Using Data Lineage for Sub-image Processing

Johnson Mwebaze[1,2], John McFarland[2], Danny Boxhoorn[2],
Hugo Buddelmeijer[2], and Edwin Valentijn[2]

[1] Makerere University, P.O. Box 7062, Kampala, Uganda
[2] University of Groningen, Landleven 12, 9700 AV Groningen, The Netherlands

Abstract. In the paper, we show that lineage data collected during the processing and analysis of datasets can be reused to perform selective reprocessing (at sub-image level) on datasets while the remainder of the dataset is untouched, a rather difficult process to automate without lineage.

1 Introduction

In some scientific applications, most often users are interested in a source (e.g., moving, variable, or extreme in some colour index) that lies on a few pixels of an image. The approach adopted by most observation systems is processing the entire image or set of images even when the sole source of interest may exist on only a few pixels of one or a few images [6] [1]. Accordingly, out of millions of images in a survey, it is nearly impossible and wasteful to process the whole data volume. Instead of processing the whole dataset, a user should only select, retrieve and process only relevant pixels on an image where the source exists. However pipelines have been written and designed for instruments with fixed detector properties (e.g.,image size, calibration frames, overscan regions, etc.). All metadata and processing parameters are based on an instruments or a detector, moreover some image operations can not be done on a sub-image level.

Therefore, to perform processing at sub-image level, we make use of lineage data to assemble the sub-image processing pipeline and to select all necessary inputs to the pipeline. By matching and retrieving existing pre-processed information in the system and knowing the relationship between what we want to process and what has been processed before, we are then able to determine the difference between pipelines (and objects). We can then modify any new pipelines/objects/parameters so that the new processing follows the new user processing requirements for a particular region on the image.

Data lineage (provenance) is a well-defined problem with known solutions as pointed out in recent workshops [4] and surveys [5]. The use of provenance has also gained significant attention [4]. Several workflow management systems (e.g [2, 3, 7, 9]) do exploit provenance information for different purposes. To the best of our knowledge, this is the first work that leverages lineage information to support sub-image processing to simplify and automate the reprocessing of objects. Since we are working with pixels, this framework required lineage at pixel level. We extended our lineage model presented in [8] to trace lineage at pixel level and then used pixel lineage for sub-image processing.

D.L. McGuinness, J.R. Michaelis, and L. Moreau (Eds.): IPAW 2010, LNCS 6378, pp. 289–291, 2010.

2 Sub-image Processing

Sub-image processing requires the ability to store and mine provenance data. Since no input data, metadata and parameters exists for sub-images, we use data from previous runs to enable sub-image processing. The underlying assumption is that the target has been processed before as part of a full image and probably a user would like to carry out a detailed analysis or a computation to a target that lies on few pixels of an image (or sub-image). The pipeline for sub-image processing is thus built based on lineage data. Likewise all input data, attributes and parameters to be used in the subimage pipeline is selected from the same lineage data. However, since we are processing a sub-image, some of the input data, parameters and parts of the pipelines have to be modified. When such changes occur then the part of pipeline affected by the changes must be re-run. The rerun will take data dependencies into account and only execute those parts of the workflow affected by the changes.

The starting point of sub-image processing is the selection of the target. i.e. set of sky positions. The system then builds a directed graph representing the data dependencies with nodes representing objects and edges representing all dependencies attached to an object. The graph begins with the topmost node, which is the target to be made. New edges are added starting at this trigger and expanding outward, using the dependency logic derived from lineage data. The dependency graph is built and checked recursively till the last dependency (in this case raw data from the telescope).

Each node in the graph is associated to an object. Each object is identified with a unique ID. Using this unique ID, we can query for all data that went into the processing of the object. The queried data is modified and used as input to the sub-image pipeline. This data also includes the software code and version that was used. If a unique ID is associated with an image, a cutout is made of the pixels of interest from this image and used as input to the module. The pixels extracted as a cutout are determined through pixel lineage.

However, in some cases input to the sub-image pipeline might be selected from any other related processing. For example, a critical step for astronomical processing is deriving an astrometric solution. This is derived by fitting distortion polynomials to images, taking into account the objects seen. For accurate results, several reference stars are used to derive the final solution. For the case of sub-image processing, such a process would fail since reference stars on the sub-image will be very few. Therefore, in such cases, we use the astrometric solution of another set of images of the same field, processed using the same parameters as needed to process the sub-image. The solution is then modified and fitted to the pixels of the sub-image.

After assembling the pipeline and collecting all necessary input data, the sub-image is processed. Source extraction is then run on the sub-image resulting in a new catalog of sky positions, and/or any other user specific processing done on the sources extracted.

3 Use Case: Analyzing Transitioning Galaxies

We demonstrate the use of provenance using a usecase of analyzing transitioning galaxies. These are galaxies that fall into galaxy clusters that interact with their environment. Initially a full image is processed and an initial photometric catalog of the sources on the image is extracted. The density of galaxies around each source is calculated using the galaxy position. The magnitudes and densities of galaxies that undergo a transitional phase can be identified. During processing of the full image the system records all lineage for this task and therefore a provenance graph can be queried and displayed for this image. Out of hundreds of galaxies that were observed in this processing only transitioning galaxies will then be further analyzed by extracting sub-images from the raw images where these galaxies lie and reprocessing only these sub-images to estimate more complex and time consuming parameters such as quantifications of the internal structure of the galaxy. To identify the images required for this task and the position of the galaxy in all images, we work backwards from the galaxy through all the dependencies. The other inputs (any other sub-images, calibration objects, processing parameters, etc) to the sub-image pipeline are also selected from the initial lineage recorded during the initial processing of the full image. By performing selective processing we save hours/days/weeks of computational time.

References

1. Astro-wise portal, `http://www.astro-wise.org/portal/aw_prompt.shtml`
2. Anderson, E.W., Ahrens, J.P., Heitmann, K., Habib, S., Silva, C.T.: Provenance in comparative analysis: A study in cosmology. Computing in Science and Engg. 10(3), 30–37 (2008)
3. Ellkvist, T., Koop, D., Anderson, E.W., Freire, J., Silva, C.: Using provenance to support real-time collaborative design of workflows, pp. 266–279 (2008)
4. Freire, J., Koop, D., Moreau, L. (eds.): Provenance and Annotation of Data and Processes: Second International Provenance and Annotation Workshop. Springer, Heidelberg (2008)
5. Freire, J., Koop, D., Santos, E., Silva, C.T.: Provenance for computational tasks: A survey. Computing in Science and Engineering 10(3), 11–21 (2008)
6. Greenfield, P.: Reaching for the stars with python. Computing in Science and Engg. 9(3), 38–40 (2007)
7. Groth, P., Miles, S., Fang, W., Wong, S.C., Zauner, K.P., Moreau, L.: Recording and using provenance in a protein compressibility experiment. In: HPDC 2005 Proceedings of the High Performance Distributed Computing, pp. 201–208. IEEE Computer Society, Washington (2005)
8. Mwebaze, J., Boxhoorn, D., Valentijn, E.: Astro-wise: Tracing and using lineage for scientific data processing. In: International Conference on Network-Based Information Systems, pp. 475–480 (2009)
9. Scheidegger, C., Vo, H., Koop, D., Freire, J., Silva, C.: Querying and creating visualizations by analogy. IEEE Transactions on Visualization and Computer Graphics 13(6), 1560–1567 (2007)