Using of Google Earth Engine in monitoring systems

Elena Fedotova[1*], and *Anna* Gosteva[1,2]

[1]FRC KSC SB RAS, Krasnoyarsk, Russia
[2]SFU, Krasnoyarsk, Russia

Abstract. Google Earth Engine (GEE) cloud service is a powerful tool for environmental research. An example of using GEE to solve a typical research problem is shown. The following data extraction and analysis operations were used: filtering data from sets, constructing functions, building graphs, selecting data using vector and raster masks. GEE interface in the form of JavaScript code was used. Correlation between surface runoff and precipitation and snow depth in areas with forest dieback was analysed for Krasnoyarsk region in Russia (r = 0.30 for precipitation and r = 0.57 for snow depth).

1 Introduction

Cloud GIS offers powerful software and hardware for solving environmental monitoring and research tasks by implementing geoinformatics methods: open access geographic servers with electronic map generation and implementation of multivariate analysis algorithms. Cloud GIS allows to receive data in real or near-real time and display them directly on the user's system. The Google Earth Engine platform [1] is unique as an integrated platform for empowering not only remote sensing professionals, but also a much wider audience that lacks the technical capabilities required to use traditional supercomputers or large-scale cloud computing resources [2].

GEE includes an interactive application server working with open data catalogue, computing environment in the form of a code editor (integrated development environment (IDE) for writing and running scripts), geospatial API (client libraries provide Python and JavaScript wrappers for web APIs in REST architecture).

The data catalogue contains a large repository of publicly available geospatial datasets, including images from a variety of satellite and aerial systems in both optical and microwave wavelengths, environmental variables, weather and climate forecasts, land cover maps, topographic and socio-economic datasets. All of this data is pre-processed into a ready-to-use form that provides efficient access and removes many barriers associated with data management. Operators are implemented in a parallel processing system that automatically splits and distributes computations, providing high-throughput analysis capabilities. Users access the API through a thin client library or through an interactive web development environment built on top of this client library [2].

* Corresponding author: elfed@ksc.krasn.ru

Data collections have descriptions and sample JavaScript codes for opening and rendering data in GEE. GEE client libraries are proxy objects for images (ee.Image), collections (ee.ImageCollection, ee.FeatureCollection), and other data types: numbers (ee.Number), strings (ee.String), vector objects with geometry (ee.Geometry), lists (ee.List). Images grouped by source, destination, etc. are combined into image collections (ImageCollection), that have the functionality to select data from the collection by filtering and sorting according to specified spatial, temporal, or other criteria. The API library allows performing operations on images using raster algebra and support of higher-order functions: map () and iterate (), that allow to apply arbitrary functions to image collections. The reduce () operator is used to calculate statistics on image collections, which, for example, can aggregate data using the sliding window method [2, 3]. Users can load their own data (both raster and vector type) using the library of operators provided by the Earth Engine API. Users actively use GEE to work with remotely sensed data. [4].

Figure 1 shows a working window, which displays a table of user's scripts, a part of the JavaScript code for data visualization, an image in a map window and a graph of parameter's values in console.

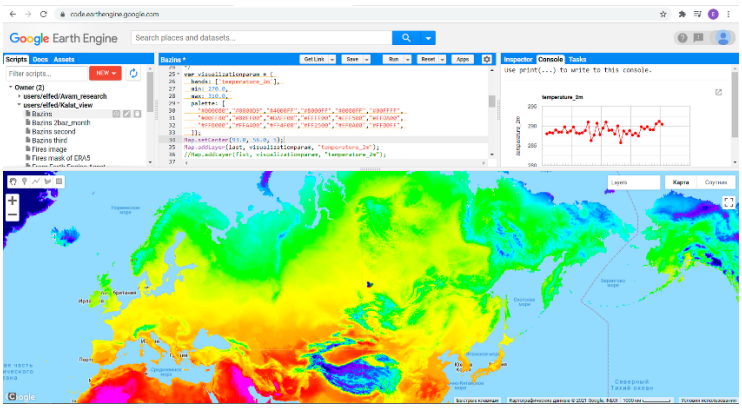

Fig.1. The Earth Engine interactive development environment

Thus, users have a set of analysis tools for spatial data visualization and processing. With GEE it is possible to create a very wide range of GIS analyses.

