service, which do not always make this interaction as convenient as it may be without these restrictions: the service should implement some *processes*, each process has a number of *input* and *output* parameters; the parameters can be passed by value (the *literals*) or in the form of a fragment of the XML code of request

or response or by URL (for *complex data*); if a process has single output parameter, it can immediately return its value in the response to the `Execute` request.

There exist software libraries for implementation of WPS services, for example, the 52°North WPS library for Java or PyWPS or OWSLib for Python, but in fact for a small service it may be easier to implement the processing of the WPS requests from scratch.

In the current version of the WPS service for obtaining raster fragments we have implemented two processes: `Export`, which returns the fragments in GeoTIFF file format, and `QueryPoints`, which retrieves the DEM values in the list of points. The `Export` process has the following input parameters: `name` – the name of a registered raster; `bboxin` — bounding rectangle of the exported area; `level` — the level of details (0 denotes the most detailed level, corresponding to the resolution of the `MRG` file, then increasing the level by 1 reduces the resolution by half). It returns GeoTIFF file with the contents of the specified fragment of the selected raster at the specified level of details.

The value of the `name` parameter should belong to the list of names of the registered `MRG` files. The WPS standard allows us to specify the list of valid values of the input literal `name` in the `DescribeProcess` response. The WPS client, which can use this information, may create a drop-down list for the name selection field. This method turns out to be not very convenient when there are a lot of registered `MRG` files: while the WMS protocol allows us to organize the map layers into a tree, here we are limited to a linear list only. And the more serious limitation of the WPS protocol is that we can't specify the valid range of values of the `bboxin` and `level` parameters in the process description, since these constraints depend on the selected layer, and this kind of inter-dependencies is not supported by the WMS standard. It would be possible to implement additional processes for obtaining this kind of meta-information about the registered `MRG` files, but the WPS standard still will not support the automated checking of these constraints.

To get the image of a `MRG` file with the minimal level of details and simultaneously find out which area it covers, one can pass the `level`$= -1$ (here we use the negative indexes in the style of the Python arrays) and the default `bboxin`, which corresponds to the maximum bounding box ($-180°$ -90°; $180°$ 90°).

The way the resulting image fragment will be returned by the `Export` process depends on its size, the size limits can be set in the service configuration file. The following threshold values are currently selected: the images with the size of up to 4Mpx are directly returned; if the number of pixels exceeds this threshold, the file is first saved in a special folder, which can be accessed by the geoportal Web server, and the Execute request returns the XML document with the URL corresponding to this file on the Web-server. Since the MRG format's capability to store large-scale raster data is significantly superior to that of GeoTIFF, it makes no sense to try to get all the `MRG` information at level 0 in one request: even if the file were created, it would be impossible to work with it. Therefore we have another threshold: the maximum image fragment size. The current value of this threshold is 1024Mpx.

For testing WPS services in QGIS, we were able to find only one plug-in – the WPS Client which belongs to the experimental category. However, the capabilities of this plug-in turned out to be sufficient for debugging our service. In particular, it creates a drop-down list of valid values for the `name` parameter, and also it automatically substitutes the bounding box of the current map view for the value of the `bboxin` parameter (Figure 6(a)). And to visually select

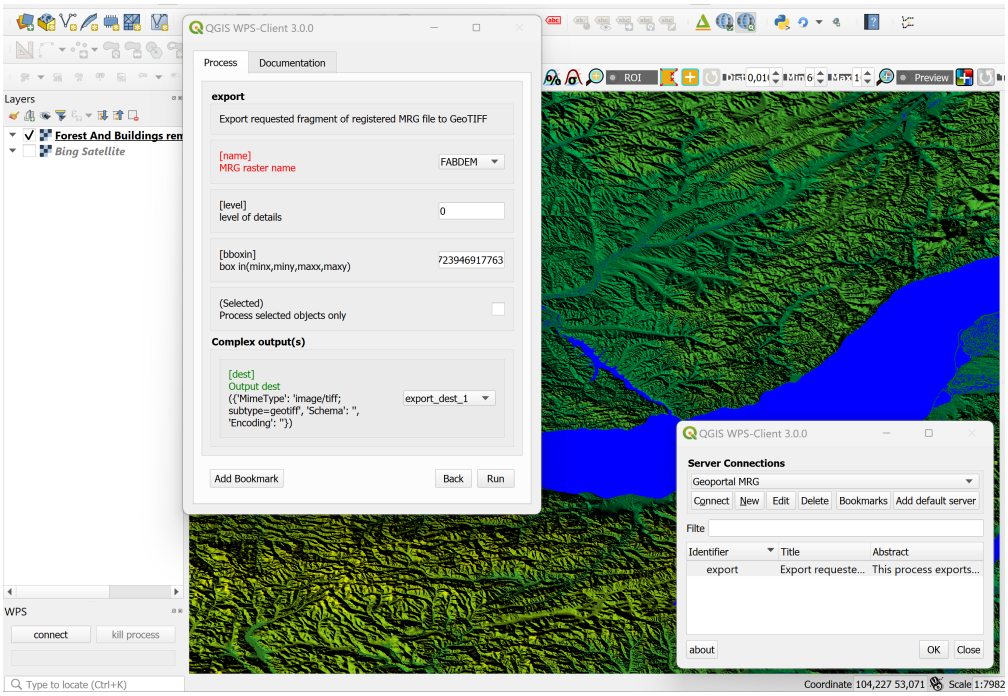
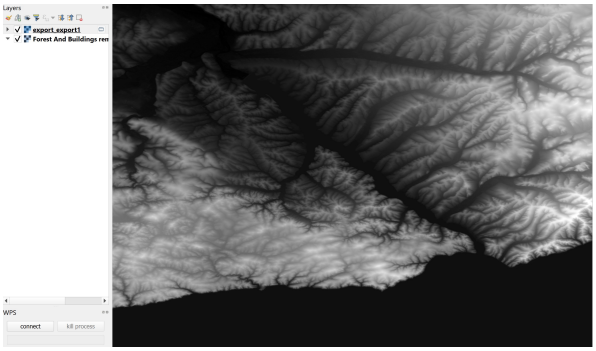


