Computer Science Tripos

Part II Project Proposal Coversheet

Please fill in Part 1 of this form and attach it to the front of your Project Proposal.

Part 1

Name:	George Pool	CRSID:	gp528
College:	Emmanuel College	Project Checkers:(Initials)	afb21, ndl32
Title of Project:	Effective geospatial code in OCaml		
Date of submission:	18/10/2024	Will Human Participants be used?	No
Project Originator:	Anil Madhavapeddy		
Project Supervisor:	Michael Dales, Patrick Ferris		
UTO Supervisor:	Anil Madhavapeddy		
Director of Studies:	Thomas Sauerwald, Anthony Harris		
Special Resource Sponsor:			

Part 2

Project Checkers are to sign and comment in the students comments box on Moodle.

Part 3

For Teaching Admin use only

Date Received:		Admin Signature:	

Effective geospatial code in OCaml

George Pool (gp528)

October 11, 2024

Project supervisors: Michael Dales (mwd24), Patrick Ferris (pf341)
UTO: Anil Madhavapeddy (avsm2)
College: Emmanuel College
Director of Studies: Thomas Sauerwald (tms41), Anthony Harris (awh28)

1 Introduction and Description

Geospatial processing is critical in environmental science and is used in a range of environmental scenarios, both in areas of carbon accounting like digital Monitoring, Reporting and Verification in Reforestation Carbon Removal, and also biodiversity analysis like calculating worldwide extinction rates.

However there are problems with the current state of affairs, Python is one of the most popular data science libraries, and when doing geospatial data science with Python you use the GDAL geospatial data library. This workflow is not ideal [2]. For environmental scientists, interacting with the geospatial data in the imperative way that Python requires can be very difficult, and it is also not designed for the scale of computation needed for large geospatial projects. Furthermore, Python's dynamic type system means that tricky type errors due to heterogeneity of data points can go unchecked. GDAL also only allows synchronous IO for loading GeoTIFF files, which can hurt performance.

I will write a non-blocking library for reading GeoTIFF files in OCaml (built on top of the WIP OCaml-TIFF library[1] written by one of my supervisors, Patrick Ferris). I will then use OCaml to write an embedded Domain Specific Language (eDSL) which will be used to interact with geospatial data in a declarative way. The language (if all extension tasks are completed) will be able to express tricky tasks in GDAL like:

- Deciding if data sets overlap and allowing an intersection / union of these (core task)
- Rasterization of vector layers (early extension task)
- Automatic resource management of large geospatial datasets (later extension task)

While the Yirgacheffe library[2] in Python has been written by one of my project supervisors, Michael Dales, to do similar tasks, it requires interacting with the data in Python's imperative way. On the other hand, since my eDSL will be declarative its interaction should be more intuitive for environmental scientists, describing "what" they want to happen to do, instead of having to wrangle with the mechanics of GDAL's implementation. Furthermore, OCaml's type checker should be able to catch the type errors that Python's dynamic typing cannot.

[1] https://github.com/geocaml/ocaml-tiff
[2] https://github.com/quantifyearth/yirgacheffe

2 Starting Point

OCaml

I have not used OCaml beyond the courses in IA and IB Computer Science so far. I understand the concepts behind functional programming through these courses but I have not used it to create a large scale program before.

Geospatial analysis

I have not written any code yet. As mentioned in the background, the project will be inspired by the Yirgacheffe Python library (written by my supervisor Michael Dales) so the preparatory work has been talking with him and my other supervisors to familiarise myself with the context for this project and geospatial programming.

3 Substance and Structure

The main components of the project are as follows:

- I firstly have to create a way to interact with the geospatial data files in OCaml. To get around the synchronous nature of GDAL, I am going to instead build on top of the WIP OCaml-TIFF library to allow for loading geoTIFFs with OCaml. The first implementation of this will allow for a subset of GeoTIFF files (namely uncompressed ones of a single projection), and extensions can allow for LZW compressed GeoTIFF files and for different projections of GeoTIFF files
- I will then create the eDSL for interacting with the geospatial datasets declaratively, allowing the common interactions with the geospatial datasets like intersections and unions of datasets, and (as an extension) rasterization of vector layers
- Create the scheduler for parallel operations / memory management which allows for the automatic resource management (again as an extension)