2 Data and methods

In order to develop a universal set of simple operations in GEE for solving monitoring tasks, a typical task of analysing the relationship between two environmental parameters - the dynamics of surface runoff and the dynamics of vegetation in a river basin - was set.
Surface flow dynamics in the areas where forest cover was destroyed in 2013 and 2016 was analysed. The regions were identified according to the Hansen forest cover change project [5]. These are the territories of the Kas and Sym basins, two left tributaries of the Yenisei River in Krasnoyarsk Region, Russia.

Hydrological and climatic data were selected from ERA5-Land Monthly Averaged [6]. The data cover the time interval from January 1981 to August 2021 (at this time, in general three months behind real time) and have a spatial resolution of 0.1 degrees. ERA5-Land is one of Climate Data Store (CDS) land surface dataset, from 1981 for monthly averaged values to present time (2-3 months in arrears), produced at higher resolution (9km) and forced by ERA5 atmospheric parameters with lapse rate correction, but with no additional data

assimilation [7]. Reanalysis datasets are most suitable for spatiotemporally consistent environmental analysis [8]. The ERA5 Land dataset is actively used in climate change research [9-11].

From the parameter values (50 parameters) represented mean daily value for every month in year, five were used:

– surface runoff, is the depth the water would have if it were spread evenly over the grid box, m;

– total precipitation, accumulated liquid and frozen water, including rain and snow, that falls to the Earth's surface, m;

– total evaporation, accumulated amount of water that has evaporated from the Earth's surface, including a simplified representation of transpiration (from vegetation), into vapour in the air above, m;

– air temperature, Temperature of air at 2m above the surface of land, K;

– snow depth water equivalent (SDWE), depth of snow from the snow-covered area of a grid box, m of water equivalent [6].

The sequence of actions is here presented to solve the above task: data selection, their visualization, selection or calculating the required parameters, analysing their dynamics (plotting graphs, calculating statistics).

The data selection from the database and their visualization is performed using filtering operation. The characteristics of the data in the database and codes for this operations are presented in the GEE for all data collections. Five above parameters from the ERA5-Land database from 2006 to 2020 from March to November for each year were separated. JavaScript code for surface runoff is shown (fig.2).

```
// Create an ImageCollection from an Earth Engine Table.
var dataset = ee.ImageCollection("ECMWF/ERA5_LAND/MONTHLY")
                .filter(ee.Filter.date('2006-01-01', '2020-12-31'))
                .filter(ee.Filter.calendarRange(03, 11,'month'));
//select parameter for obteining
      var parameter = dataset.select('surface_runoff');
```

Fig. 2. JavaScript code of surface runoff filtering from 2006 to 2020 years for March-November

In the next step, it is necessary to extract characteristics for objects (by masks) from the filtered data: by vector objects - points, including the selection of pixels in the Inspector, along a line - profiles, by a polygon. To highlight the analysed area, it can be used raster masks from other raster datasets. In our task data from ERA5–Land DB were selected using the raster deforestation map [3] as a raster masks (fig.3).

```
var image = ee.Image("users/elfed/mask101");
var lossArea = image.select('b1');
var collection_ERA5 = parameter.map(function(img) {
                  return img.updateMask(lossArea);
            });
```

Fig. 3. JavaScript code for raster mask realization: mask101 – forest loss raster map, b1 – name of layer in mask101.

Values of five above mentioned parameters for 2006-2020 corresponding to this mask were exported from the GEE in csv format using print operation.

Reduce operation is used for per-pixel calculation of the statistic for every image in image collection. Collection of images is reduced to a single image. For basic statistics like min, max, mean, etc., ImageCollection has quick access methods like min (), max (), mean (), etc. Both built-in and user-created functions are used to calculate specific characteristics of the

data, such as vegetation indices. The function is applied to every image in the collection. Function to calculate daily values of runoff, total precipitation, total evaporation in millimetres and monthly values was made (fig. 4).

```
// Create function (*30 - number of days in month, *1000 - mlm)
var mult30 = function(image) {
  var mult = ee.Image(0).expression(
    'runoff * 30*1000', {
      runoff: image.select('runoff'),
});
  return image.addBands(mult.rename('mult_mil_mon'));
};
print(Chart.image.series(parameter,      Kas,      ee.Reducer.mean(),
30).setOptions(options));
```

Fig. 4. JavaScript code for printing the graph of monthly mean values of surface runoff in millimetres in Kas river basin.

