Making METOC data portable with video codecs

Daan Gommers
Datalab
COMMIT/JIVC/KIXS
Utrecht, The Netherlands
dj.gommers@mindef.nl

Dennis Strik
Datalab
COMMIT/JIVC/KIXS
Utrecht, The Netherlands
dm.strik@mindef.nl

Vincent van Leijen
Datalab
COMMIT/JIVC/KIXS
Utrecht, The Netherlands
av.v.leijen@mindef.nl

***Abstract*— Numerical Weather Predictions and Ocean forecasts (METOC) are typically high quality datasets with a large file size. Some users have limited bandwidth available and require much smaller file size that can be shared by email or satellite communications. Compression of the datasets can be a solution for these use cases. In previous research, we experimented with lossy compression to find the right balance between compression factor, information loss and speed of operations. In this work, we expand on this by evaluating the use of common video codecs to make METOC data portable. Such codes are maintained for a very large user community and offer decompression in near-real time of small compressed datasets while preserving most of the important information.**

Keywords— meteorology, oceanography, compression

I. INTRODUCTION

Numerical weather predictions (NWP) and ocean forecasts are highly valuable products for any planning activity [1, 2]. Datasets of such environmental forecasts are still improving in availability, reliability, accuracy and spatial and temporal resolution. Military planners use these products in tactical decision aids [3] that evaluate the impact of the expected state of the environment on the performance of sensor systems such as radar [4, 5] and sonar [6]. The motivation for this work is to make modern forecast products available to planners in the field.

Daily updates of large forecasting products are demanding in terms of storage hardware, data transmission time and communication bandwidth. To save money and time it is common practice to access and share these datasets using public cloud technology. Military users value the option to be independent of third-party storage and like to transfer these datasets to their own networks on a daily basis. It is common practice to use compression techniques when handling larger METOC files. Environmental datasets are usually distributed as a netCDF/HDF-5 or GRIB/GRIB 2 [7, 8], and these formats offer some lossless compression capabilities such as JPEG2000. This regular support of (lossy) compression offers a file size reduction in the order of factor 10. This work aims to support a category of users in the field that have very limited bandwidth available and would benefit from a more ambitious compression factor in the order of 25 or even 100. To achieve this goal, we have experimented with the use of lossy compression algorithms to make NWP portable and to share highly compressed datasets by email and satellite communications.

Initial prototypes of custom lossy compression algorithms demonstrated the feasibility of larger reduction factors [9, 10]. However, these methods came at a price. Either the information loss made the data unusable, or the time to (de)compress was unreasonable. In our search for the right balance between compression factor, information loss and speed of operations, we started the current experiment on third party compression algorithms in the form of video codecs. These highly optimized algorithms are fine-tuned for fast encoding, and decoding and small file sizes while keeping differences minimal. Video codecs have a long development history, a very large user community, are available on various operating systems, and are being maintained to keep up with technological advances such as multithreading or GPU's.

This work aims to identify suitable video codecs to compress large environmental datasets. We describe a prototype software application to compress and decompress datasets and assess the overall quality and performance of portable METOC datasets.

II. METHOD

A. Benchmark of METOC datasets

To study the performance of video codecs on the compression of gridded datasets we created a benchmark with outputs of meteorological and oceanographic models. Such models describe the physical state of the atmosphere or ocean in the past, present or future. The typical dimensions are time, latitude, longitude, and either height or depth. Some parameters are 2D (lat-lon) like altimetry, bathymetry, land-sea mask or geopotential. Other parameters are 3D such as sea surface temperature (lat-lon-time) or geopotential height (lat-lon-height). The most voluminous parameters are 4D such as pressure, temperature, humidity, salinity, wind, current, etc.

The datasets in this study were sourced from the European Centre for Medium-Range Weather Forecasts (ECMWF) and the Copernicus Climate Change Service (COPERNICUS) [11, 12]. By nature, environmental model parameters have smooth transitions. Solid bounds and steep gradients are mostly absent, with an exception for the ground, ocean bottom and coast lines. Therefore, the benchmark covered an area with both land and sea.

