NOAO SDM Document PL021

The ODI Demonstration Tier 1 Pipeline

F. Valdes[1], S. Marru[2]

National Optical Astronomy Observatories
Science Data Management

V1.1: September 26, 2010

[1] NOAO Data Products Program, P.O. Box 26732, Tucson, AZ 85732
[2] Pervasive Technology Institute, Indiana University, Bloomington, IN 47408

Table of Contents

List of Figures

Abstract

An ODI tier 1 demonstration workflow, performing basic flux calibration and dither stacking of ODI data is described. One purpose for this demonstration is to show processing of actual ODI format data under the execution frameworks selected for ODI tier 1 processing. Another purpose is to highlight application of IRAF science processing tools for both interactive users and pipeline modules. In particular, to demonstrate IRAF science modules running in a Teragrid environment. This document describes the simulated ODI data, processing workflow, high level structure of the pipeline implementation, and results.

Keywords: pipeline, ODI, NHPPS, OGCE, Teragrid

Purpose of this Document

The purpose of this document is to describe the elements of the ODI demonstration tier 1 pipeline to astronomers and workflow scientists. These elements include the data being processed, the workflow being demonstrated, the processing tools used to build a pipeline, the high level implementation architecture, and the output results of the pipeline. Since a pipeline is essentially a ”black-box” architecture, a real-time demonstration has little visual information other than monitoring tools indicating the state of the processing. Therefore, this document provides a more astronomical and data oriented view of the pipeline, particularly with the figures showing the input and output image data as astronomers are used to visualizing them.

1 Introduction

This document describes an ODI (One-Degree Imager) workflow demonstration providing basic flux calibration of ODI format data and creation of a final image with detector and cell gaps removed by dithering. This workflow is representative of the planned ODI tier 1 pipeline. The ”tier 1” qualifier, which primarily means standard instrumental calibration and stacking, will be dropped in the remainder of this document. In this document we present a description of the ODI demonstration data, the processing steps performed, a high level outline of the demonstration pipeline structure, and the results of running the pipeline on the demonstration ODI data. This description goes along with an available real-time demonstration.

Readers, particularly astronomers, may wish to go directly to appendix A for the figures showing the raw and processed data. As the adage goes – *a picture is worth a thousand words*. The structure and content of this simulated data is described in §3.

The demonstration provides proofs of concept and capability in the following areas:

- the creation of an ODI IRAF data reduction package
- use of IRAF for an ODI workflow in an NHPPS/local cluster context
- use of IRAF for an ODI workflow in an OGCE/NHPPS/Teragrid context
- an NHPPS ODI pipeline application
- an OGCE/NHPPS ODI pipeline application

IRAF is a well-known, highly capable and widely used astronomical data reduction system. The rapid development of a initial ODI package for IRAF, one specifically designed to handle the unique orthogonal transfer arrays (OTA) in the ODI data format, is significant and demonstrates why IRAF is a good choice for the underlying ODI science processing component. It also provides a standard path to distributing user-oriented processing tools to ODI users.

Use of IRAF in NOAO High Performance Pipeline System (NHPPS) pipeline applications is already well demonstrated by existing production pipelines for NOAO mosaic instruments. These applications run on local clusters at NOAO. The proof of concept embodied in this ODI demonstration is use of IRAF in a Teragrid environment using the Open Grid Computing Environment Tools.

The Open Grid Computing Environments (OGCE) project `www.ogce.org` is a provider of science gateway software. It includes several components that can be used by themselves or can be integrated to provide more comprehensive solutions. These components include the OGCE Gadget container, a Google gadget-based tool for integrating user interface components; XRegistry, a registry service for storing information about other online services and workflows; XBaya, a workflow composer and enactment engine; GFAC, a factory service that can be used to wrap command line-driven science applications and make them into robust, network-accessible services; and the OGCE Messaging Service, which supports events and notifications across multiple cooperating services.

An NHPPS ODI pipeline application is a clearly achievable goal in as much as ODI is simply a much larger mosaic camera than existing NOAO instruments. The special features of the orthogonal transfer array (OTA) detectors only add modest requirements. The NHPPS version of the demonstration pipeline shows the ability to handle the OTA format. This provides a low-risk fallback to the more ambitious OGCE/Teragrid version as well as a high performance development and test capability.

Finally, demonstrating the packaging an NHPPS pipeline application in the OGCE framework as a workflow of services running in a Teragrid environment provides a real proof of concept for the ultimate goal of an ODI science pipeline using the full resources of the Teragrid.

2 The Demonstration Use Case

The use case to be demonstrated is taking a set of raw ODI science exposures from a dither sequence (multiple exposure with the telescope slightly offset between exposures), applying standard instrumental flux calibrations (bias removal and sensitivity calibration), and combining them to produce a final image of the field of view with instrument gaps filled in.

Prerequisites for this case are master bias and dome flat calibration data available from a calibration library. This demonstration by-passes astrometric calibration by arranging that after flux calibration the exposure data are correctly registered apart from simple integer offsets. In other words, the steps of coordinate calibration and remapping to a common sky sampling is implied but not included in the demonstration.

3 The ODI Demonstration Data

The demonstration data is based on the current understanding of the expected ODI data format and basic instrumental characteristics. An ODI exposure consists of 64 files, one per orthogonal charge transfer (OTA) detector. The files are multiextension format (MEF) FITS files with a dataless primary header containing global metadata (i.e. keywords) and 64 image extensions for the ”cells” making up the OTA. The cell images are 624x608 pixels. The first 590x598 pixels are actual detector pixels, the last 34 columns and last 10 rows are electronic overscan. The active pixel regions are physically related by the structure of the OTA to having gaps of 18 columns and 10 rows. This physical relation has the consequence that a single image, with cell gaps, can be constructed without further geometric calibration or metrology.