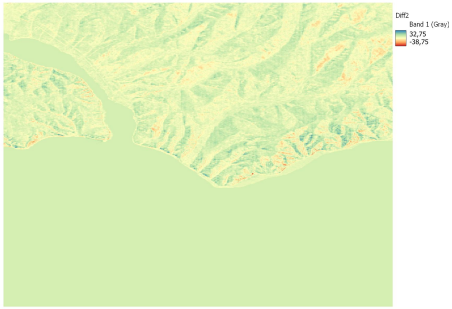
Figure 6: Request of MRG file fragment using the WPS Client plug-in dialog in QGIS

the area of interest we can use the images obtained from the WMS service of our MRG server.

After executing the request, a temporary raster layer is created in QGIS, which can be further used to solve various problems of processing and analyzing the received data. For example, different terrain models can be compared to each other (Figure 6(b)). And the use in QGIS is not the main purpose of the developed WPS service: the geo-portal software allows its users to create compositions of services, which can now include the data fetched from the MRG format.



(a) New layer added as a result of the WPS request



(b) Comparison of Copernicus DEM and SRTM in QGIS

Figure 7: Processing the requested MRG file fragments in QGIS

5. Conclusion

The MRG file format and the software developed for its processing can substantially streamline access to the global raster data-sets, that are available now. The data-sets in MRG format take less space than in their original representations, and the main advantage of this representation is that any data fragment can be quickly accessed with any level of details.

The services developed for publishing MRG data follow the OGC WMS and WPS standards and allow non local users to easily access the data. The resulting data layers make their contribution into our geo-portal data collection. Moreover, the WPS service results can be used for development of more complex data processing algorithms. For example, the information about point elevations from the GDEMs may be used for training neural networks.

References

[1] A. E. Hmelnov, The MRG file format for compact representation and fast decompression of large digital elevation models (in russian), Computational technologies 20 (2015) 63–74.

[2] A. E. Hmelnov, A lossless compression algorithm for integer differences sequences by optimization of their division into intervals of constant bit depth values (in russian), Computational technologies 20 (2015) 75–98.

[3] Shuttle radar topography mission (SRTM) global, 2013. URL: https://spacedata.copernicus.eu/collections/copernicus-digital-elevation-model. doi:10.5069/G9445JDF.

[4] C. Bielski, C. Lopez-Vazquez, C. H. Grohmann, P. L. Guth, L. Hawker, D. Gesch, S. Trevisani, V. Herrera-Cruz, S. Riazanoff, A. Corseaux, H. I. Reuter, P. Strobl, Novel approach for ranking DEMs: Copernicus DEM improves one arc second open global topography, IEEE Transactions on Geoscience and Remote Sensing 62 (2024) 1–22. doi:10.1109/TGRS.2024.3368015.

[5] Copernicus DEM - global and european digital elevation model (COP-DEM), 2022. URL: https://spacedata.copernicus.eu/collections/copernicus-digital-elevation-model. doi:10.5270/ESA-c5d3d65.

[6] M. Buchhorn, M. Lesiv, N.-E. Tsendbazar, M. Herold, L. Bertels, B. Smets, Copernicus global land cover layers–collection 2, Remote Sensing 12 (2020). URL: https://www.mdpi.com/2072-4292/12/6/1044. doi:10.3390/rs12061044.

[7] L. Hawker, P. Uhe, L. Paulo, J. Sosa, J. Savage, C. Sampson, J. Neal, A 30 m global map of elevation with forests and buildings removed, Environmental Research Letters 17 (2022) 024016. doi:10.1088/1748-9326/ac4d4f.

[8] J. Neal, L. Hawker, FABDEM V1-2, 2023. URL: https://data.bris.ac.uk/data/dataset/s5hqmjcdj8yo2ibzi9b4ew3sn. doi:10.5523/bris.s5hqmjcdj8yo2ibzi9b4ew3sn.

[9] N. Lang, W. Jetz, K. Schindler, J. D. Wegner, A high-resolution canopy height model of the Earth, Nature Ecology & Evolution 7 (2023) 1778–1789. URL: https://doi.org/10.1038/s41559-023-02206-6. doi:10.1038/s41559-023-02206-6.

[10] N. Lang, W. Jetz, K. Schindler, J. D. Wegner, A high-resolution canopy height model of the Earth, 2023. URL: https://langnico.github.io/globalcanopyheight/.

[11] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, J. R. G. Townshend, High-resolution global maps of 21st-century forest cover change, Science 342 (2013) 850–853. URL: https://www.science.org/doi/abs/10.1126/science.1244693. doi:10.1126/science.1244693. arXiv:https://www.science.org/doi/pdf/10.1126/science.1244693.

[12] M. C. Hansen, P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, J. R. G. Townshend, Global forest change, 2023. URL: https://glad.earthengine.app/view/global-forest-change.

[13] A. E. Hmelnov, Reversible integer approximation of color space transforms for lossless compression of big color raster data, Computer Optics 46 (2022) 492–505. doi:10.18287/2412-6179-CO-1052.