METOC datasets are mostly stored and distributed as files in GRIB or netCDF format [13]. GRIB is a widely used format, however it is not self-describing. A GRIB Table is needed for reading these files, and different suppliers may have their own conventions and practices when it comes to these GRIB Tables. This can be a nuisance as not every GRIB (or GRIB2) file can be read by standard GRIB readers. So for our convenience we convert any GRIB data to netCDF 4, prior to compression. Since netCDF 4 is based on the HDF5 format, it allows for easy subsetting, slicing and extraction of specific regions or variables.

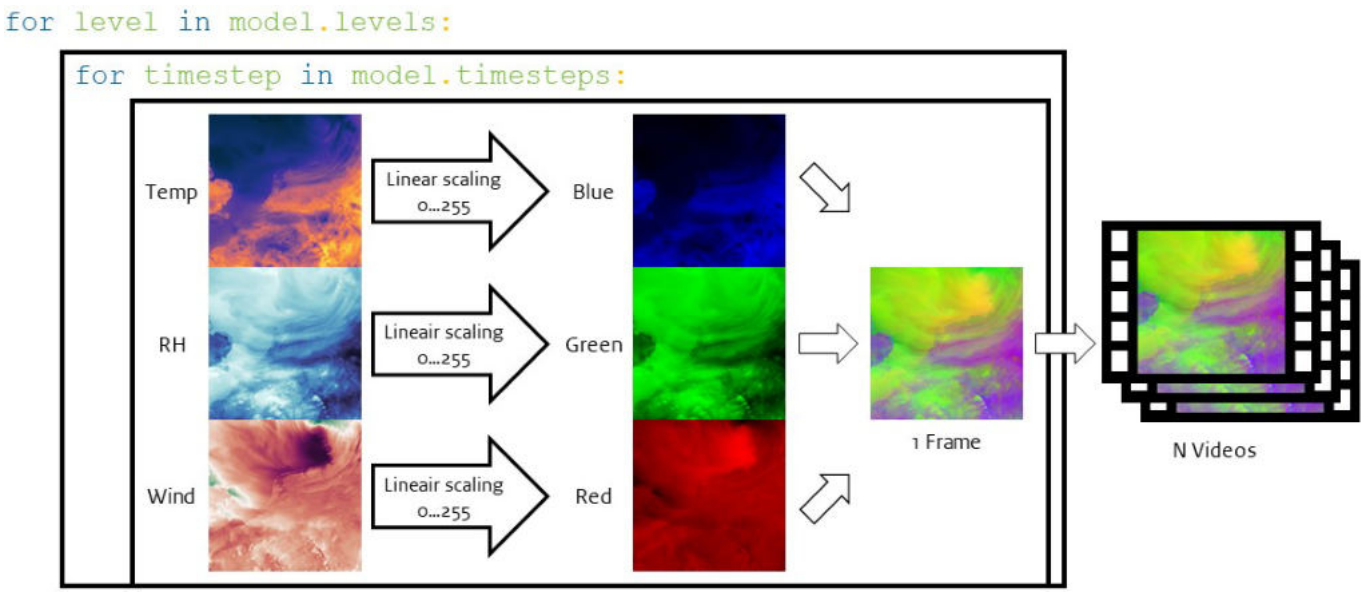

Fig 1. Schematic representation of the algorithm that stores data slices as video frames

B. *Selected video codecs*

A video codec is hardware or software that combines coding and decoding of digital video to make file sizes smaller. In this study we compared the video codecs FLV, libx264 (H.264), MJPEG, MPEG2Video, MPEG4, and Snow. These codecs are available in FFmpeg and are generally recognized as the state of the art algorithms for streaming media [14].

