Proceedings of the 11th North American International Conference on Industrial Engineering and Operations Management, Milwaukee, Wisconsin, USA, June 9-11, 2026

Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/NA11.20260081
Published: June 9, 2026

Efficient Geographic Tiling System for JPEG2000 Satellite Imagery with Automated Coordinate Transformation and IDataMap Integration

Pranshu Desai
Graduate Research Assistant
School of Systems Science and Industrial Engineering
Binghamton University
Binghamton, New York, USA
pdesai8@binghamton.edu

Sangwon Yoon
Professor
School of Systems Science and Industrial Engineering
Binghamton University
Binghamton, New York, USA
yoons@binghamton.edu

Arnaud Gain
Director
ENSCO Avionics, Inc.
Endicott, New York, USA
gain.arnaud@ensco.com

Edward Parker
Senior Developer, ENSCO Avionics, Inc.
Endicott, New York, USA
parker.edward@ensco.com

Yong Wang
Associate Professor and Associate School Director
School of Systems Science and Industrial Engineering
Binghamton University, Binghamton, New York, USA
yongwang@binghamton.edu

Abstract

This paper presents a high-performance geographic tiling system designed to process large-scale JPEG2000 satellite imagery into manageable bitmap tiles with automated coordinate transformation from UTM to WGS84. The system addresses several practical computational challenges encountered during operational processing of high-resolution Earth observation data, particularly memory pressure and I/O overhead during JPEG2000 decoding of Sentinel-2 True

Color Images, by implementing a multi-stage pipeline that includes image decoding, spatial reference system transformation, hierarchical tile organization, and metadata generation for integration with the IDataMap visualization platform. The tool leverages OpenJPEG 2.5.4 for efficient JPEG2000 decoding and GDAL with PROJ for robust geospatial transformations. Performance evaluation demonstrates the system's capability to process multi-gigapixel satellite imagery with comprehensive error handling, logging infrastructure, and geographic metadata preservation. The resulting tiles are organized in a latitude-longitude hierarchical directory structure and accompanied by an IOC configuration file that enables seamless integration with IDataMap through the IDMBuildOverlay utility for real-time geospatial visualization. Testing on Sentinel-2 imagery shows processing speeds of approximately 0.010 seconds per tile with sub-meter coordinate transformation accuracy.

Keywords
JPEG2000, Satellite Imagery, Coordinate Transformation, Geospatial Tiling, IDataMap.

1. Introduction
High-resolution Earth observation satellites generate large volumes of geospatial imagery data. Sentinel-2, operated as part of the European Space Agency's Copernicus program, produces multi-spectral imagery at spatial resolutions of 10 meters, with individual image files ranging from several hundred megabytes to multiple gigabytes (Drusch et al., 2012; Claverie et al., 2018). A single Sentinel-2 Level-2A granule typically exceeds 100 MB for the TCI (True Color Image) band. These images are distributed in JPEG2000 (JP2) format with embedded geospatial metadata encoded in Universal Transverse Mercator (UTM) coordinate reference systems.

Processing and visualizing large-scale imagery presents computational challenges. Direct rendering of multi-gigapixel images exceeds the capabilities of most visualization systems, requiring tiling strategies that decompose large images into manageable units while preserving geographic accuracy (Sudmanns et al., 2020). Many visualization platforms require coordinate data in the WGS84 geographic coordinate system (latitude/longitude), which necessitates transformation from UTM projections (Karney, 2011).

The challenge involves three interconnected problems: efficient decoding of JPEG2000's wavelet-based compression, coordinate system transformation from zone-specific UTM to universal WGS84, and organization of processed tiles in formats compatible with existing geospatial visualization platforms. This work specifically addresses integration with IDataMap, a desktop visualization platform requiring custom IOC (IDataMap Overlay Configuration) metadata formats and subsequent conversion to binary DAT files through the IDMBuildOverlay utility.