Figure 4 shows a cell image including the overscan regions. The pixel data are simply noise in both the data and overscan regions. Figure 5 illustrates the layout of the cells in an OTA. The 64 OTAs are arranged in an 8x8 grid.

The demonstration data is produced using Poisson noise, electronic overscan and zero bias levels, and cell amplifier sensitivity variations. The bias and sensitivity levels include slopes to mimic non-constant spatial structure. The bias and sensitivity variations are drawn from a random distribution. For signal a background rate (i.e. a lamp for flat fields or the sky for science data) is specified along with an exposure time. Finally a source sky scene, provided as an image, may be used. In the demonstration science data a deep stack from the KPNO Mosaic Imager is used as the sky scene. Two exposures are simulated using the same scene but with a dither offset between them.

The bias and flat field data are generated with instrumental signatures and then processed with the IRAF ODI package to make master calibrations. Figure 5 shows a single OTA from a raw bias calibration. A calibrated bias version, which is no shown, would look much the same but with the mean bias differences between cells removed. Figure 6 shows a calibrated OTA from the master flat field. This shows the dominant effect of different amplifier responses from each cell.

A view of four OTAs from a single raw science exposure is given in figure 7. This is one of the science exposures used in the demonstration pipeline processing.

The sample ODI data can be created with a full complement of OTAs. But for the purposes of the demonstration a 2x2 grid of OTAs (a quad-OTA or QUOTA format) is created. This also makes display of the data, as in the figures presented here, sensible.

4 The Demonstration Pipeline

The ODI demonstration pipeline application consists of three NHPPS pipelines which are executed as four OGCE services. Before describing the IRAF modules, the NHPPS pipelines, and the OGCE services we start with a brief explanation of the architecture.

We begin with individual host-callable programs called modules. These modules are generally pipeline specific scripts using a general underlying language and tools. For example IRAF is both a scripting language and a large set of astronomical image processing tools. Modules could also be Python or Unix shell scripts. The modules are typically of two kinds; those that orchestrate or organize data, for example by grouping or separating data for parallel processing, and those that do some kind of astronomical or instrumental transformation or analysis on the data.

The modules are connected together by an execution framework. If we don't consider scripting languages as execution frameworks there are two used in our architecture.

The first runs a set of modules, called a sub-pipeline or simply pipeline, on a single node to perform some higher level function such as calibrating a single piece of data. To provide workflow logic with asynchronous and parallel execution of the modules, we use the NOAO High Performance Pipeline System (NHPPS). NHPPS also supports distributed processing on a local cluster which provides a fallback and special processing capability. Figure 1 shows a block diagram of the demonstration pipeline in terms of NHPPS pipelines.

However, to use the resources of the Teragrid, with its complexities of scheduling and management, a second execution framework, the Open Grid Computing Environment (OGCE), is added. The OGCE workflow system (with its XBaya frontend) provides an intuitive and user-friendly graphical interface to browse various application registries (such as the OGCE's XRegistry) and construct task graphs as workflows. The representation is captured as an abstract, high-level, workflow-neutral format, which can be translated into workflow execution specific syntax. In this demonstration, the workflow is translated into a python script and executed locally within the XBaya interface. For production ODI deployment, we will submit the workflow to a workflow engine, the Apache Orchestration and Director Engine (ODE) `http://ode.apache.org/` that has been enhanced to support long running scientific workflows on computational grids. The workflow engine will support multi-level workflow parallelism. In the demonstration, OGCE tools executes the NHPPS pipelines and connects them together across a distributed grid computing network. Figure 2 shows how the the NHPPS pipelines are wrapped as OGCE services and two components which handle parallelization.

For ODI, parallelism will be handled between OGCE and NHPPS based on the data parallelism and resource architectures The nature of data is:

- On the top of the tree: Each campaign consisting of N nights.
- On the bottom of the tree: OTAs within one MEF file consisting of 64 cells that run most efficiently on a single resource.

Based on an empirical study, we will hand off the parallelism between two workflow scripts divided into array of data and distributed efficiently across different computing cores on a cluster node. The hand off decision will be resource architecture dependent like the number of cores on a single node, batch queue scheduling overhead among other characteristics. The higher level workflow representation will split the data according to observations:

- Campaign Data to loop over individual night observations
- Each night loop over filters used
- Each Filter loops over multiple exposures that applied the filter
- Each exposure loop over 64 OTAs.
- Each OTA is processed by handling the 64 containing cells

In this demonstration OTAs are mapped to resources and processed using NHPPS defined pipelines. OTAs are mapped to resources and processed using NHPPS defined pipelines. NHPPS-defined pipelines invoke IRAF and other image processing applications on individual cells. Based on resources, handover from OGCE to NHPPS could occur higher or lower in the stack. To start with, Exposure or OTAs can be the handover point. Depends on computation time and nature of the tier processing. Each NHPPS is invoked as a batch job. We could improve efficiency by launching NHPPS pilot jobs and feeding them OTAs, entire exposures, etc. this avoids queue waits for each OTA.