C. *Prototype EVA to study compression*

To convert METOC data into video files, a software prototype was created in Python. The input is a gridded dataset (netCDF-file) together with the requested codec and compression factor. The program reads the data (using the xarray-package [15]), and for each parameter the data is scaled linearly to integer values ranging from 0 to 255. Three parameters are combined into a Red, Green and Blue (RGB) video frame, or one parameter in a black-and-white frame, such that every cell in the data corresponds to one pixel in a video frame In this way longitude and latitude correspond to width and height of the frame, where time steps in the model data correspond to time steps in a video. For 4D data, each height or depth layer is stored in a separate video file, see Fig. 1. These (raw) video frames are then encoded by the video codec, using FFmpeg, and stored in a mp4 file. We choose the mp4 container for its compatibility with a wide range of codecs. The amount of data that is encoded for a unit of time is called the bitrate and can be varied during encoding. We derive the bitrate from the requested compression factor. The output consist of several video files and a metadata file, for bookkeeping, such as the original dimensions, units, coordinate systems, and other attributes. The set of output files are archived in a single zipped file. The prototype and compressed archive file are referred to as EVA (Environmental Video Array) and the compressed file has the extension *.eva.

Decoding of the compressed files is done in the reverse order. The video is decoded to raw RGB-frames, the pixel values for each frame are read and scaled to the original min-max using the metadata. The resulting data is stored in a netCDF file.

D. *Land-sea mask*

Extra care was needed to properly encode ocean-oriented datasets. Land, or space below the sea-bed, is encoded in source data as invalid or Not a Number (NaN). In an early iteration of EVA, these data points were converted to white (R,G,B = 255). However, due to compression artifacts of the video codec, this resulted in a changing and inconsistent sea bed in the decoded METOC files.

As a solution we filled empty grid points using a nearest neighbor algorithm, and encoded the position of these points in the metadata, using the lossless Lempel-Ziv-Markov chain algorithm (LZMA) [16].

E. *How to compare codecs*

Previous research [9, 10] described lossy compression algorithms that performed well in file size versus data quality. But this came at a price of much processing time. Video codecs are developed for streaming media and are time-efficient by design. So when applied to METOC data the question is: what codec performs best in file size versus quality?

1) *Compression factor*

The compression factor is ratio between the original file size and the reduced file size. For METOC data distribution via email or satellite communication, a typical compression factor is in the order 100. This matches with the normal use of codecs in streaming media.