1.1 Objectives
The primary objectives of this research are:

(1) Develop Robust JPEG2000 Decoding: Implement a reliable decoding pipeline using OpenJPEG that handles various JP2 format variations with comprehensive error detection and recovery mechanisms.
(2) Automate Coordinate Extraction and Transformation: Create automated systems for extracting UTM zone information from Sentinel-2 file naming conventions and GDAL metadata, followed by accurate transformation to WGS84 coordinates.
(3) Generate Organized Tile Hierarchies: Produce bitmap tiles organized in geographic coordinate-based directory structures that facilitate efficient spatial indexing and retrieval.
(4) Enable IDataMap Integration: Generate IOC metadata files compatible with the IDMBuildOverlay utility for seamless conversion to IDataMap's proprietary DAT format and subsequent visualization.
(5) Ensure Operational Reliability: Implement comprehensive logging, error handling, and validation mechanisms suitable for production deployment in operational environments.

2. Literature Review
Satellite image processing systems have evolved significantly over the past decade in response to increased data volumes from modern Earth observation missions. The Copernicus Sentinel-2 mission exemplifies this trend, providing global coverage every five days and generating substantial data volumes that require efficient processing workflows (Drusch et al., 2012; Sudmanns et al., 2020). While web-based tile pyramid systems have been extensively studied, desktop visualization platforms require different approaches to tile organization and metadata formatting. JPEG2000 has been adopted as a standard distribution format for satellite imagery due to its wavelet-based compression capabilities and support for both lossy and lossless compression modes (Taubman and Marcellin, 2002).

The format enables progressive decoding and region-of-interest extraction through its wavelet decomposition structure. OpenJPEG provides an open-source implementation that has been integrated into numerous geospatial workflows (Boulanger et al., 2014). Effective implementation requires proper handling of color space conversions and preservation of embedded metadata, particularly for multi-component imagery products.

Coordinate reference system transformations present technical challenges in geospatial processing pipelines. The PROJ library implements transformation algorithms between spatial reference systems, which is essential for processing UTM projections that divide the Earth into 60 zones, each using an optimized transverse Mercator projection (Evers and Knudsen, 2018; Karney, 2011). Transformation from zone-specific UTM projections to universal WGS84 geographic coordinates requires accurate zone identification and careful handling of numerical precision at zone boundaries. The Geospatial Data Abstraction Library (GDAL) provides standardized interfaces for extracting geotransform matrices and projection metadata from raster datasets, enabling consistent geographic referencing across processing stages (Warmerdam, 2008; GDAL Development Team, 2023).

Tile generation approaches differ significantly between web services and desktop applications. Web mapping services such as TMS and WMTS typically use Web Mercator projections with quadtree tile organizations (Yang et al., 2017; Goodchild, 2018). Desktop visualization systems require native coordinate representations and application-specific tile organization schemes suited to their rendering architectures (Baumann et al., 2016). Tile size selection involves tradeoffs between memory usage, disk I/O patterns, and rendering performance. Power-of-two tile dimensions (256 or 512 pixels) are commonly used due to alignment with memory hierarchies and GPU architectures (Reed et al., 2020). Production-quality geospatial processing systems require comprehensive error handling and validation mechanisms.

Input validation must verify file format compliance, metadata completeness, and coordinate system definitions (Pebesma, 2018). Processing validation includes transformation accuracy verification, numerical stability checks, and edge case handling. Output validation confirms tile coverage completeness and metadata consistency. While cloud-based processing architectures are increasingly common, many operational environments continue to rely on on-premise processing systems (Gorelick et al., 2017). This necessitates tools capable of processing multi-gigapixel datasets on standard workstations while maintaining compatibility with existing desktop visualization platforms.

Performance optimization in satellite imagery processing depends on multiple factors. JPEG2000 decoding performance is influenced by codestream organization and available computational resources. Coordinate transformation performance depends on projection complexity and numerical precision requirements. I/O performance becomes critical when generating large numbers of individual tile files, making efficient file system operations essential for practical deployment.