4.1 The Science Processing Modules

We now turn to the actual content of the demonstration pipeline in this architecture. As noted earlier, there are various orchestration steps which are basically implementation details. The important elements of the pipelines are three IRAF science modules described below.

otafluxcal: This module operates on a single OTA from a single exposure. It accesses the matching master bias and flat field calibration files. It performs the following calibration steps:

- for each cell collapse the overscan to a one dimensional vector, fit a function, evaluate the function for each row and subtract that value from all pixels in the row
- for each cell trim the overscan corrected pixels to just the data region
- for each cell subtract the matching master bias calibration cell
- for each cell divide by the matching master flat field cell
- produce a calibrated MEF version of the raw data calibration cell

While the operations are described separately, they are performed in a single pass through the raw file. An optimized IRAF task from the IRAF ODI package is used within an IRAF host-callable script.

otamerge: This module converts the calibrated MEF format to a simple single image. Each cell raster is added to the output image at the correct location as known from the detector structure. The gaps are filled in with a blank value. This is also an IRAF host-callable script using an optimized IRAF task from the ODI package.

stkstack: This module takes all the OTA images from the dithered exposures and creates the output simple single stacked image. The pixels are combined by appropriate scaling and averaging while taking the offsets and gaps into account. This is also an IRAF host-callable script using a standard IRAF technique with a task from the ODI package.

4.2 The NHPPS Pipelines

The total workflow is structured into three NHPPS pipelines described below.

dit: The dither pipeline takes a list of ODI dither exposures. It sends all the OTAs to parallel instances of the OTA calibration pipeline. It also extracts the references to the calibration data from the headers and includes the appropriate URIs in the data sent to the OTA pipelines. After all the OTAs are calibrated and returned it sends the calibrated OTA data to the dither stacking pipeline. (Note data is not explicitly sent; instead URIs are passed. It is up to the modules and execution frameworks to provide the appropriate access.)

ota: The OTA pipeline processes a single raw OTA file. In the demonstration it applies basic flux calibration and merges the calibrated cells into a single OTA image as described previously.

stk: The dither stacking pipeline takes a list of single image OTA images which have header information defining the dither offsets. The images are put together into a single image of the field with overlapping pixels averaged.

Figure 1 illustrates the workflow as an NHPPS pipeline application using these pipelines. Implicit in this figure is that within the pipelines NHPPS can execute multiple datasets and different modules in parallel (like process level threads) on a node. Also the arrows indicate a one-to-many relation such that the many are distributed across nodes. So in the demonstration pipeline a group of OTAs (say four) can be processed on a multiple core node and the full set of OTAs, broken up in the small groups, can be distributed in parallel across multiple nodes.

4.3 The OGCE Pipeline Services

The OGCE pipeline services wrap all or parts of the NHPPS pipelines. The OGCE pipeline services convert the hierarchical pipeline structure of the NHPPS pipelines into a directed acyclic graph (DAG).

Since NHPPS pipelines are typically connected in a hierarchical fashion this means that pipelines that connect to other pipelines and wait for a return are separated at those points into OGCE services. Typically there are computational stages before and after the break in the NHPPS pipeline is made. This leads to more OGCE services than NHPPS pipelines. On the other hand an implementation may include subpipelines as part of an OGCE service running on a single node.

In the demonstrations the top level NHPPS pipeline, dit, is very simple and is entirely for orchestration. The OGCE implementation of this is able to eliminate the second half of the dit pipeline entirely as described below.

Dither_Pipeline_Extract_OTA_Headers: Execute the first part of the dit pipeline up to calling the ota pipeline.

Instrument_Flux_Calibrations_Remove_Bias: Run one or more instances of the ota pipeline.

NOP: Because in this simple version there are no stages in the last half of the dit pipeline a simplification is made by connecting the output from the ota pipeline directly to the stacking pipeline and have that be the output service.

Stacking_Pipeline_Dither_OTAs: Take the results of the ota_calibrate service and create the dither stack.

Figure 2 shows the OGCE services wrapping the NHPPS pipelines as well as the one that can be elimnated. Figure 3 shows the final OGCE demonstration implementation. The figure is a screen capture of the OGCE composing and visual execution tool. In production this would not be used.

First note that to convert the hierarchical tree-like structure of the NHPPS workflow into a directed acyclic graph, the NHPPS pipelines are split wherever double-ended arrow transitions to other pipelines occur. Within the NHPPS pipelines the parallelism on a single node is still present. The parallelism across many nodes, in this case the Teragrid, is implemented by OGCE components[1] that expand and merge the workflow based on output data from the service to which it is connected.

5 The Demonstration Results

The workflow implemented by the modules, pipelines, and services just described begins with a dither sequence list of raw ODI science exposures, specified by the root name of the multiple OTA MEF files per exposure. The calibration data associated with each is exposure is defined in the

[1] The current demonstration does not use the parallelization components. Instead, the OTA stage internally processes all the pieces in parallel on one node.

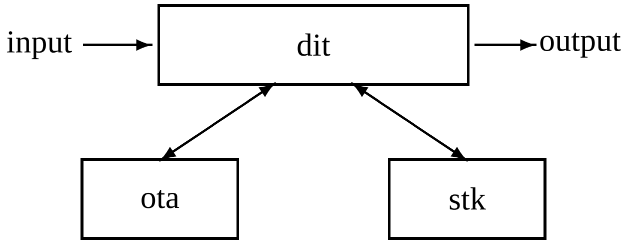

Figure 1: NHPPS pipeline tree. The connections between pipelines are one-to-many which includes one-to-one.