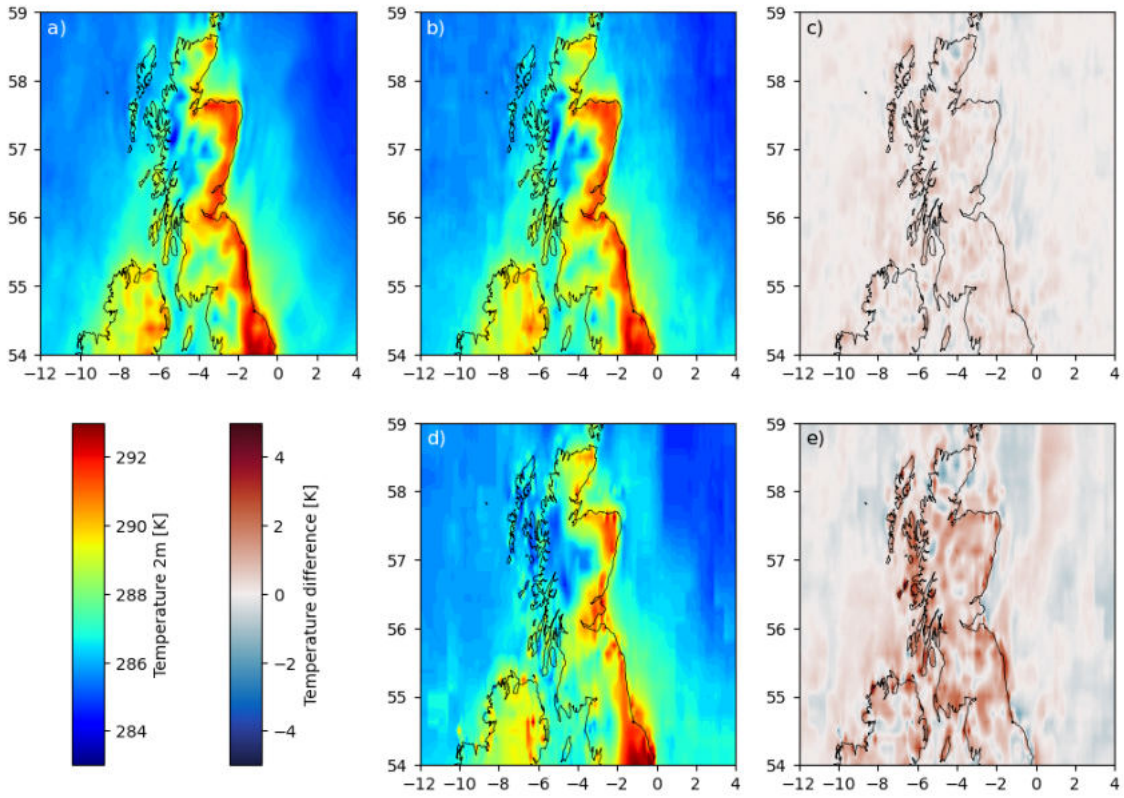

Fig 2. Surface temperature at 2m on 10th of June 2022 at 12:00 UTC. a) Original data. b) Compressed and decompressed temperatures, using EVA with a compression factor of 25. c) Difference in surface temperature, between figure a and b. d) Compressed and decompressed temperatures, using EVA with a compression factor of 100. e) Difference in surface temperature between figure a and d.