It is possible to output a graph with several values, for every polygon in the vector layer, for example to facilitate export in csv format (fig. 5).

Thus, for each step of data analysis JavaScript code has been created using GEE data descriptors and other auxiliary tools. By changing parameter names, time ranges, palettes for map visualization, equations in functions, the above scripts can be used for other data.

```
var Mothpoly = ee.FeatureCollection("users/elfed/Moth");
// Create a ImageCollection from an Earth Engine Table.
var dataset = ee.ImageCollection("ECMWF/ERA5_LAND/MONTHLY")
                .filter(ee.Filter.date('2006-01-01', '2020-12-31'))
                .filter(ee.Filter.calendarRange(1, 12,'month'));
var parameter = dataset.select('temperature_2m');
// Define the chart and print it to the console.
var chart =
    ui.Chart.image
        .seriesByRegion({
          imageCollection: dataset,
          band: 'temperature_2m',
          regions: Mothpoly,
          reducer: ee.Reducer.mean(),
          scale: 500,
          seriesProperty: 'Id',
          xProperty: 'system:time_start'})
        .setOptions({
          title: 'temperature_2m',
          hAxis: {title: 'Date', titleTextStyle: {italic: false, bold:
true}},
          vAxis: {
            title: 'temperature_2m',
            titleTextStyle: {italic: false, bold: true}},
          lineWidth: 5,
          colors: ['f0af07', '0f8755', '76b349'],});
print(chart);
```

Fig. 5. JavaScript code for printing the graph of mean air temperature for three polygons (2013, 2016, 2017 years of forest loss) in Moth vector layer.

3 Results and discussion

Extracted from ERA5–Land DB data were analysed in Microsoft Excel. Surface runoff anomalies were calculated, only for April – October months of each year.

The results are demonstrated in the fig. 6. A noticeable change (both a decrease and an increase) in the runoff value for the year following the destruction of forests is not observed for either 2013 or 2016. There is some correlation with precipitation in the same month (r = 0.30) and sum of SDWE in October-May (r = 0.57). The high level of May runoff in 2014 and 2015 in both areas occurs against the background of a high level of SDWE sum from October to May.