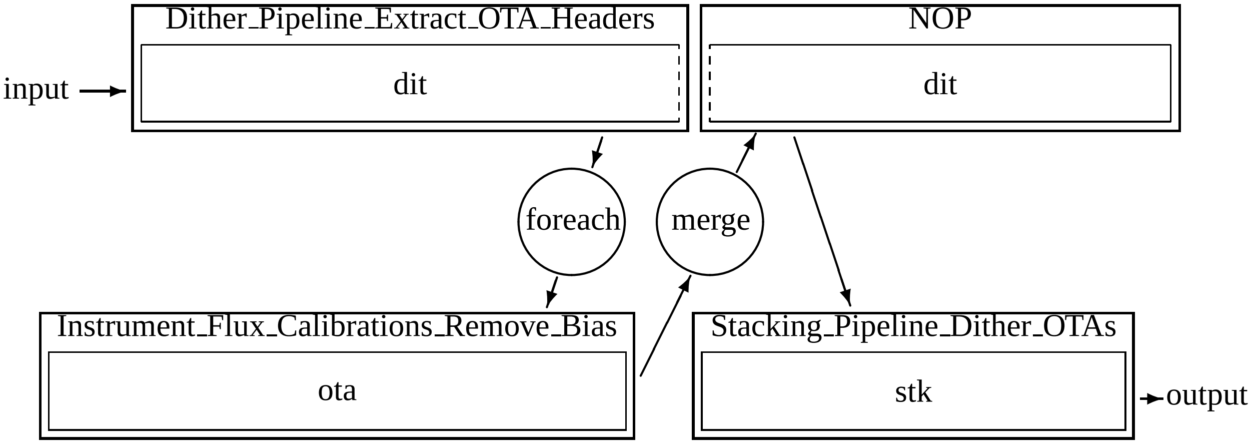
Dither_Pipeline_Extract_OTA_Headers
dit
NOP
dit
input
foreach
merge
Instrument_Flux_Calibrations_Remove_Bias
ota
Stacking_Pipeline_Dither_OTAs
stk
output

Figure 2: OGCE pipeline DAG. The NHPPS dit pipeline is broken in two and the OGCE foreach and merge components provide the distributed parallelization and merging.

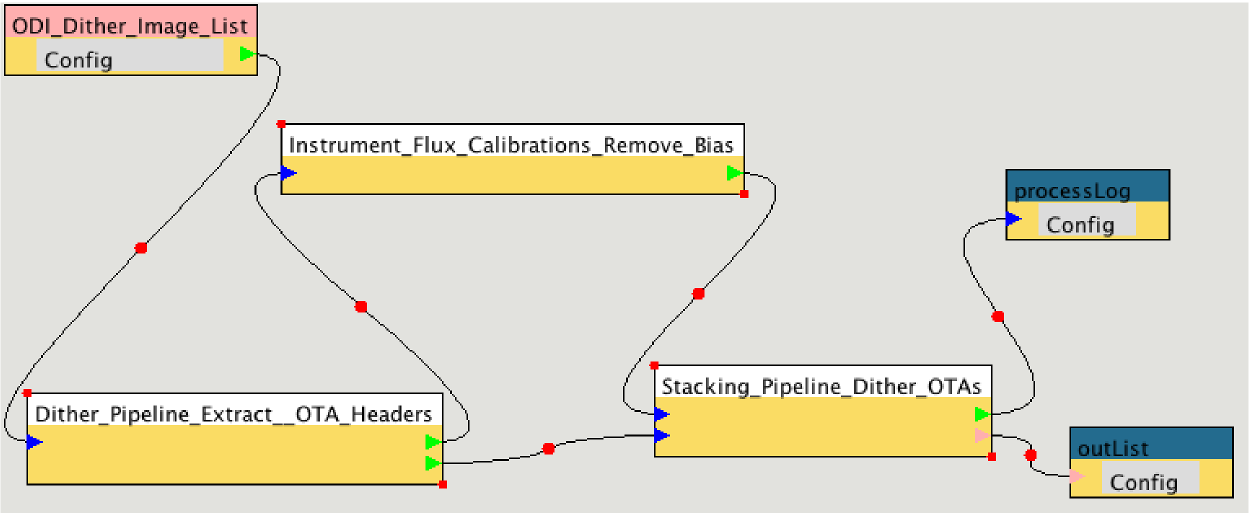
ODI_Dither_Image_List
Config
Instrument_Flux_Calibrations_Remove_Bias
processLog
Config
Dither_Pipeline_Extract__OTA_Headers
Stacking_Pipeline_Dither_OTAs
outList
Config

Figure 3: Graphical representation of wrapped NHPPS pipelines to stack ODI dithered images, orchestrated and execution is monitored in OGCE XBaya.

headers of the science exposures. In a production system this assignment of calibrations would be handled by pipelines and services that interact with a calibration library.

The end goal of the demonstration is to show these representative raw ODI dithered science exposures processed into final science images. There are two types of data products produced by the demonstration pipeline. One is the instrumentally calibrated version of each single exposure. Figure 7 shows one such calibrated exposure.

The other data product is a single ”dither stack” image of all the input exposures with the gaps between cells and OTAs filled in. This data product from the two dithered science exposures is presented in figure 9.

As noted in section 3, it is possible to create data with a full complement of OTAs. Similarly, the demonstration pipeline is capable of processing such data. However, in this paper and in the live demonstration we limit the input and results to a QUOTA scale dataset.

A Figures

Note that the figures (except 4) are at much lower resolution than the data. This means fine detail is lost or minimized and there can be some minor aliasing artifacts. Also while the figures show a single picture, this does not mean that the actual data is necessarily structured as a single image. A display tool is used that tiles the pieces of the data format to produce the visualization shown.