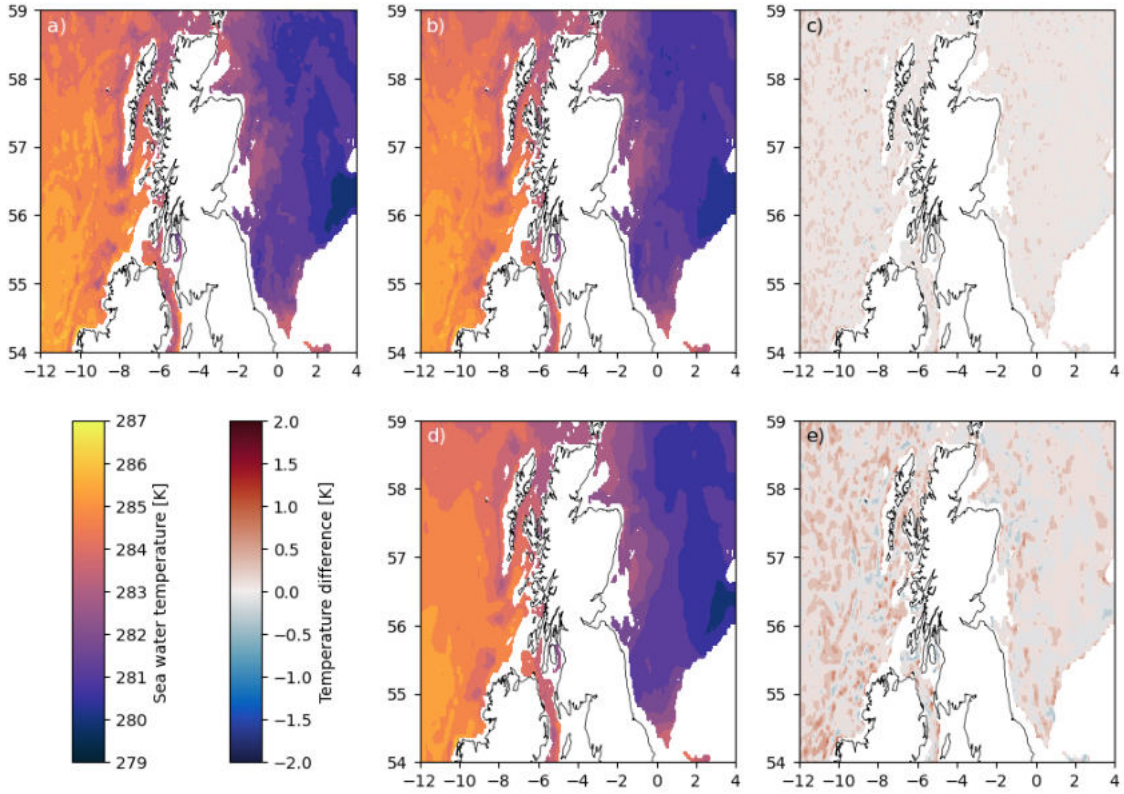

Fig 3. Sea water temperature at a depth of 60m on 10th of June 2022 at 12:00 UTC. a) Original data. b) Compressed and decompressed temperatures, using EVA with a compression factor of 25. c) Difference in sea water temperature, between figure a and b. d) Compressed and decompressed temperatures, using EVA with a compression factor of 100. e) Difference in sea water temperature between figure a and d.

2) Information loss

Information loss due to compression can be measured by comparing original data Y with decompressed data $\hat{Y}$. Common measures for information loss are the mean squared error (MSE) and the peak-signal to noise ratio (PSNR):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (1)$$

$$PSNR = 10 \cdot \log_{10}\frac{(Y_{\max}-Y_{\min})^2}{\mathrm{MSE}} \quad (2)$$

where Y and $\hat{Y}$ count n data points and Y_{max} and Y_{min} are the largest and smallest values in Y. The MSE is an intuitive measure: the unit of the root MSE is the same as for the data in Y and $\hat{Y}$ and the error drops to zero when there is no information loss. The root MSE is preferred when comparing losses for one parameter, since it has the same unit as the parameter. However when comparing the losses for multiple parameters, with different units, the PSNR is a better candidate. The PSNR is expressed in dB and grows higher when the losses are smaller. In our use case we defined a PSNR of 32 dB as limit for acceptable quality, but different use cases might prefer other limits.

III. Results and Discussion

A. Making METOC data portable with video codecs

First of all, it is absolutely feasible to use video codecs on METOC data. The EVA prototype can quickly compress large environmental datasets into small files, do a fast decompression, and maintain much of the relevant information (see Fig 2 and 3). At higher compression rates, video artifacts become more noticeable and for most applications these are acceptable. This approach makes METOC data portable in the sense that file sizes become smaller and storage and distribution much easier.

B. Performance comparison of video codecs

Using EVA, six video codecs were compared on their performance in terms of file size versus quality. Each codec was used on the same benchmark of METOC data. For a range of requested compression factors the information loss was evaluated by the PSNR. The results are plotted in Fig. 4.

The best performing codec has the highest PSNR. Depending on the demanded compression factor, libx264 and Snow outperform the others. However, with a general difference of 3 to 5 decibels it is a close call. The performance of Snow peaks above the others but only for a limited range of compression factors. Above a compression factor of 250,