4 Success Criteria

Core criteria

- My GeoTIFF loading implementation can do batch processing on multiple GeoTIFFs faster (because of OCaml's non-blocking IO) than GDAL (because of GDAL's synchronous loading then processings schema)
- My project can catch typing errors through OCaml's static typing system that Python's dynamic typing would not catch
- The project must have the eDSL which allows you to union/intersect data sets

I will evaluate the project using an existing geospatial task centered around biodiversity metric calculation [1]. I will compare the visualisations using the previous GDAL Python pipelines and the new OCaml pipelines. If it is successful I will be able to create pixel by pixel identical visualisations, but instead with my eDSL instead of imperative Python code

Extension criteria

- Allow rasterization of vector layers and supporting more geospatial data types
- Support for compressed GeoTIFF files
- Support for multiple GeoTIFF projection algorithms
- Allow smarter resource management (e.g. be able to load and interact with a geoTIFF that is too large for the laptop if loaded normally but the library can deal with it).

5 Plan of Work

I have created a plan for my project, breaking it down into 15 work packages, each package having a date range of two weeks, a plan of work, and (usually) a deliverable for the end of the work package.

Work packages

Dates	Work to do	Deliverables
10/10/2024 - 23/10/2024	Submit final proposal. Carry out preparation work by looking at chapters of Real World OCaml relevant to my project. Review the OCaml-TIFF library that I will be building on top of. Continue studying Yirgacheffe. *Note: First lecture on 10/10*	Final proposal
24/10/2024 - 6/11/2024	Create my OCaml-TIFF implementation to allow loading GeoTIFFs into OCaml.	Ability to load uncompressed GeoTIFFs into OCaml.
7/11/2024 - 20/11/2024	Contingency for finishing my OCaml-TIFF implementation. Design the eDSL and plan the functions that I will create.	Design doc of eDSL that I will show to my supervisors.
21/11/2024 - 4/12/2024	Write eDSL for simple union / intersection of layer operations. *Note: last lectures on 4/11*	Ability to do union / intersection of layers with the eDSL
5/12/2024 - 18/12/2024	Contingency	
19/12/2024 - 8/1/2025	Holiday (and extra contingency if necessary)	
9/1/2025 - 22/1/2025	Write Progress Report and begin work on extensions.	Progress Report and Presentation ready
23/1/2025 - 5/2/2025	Continue work on extensions. *Note: First lecture on 23/1. Progress report deadline upcoming*	Report to supervisor on feasibility of extensions
6/2/2025 - 19/2/2025	Finalise work on extensions. Write introduction and preparation sections of dissertation	First section of dissertation is drafted
20/2/2025 - 5/3/2025	Write implementation and evaluation.	Implementation and evaluation are drafted
6/3/2025 - 19/3/2025	Contingency for drafting the main body of dissertation. Write conclusion.	First draft ready and handed in to supervisors
20/3/2025 - 2/4/2025	Act on first set of feedback and submit my next version of my dissertation.	Second draft ready and handed in
3/4/2025 - 16/4/2025	Act on the second set of feedback and have the final version of the dissertation ready.	Final version of dissertation ready.
17/4/2025 - 30/4/2025	Contingency. Submit dissertation.	Dissertation submitted and finished.
1/5/2025 - 15/5/2025	Emergency Contingency. *Note: Dissertation deadline on 19/5*	

6 Resource Declaration

I will be using my own laptop (Dell XPS 15 9510, 64GB RAM) for my project. The contingency plans will include backing up to GitHub as well as Google Drive. *I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure.* I may also use the Sherwood or Kinabalu cluster machines (AMD EPYC 7702 64-Core Processor, 1 TB RAM) for testing, which my supervisors can give me access to.

References

[1] Alison Eyres, Thomas Ball, Michael Dales, Tom Swinfield, Andy Arnell, Daniele Baisero, América Paz Durán, Jonathan Green, Rhys E Green, Anil Madhavapeddy, and Andrew Balmford. Life: A metric for quantitatively mapping the impact of land-cover change on global extinctions, 2024.

[2] Patrick Ferris, Michael Dales, Sadiq Jaffer, Amelia Holcomb, Eleanor Toye Scott, Thomas Swinfield, Alison Eyres, Andrew Balmford, David Coomes, Srinivasan Keshav, and Anil Madhavapeddy. Planetary computing for data-driven environmental policy-making, 2024.