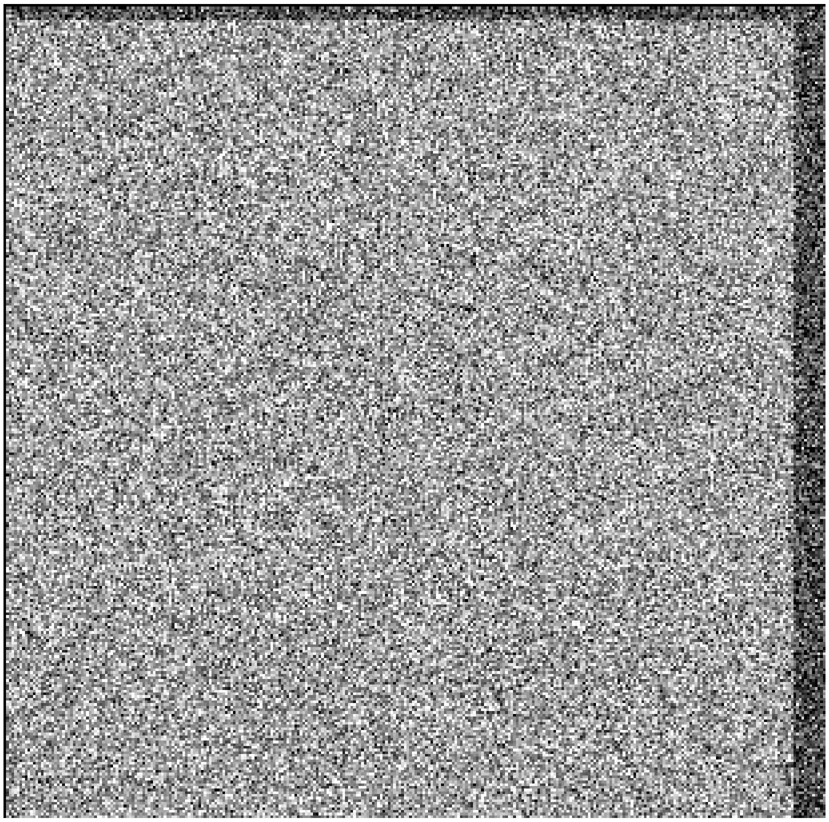

Figure 4: OTA cell (624x608 pixels) showing data and overscan regions.

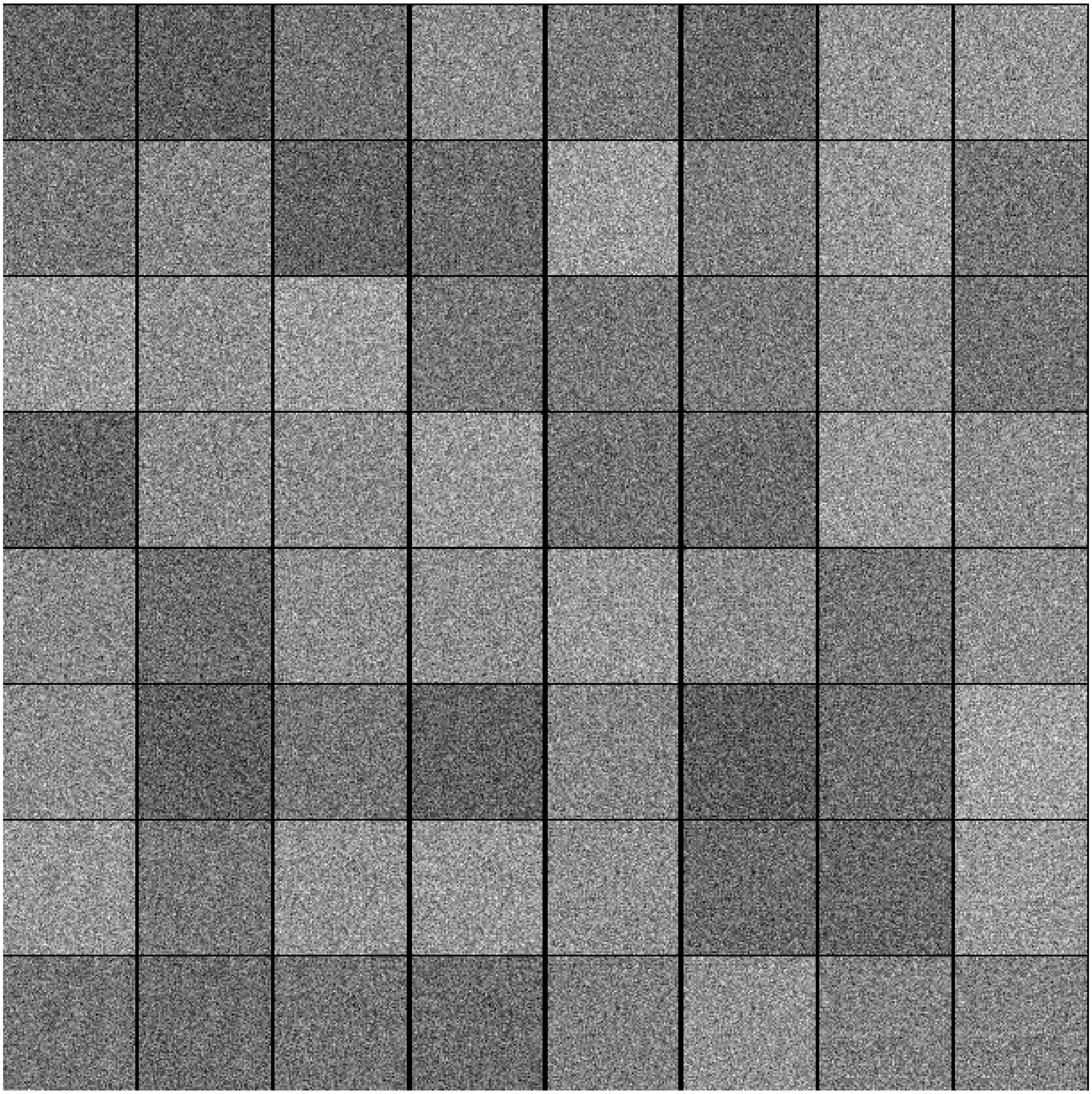

Figure 5: A single OTA (4846x4854 pixels) from a single raw bias exposure. The overscan regions are not shown but the differing bias levels are reflected in the cell levels. A calibrated bias would appear as pure noise around a uniform level near zero.