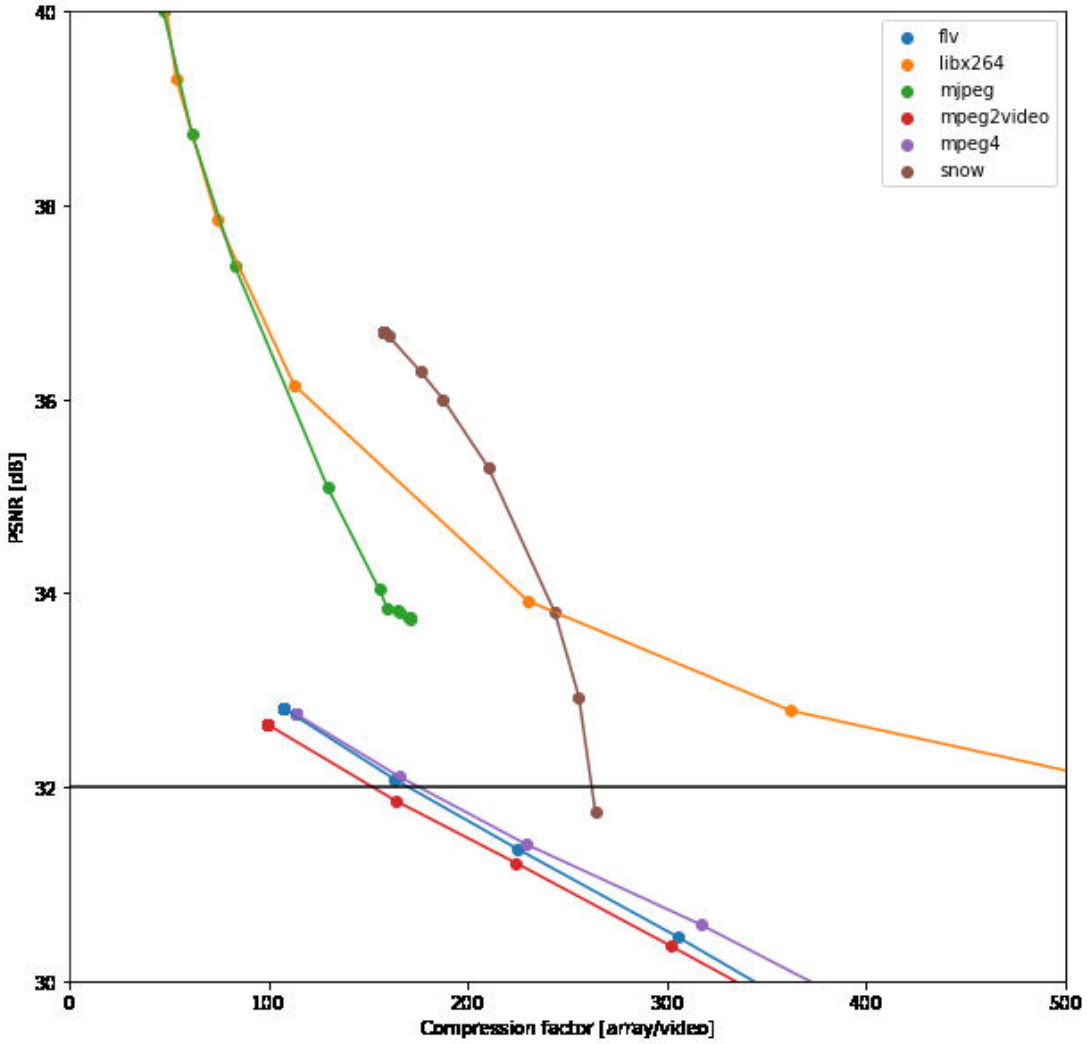

Fig. 4. Performance comparison of six video codecs on a benchmark of combined meteorological and oceanographic data.

datasets that are compressed with Snow quickly loose information. In contrast, libx264 has a much wider range of good performing compression factors and has become the codec of choice in EVA.

C. *Bit Depth*

A significant contributor to file reduction is the color bit depth in the algorithm. This value describes how many bits are used to encode the color of a pixel, with more bits allowing for more precise colors. In most applications (including EVA), 24-bit color is used, giving us 8-bits per color channel, or values 0 to 255. However, some professional cameras are now using 30 or 36-bit color, with color values ranging 0 to 1024 or 2048 per channel.

In EVA, using 10 bits instead of 8 would allow for larger intervals in the scaling of the parameters and potentially result in less information loss. In a quick first analysis using 10 bits in the H.266 codec, we found little impact on the high compression factors. EVA scales each slice of data to 8 bit values and impact of 2 more bits to the precision is apparently neglectible when compared to the loss due to high compression factors.

D. *Availability of EVA*

EVA is available in Github at https://github.com/datalab-nld/eva .

IV. Conclusion

It is absolutely feasible to use video codecs on METOC data. The EVA prototype can quickly compress large environmental datasets into small files, do a fast decompression, and maintain much of the relevant information. This approach makes METOC data portable in the sense that file sizes become smaller and storage and distribution much easier.

Acknowledgment

This study has been conducted using E.U. Copernicus Marine Service Information; https://doi.org/10.48670/moi-00016.

References

[1] P. Bauer, A. Thorpe & G. Brunet, "The quiet revolution of numerical weather prediction". Nature 525, 47–55 (2015) doi: https://doi.org/10.1038/nature14956.