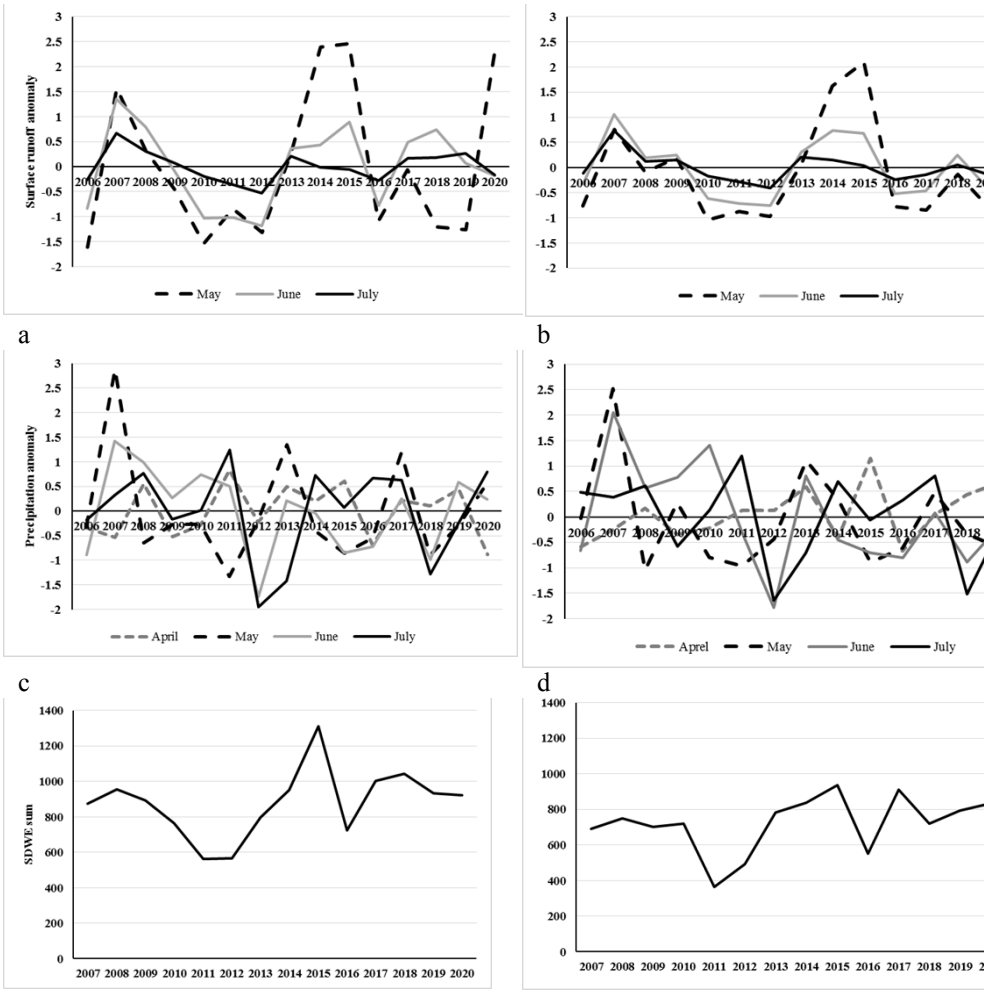

Fig. 6. Dynamics of surface runoff anomalies in areas of deforestation in 2013 and 2016 (a, b). Precipitation anomalies (c, d) and sum of SDWE for October – May (e, f). Surface runoff and precipitation – daily average values for a month.

Visual analysis of the 2015-2020 winter Landsat images was carried out in GEE without downloading to the user's computer. As can be seen in the winter satellite images (fig. 7), forest mortality results in a gradual change of the land cover over several years. Only

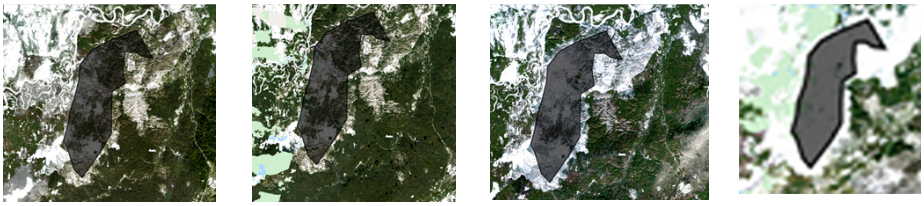

Fig. 7. Forest areas with damaged in 2013 year stands on the Landsat images of 2015-2020 years (true color).

in the 2020 image the site looks like treeless. This gradual change in the surface may also result in no abrupt change (increase) in surface runoff.

The research was funded by RFBR, Krasnoyarsk Territory and Krasnoyarsk Regional Fund of Science, project number 20-44-240007.

References

1. Google Earth Engine, https://earthengine.google.com/
2. N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, R. Moore, Rem. Sens. of Envir,. **202,** 18 – 27 (2017)
3. V. Lobanov, https: habr.com/ru/post/500020/ (2020)
4. A.A. Jamali, R.G. Kalkhajeh, T.O. Randhir, S. He, J. Environ. Manag., **302**, 113970 (2022)
5. 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, Sci, **342,** 850 – 853 (2013)
6. ERA5-Land Monthly Averaged - ECMWF Climate Reanalysis, https://developers.google.com/earth-engine/datasets/catalog/ECMWF_ERA5_LAND_MONTHLY?hl=en#description
7. ERA5 dataset, https://confluence.ecmwf.int/display/CKB/ERA5
8. H. Zandler, T. Senftl, K.A. Vanselow, Sci, **10**, 22446 (2020)
9. S. Gleixner, T. Demissie, G.T. Diro, Atmosphere, **11**, 996 (2020)
10. A.K. Betts, D.Z. Chan, R.L. Desjardins, Front. Environ. Sci. (2019)
11. M.H. Pesci, F. Voges, N. Rüther, K. Förster, EGU General Assembly, https://doi.org/10.5194/egusphere-egu2020-2405 (2020)