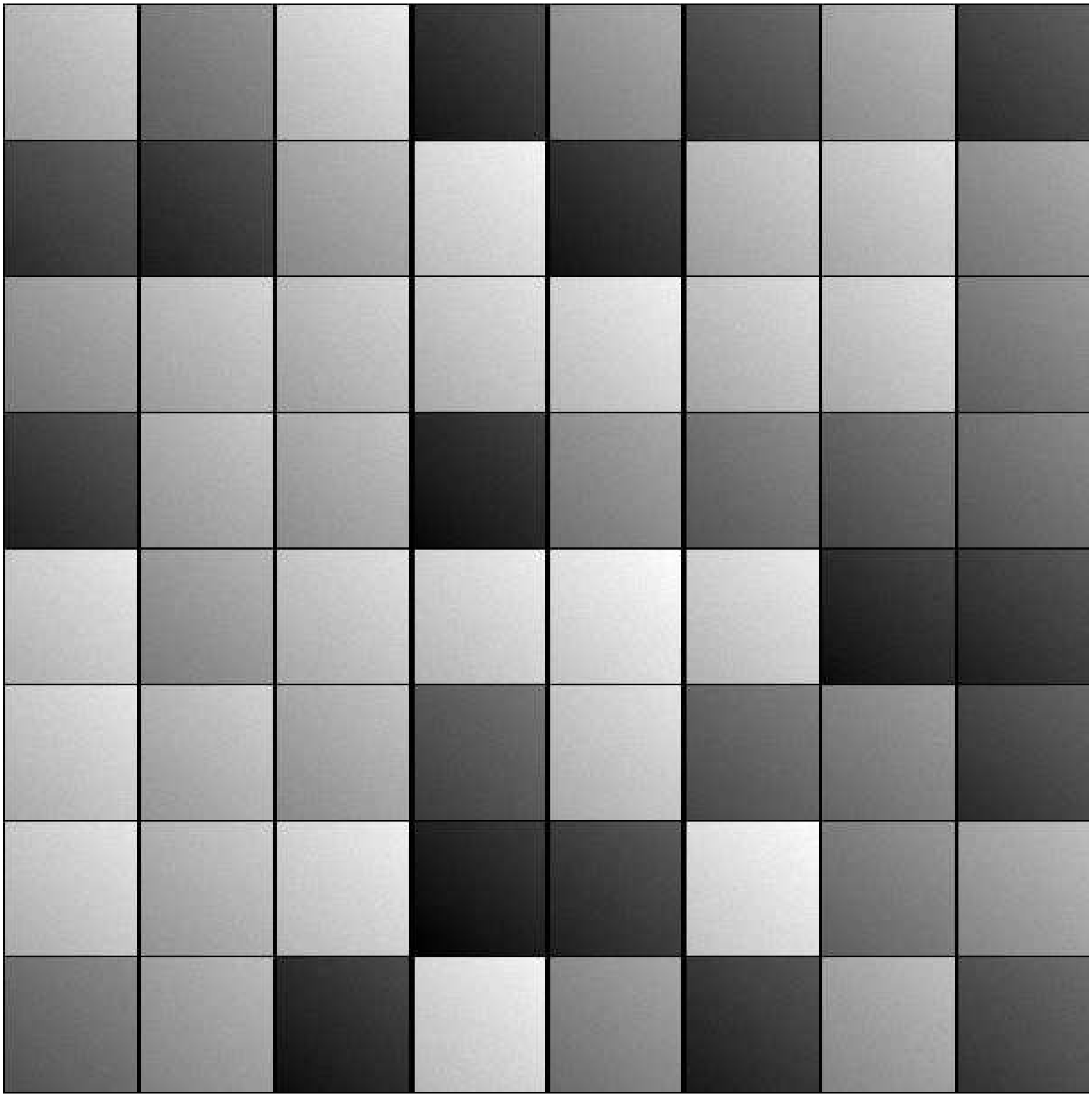

Figure 6: A single OTA (4846x4854 pixels) from a master flat field calibration. The variations between the cells are due to sensitivity differences in the amplifiers for each cell. The amplitude of these differences is such that the grayscale stretch does not show the pixel noise. A raw flat field exposure would not appear different because the overscan and bias levels would also be lost in the grayscale rendering.