[2] E.W. Blockley, M.J. Martin, A.J. McLaren, A.G, Ryan, J. Walters, D.J. Lea, I. Mirouze, K.A. Peterson, A. Sellar, D. Storkey, "Recent development of the Met Office operational ocean forecasting system: an overview and assessment of the new Global FOAM forecasts" Geosci. Model Dev. 7, 2613–2638 (2014). doi: https://doi.org/0.5194/gmd-7-2613-2014.

[3] E. Mokole, "Technical Evaluation for 'Bridging the Gap between the Development and Operational Deployment of Naval Tactical Decision Aids'", STO TER REPORT, SET-244 Symposium, Den Helder (2017).

[4] R. E. Marshall, W. D. Thornton, G. Lefurjah, and T. S. Casey, "Modeling and simulation of notional future radar in non/standard propagation environment facilitated by mesoscale numerical weather prediction modeling,," in Naval Engineers Journal, vol. 120, pp. 55–66 (2009) doi: https://doi.org/10.1111/j.1559-3584.2008.00165.x.

[5] E. H. Burgess & K. L. Horgan, "Applying Numerical Weather Prediction Data to Enhance Propagation Prediction Capabilities to Improve Radar Performance Prediction", NATO STO-MP-SET-244-7A, Den Helder (2017).

[6] A. V. van Leijen & T. Wilbrink, "Royal Netherlands Navy Sonar TDA's: Past, Present and Future", NATO STO-MP-SET-244-11B, Den Helder (2017).

[7] X. Delaunay, A. Courtois, F. Gouillon, "Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netCDF-4 or HDF5 files". Geosci. Model Dev. 12, 4099-4113 (2019). doi: https://doi.org/10.5194/gmd-12-4099-2019.

[8] S. Sullivan, "Comparison of Netcdf4 (HDF5) and Grib2 compression methods for meteorological data", UCAR report, June, 2011.

[9] M. Boone, "De Koninklijke Marine voor de wind - Het comprimeren van datasets van atmosfeermodellen om operationeel gebruik op zee mogelijk te maken", B.Sc. thesis, Amsterdam University of Applied Sciences, June 2016 (in Dutch).

[10] A.V. van Leijen, M. Boone and K.L. Horgan, "Making numerical weather predictions portable, compression of weather data for use in radar propagation modeling" 2017 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), San Diego, CA, (2017), pp. 1-2, doi: https://doi.org/10.1109/USNC-URSI.2017.8074867.

[11] H. Hersbach, B. Bell, P. Berrisford, G. Biavati, A. Horányi, J. Muñoz Sabater, J. Nicolas, C. Peubey, R. Radu, I. Rozum, D. Schepers, A. Simmons, C. Soci, D. Dee, J-N. Thépaut, "ERA5 hourly data on single levels from 1940 to present." Copernicus Climate Change Service (C3S) Climate Data Store (CDS), (2023). doi: https://doi.org/10.24381/cds.adbb2d47.

[12] Copernicus Climate Change Service, Climate Data Store, "ORAS5 global ocean reanalysis monthly data from 1958 to present." Copernicus Climate Change Service (C3S) Climate Data Store (CDS), (2021). doi: https://doi.org/10.24381/cds.67e8eeb7.

[13] H. Zu, F. Abdul-Kadar and P. Gao, "An information model for managing multi-dimensional gridded data in a GIS." IOP Conference Series: Earth and Environmental Science. Vol. 34. No. 1. IOP Publishing, 2016. https://doi.org/10.1088/1755-1315/34/1/012041.

[14] I.E. Richardson, "Video codec design: developing image and video compression systems" John Wiley & Sons, 2002.

[15] Hoyer, S. & Hamman, J., "xarray: N-D labeled Arrays and Datasets in Python", Open Research Software, vol. 5(1), pp. 10 doi: https://doi.org/10.5334/jors.148.

[16] I. Pavlov, "LZMA Specification in LZMA SDK", 2013. https://7-zip.org/sdk.html.