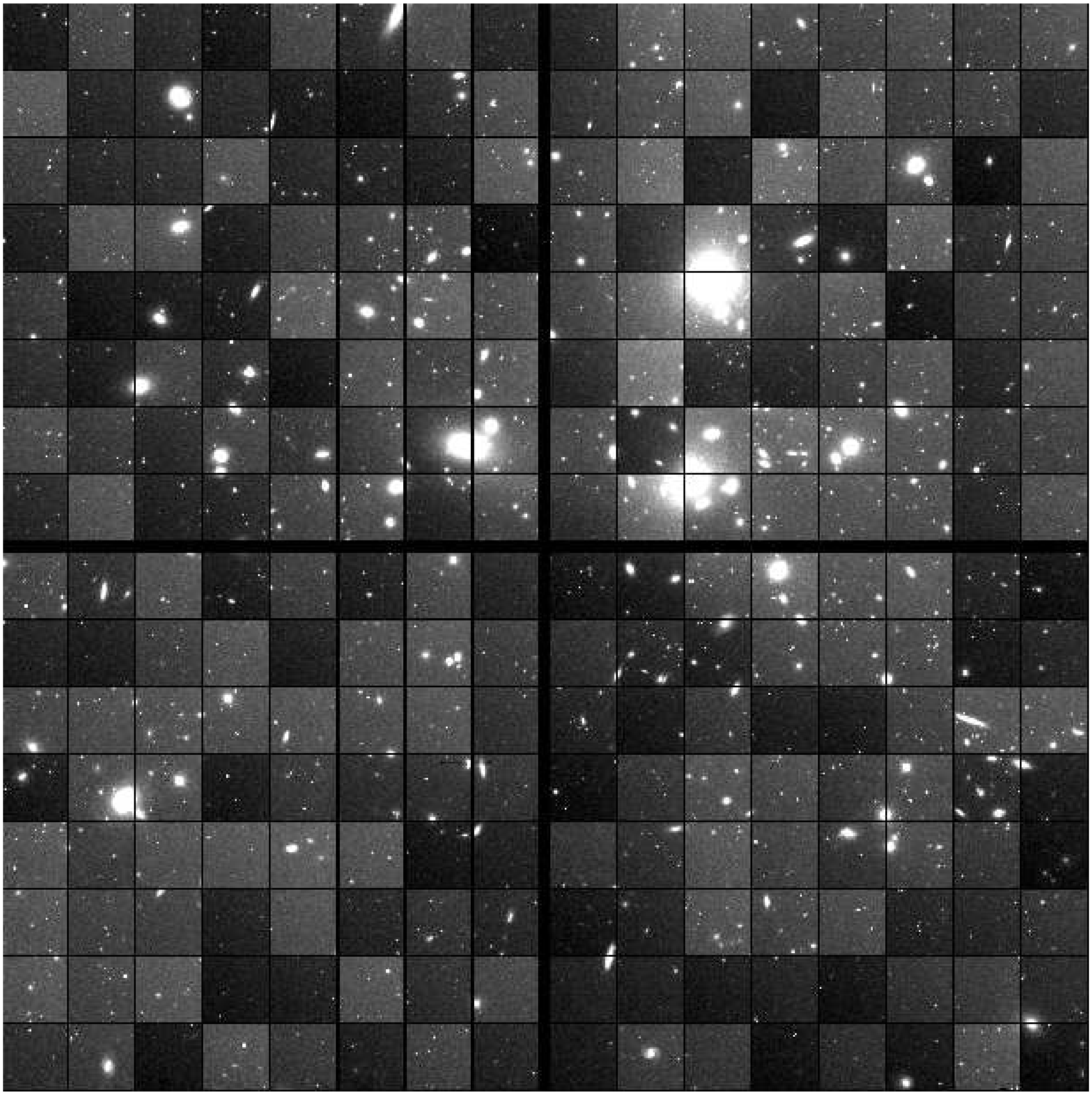

Figure 7: Four OTAs (9692x9708 pixels) from a raw science exposure. The principle variations between cells are due to the sensitivity differences which are calibrated by flat fielding.

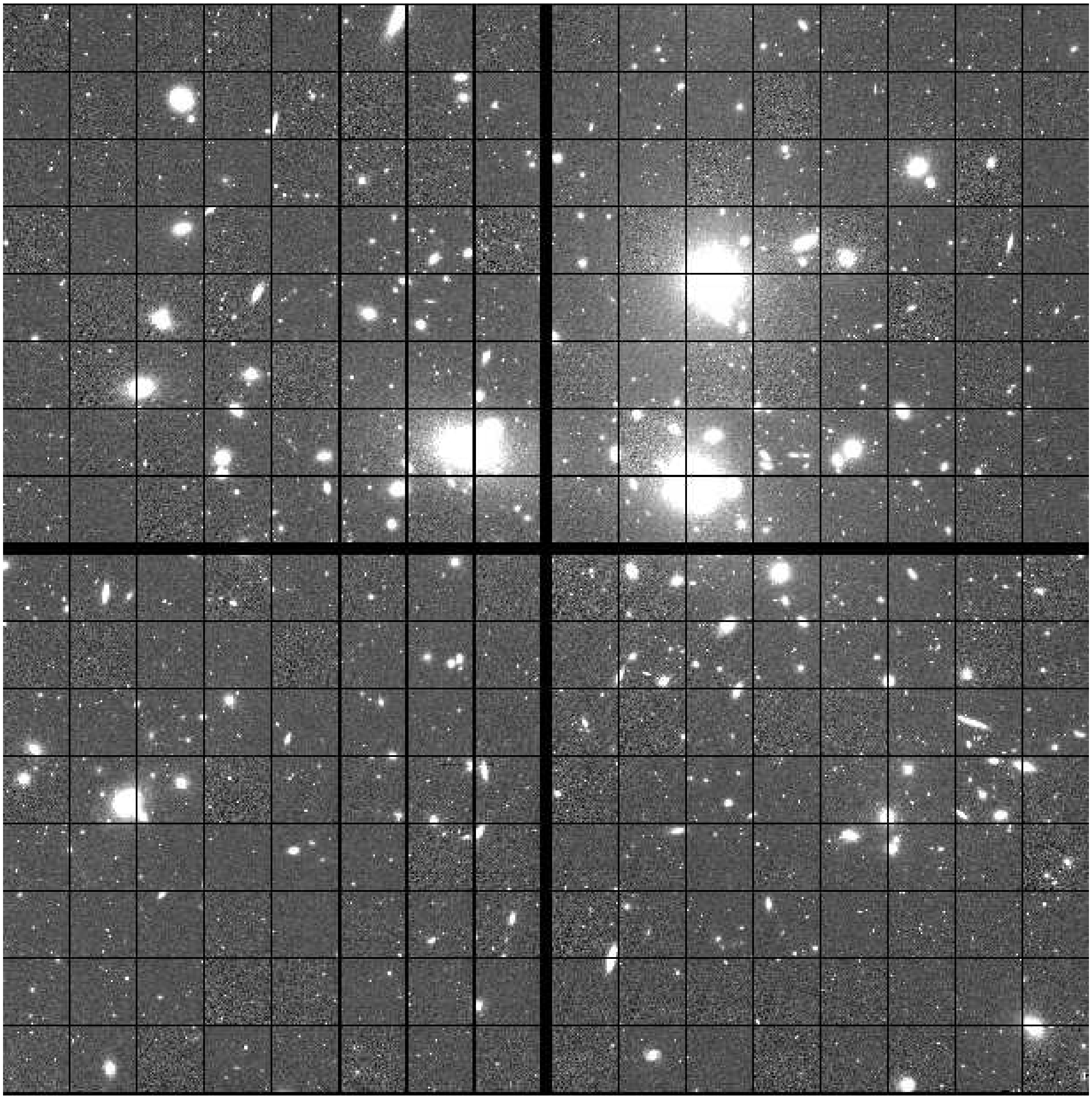

Figure 8: Four OTAs (9692x9708 pixels) from a calibrated science exposure. The calibrations include overscan and zero bias subtraction and flat field normalization.

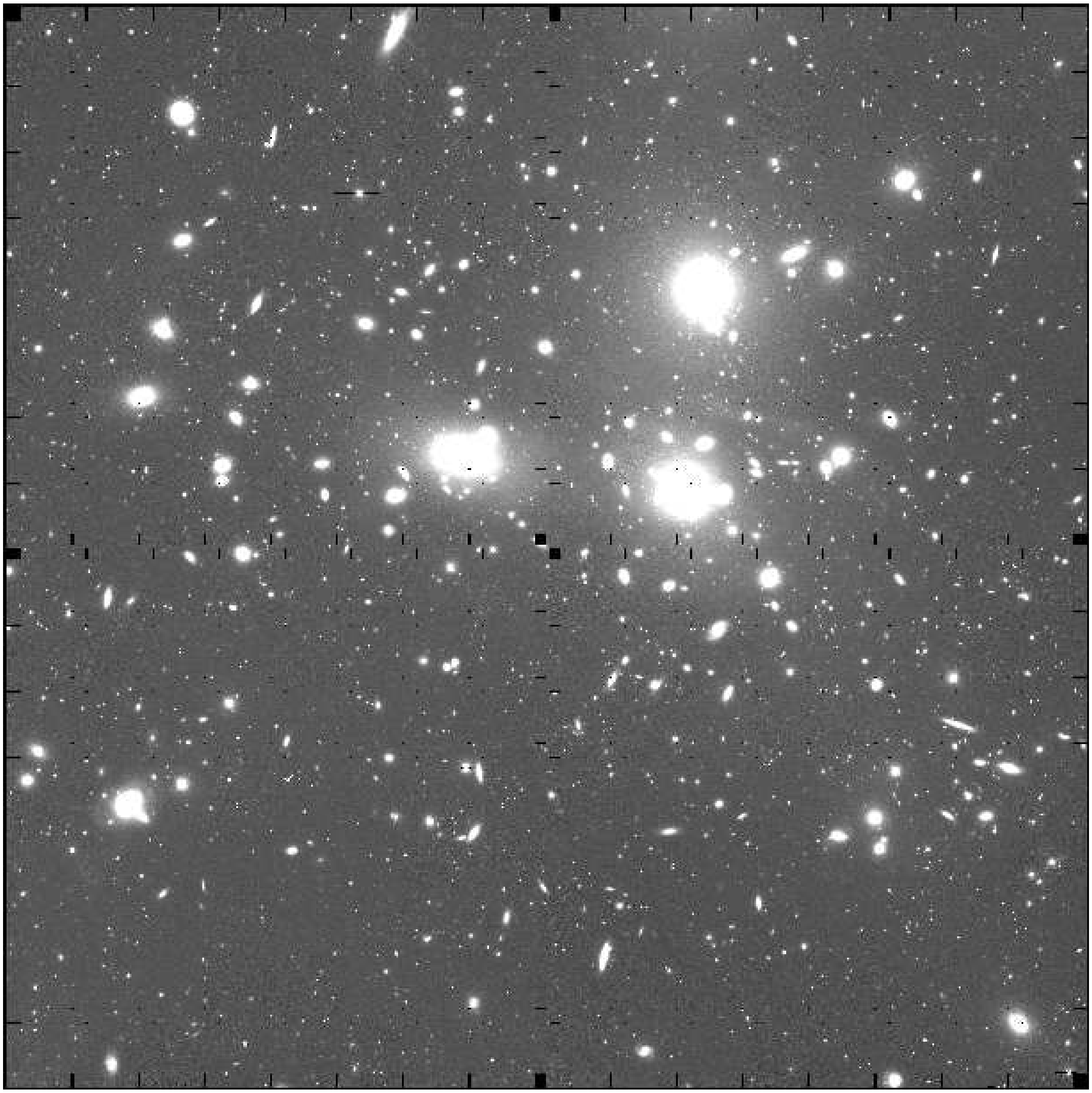

Figure 9: A dither stack (9922x9938 pixels) using just four OTAs from two calibrated science exposures. A full ODI field would be sixteen times bigger. The rectangular dark regions are places where a two exposure dither is insufficient. Dither sets typically have four or more exposures to fill in the entire field except at the edges. The grayscale stretch is the same as in figure 8 for comparison.