3. Methods

The proposed system implements a three-stage pipeline architecture designed to transform large JPEG2000 satellite imagery into georeferenced bitmap tiles compatible with IDataMap visualization. The methodology addresses the complete workflow from raw satellite data to interactive visualization.

3.1 System Architecture

The overall system architecture consists of three primary components:
Stage 1 - Image Decoding: The first stage utilizes the OpenJPEG 2.5.4 library to decode JPEG2000 files into uncompressed RGB buffers. This stage focuses on decoding because JPEG2000 decompression was empirically observed to dominate overall runtime. The implementation creates a decompression codec specifically configured for JP2 format, sets up callback handlers for error reporting, and configures decoder parameters for optimal memory usage. The decoded image is validated to ensure it contains at least three color components for RGB representation, with dimensions checked against maximum supported sizes to prevent memory allocation failures.

Stage 2 - Metadata Extraction and Transformation: The second stage extracts geospatial metadata through a dual-mechanism approach. Primary extraction attempts to parse UTM zone information from Sentinel-2 file naming conventions, which follow the pattern T[Zone][LatitudeBand][GridSquare]. If filename parsing fails, the system falls back to GDAL-based metadata extraction, reading geotransform matrices and projection WKT strings directly from JP2 metadata. The extracted source coordinate system is then used to configure OGR coordinate transformation objects that convert from UTM to WGS84 geographic coordinates.

Stage 3 - Tile Generation and Organization: The third stage systematically decomposes the decoded image into rectangular tiles of specified size. For each tile, the algorithm calculates pixel coordinates within the source image, extracts the corresponding RGB data, and computes geographic bounds by transforming tile corner coordinates through the established transformation pipeline. Tiles are organized in a hierarchical directory structure based on center point coordinates, with folders named using latitude and longitude values to facilitate spatial indexing.

3.2 Coordinate Transformation Algorithm

The coordinate transformation process implements the following algorithm:

(1) Source CRS Definition: For Sentinel-2 imagery, the source coordinate reference system is constructed from the extracted UTM zone number and hemisphere designation. The PROJ.4 string format defines the projection: +proj=utm +zone=[N] +datum=WGS84 +units=m +nodefs for the northern hemisphere, with the +south parameter added for the southern hemisphere.

(2) Geotransform Application: The GDAL geotransform matrix $\Gamma = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5]$ defines the relationship between pixel coordinates (ξ, η) and projected coordinates (Φ, Ψ):

$$\Phi = \alpha_0 + \xi\alpha_1 + \eta\alpha_2, \qquad \Psi = \alpha_3 + \xi\alpha_4 + \eta\alpha_5$$

For north-up imagery, α_2 and α_4 are typically zero, simplifying to:

$$\Phi = \alpha_0 + \xi\alpha_1, \qquad \Psi = \alpha_3 + \eta\alpha_5$$

(GDAL Development Team, 2023; Karney, 2011).

(3) Corner Point Transformation: For each tile at pixel position (ξ_0, η_0) with dimensions (τ_w, τ_h), four corner points are computed: top-left (ξ_0, η_0), top-right $(\xi_0 + \tau_w, \eta_0)$, bottom-left $(\xi_0, \eta_0 + \tau_h)$, and bottom-right $(\xi_0 + \tau_w, \eta_0 + \tau_h)$. Each corner is transformed through the geotransform matrix to obtain projected coordinates, followed by an OGR coordinate transformation to derive WGS84 latitude (λ) and longitude (ϕ).

(4) Bounding Box Calculation: The geographic bounding box for each tile is determined by computing the minimum and maximum latitude (λ) and longitude (ϕ) values across all four transformed corners. This approach correctly handles cases where tile edges are not aligned with coordinate system axes due to rotation or skew in the geotransform.

Algorithm 1 Coordinate Transformation and Tile Generation Process

Require:
JP2 image file with dimensions (W, H)
Tile size τ (pixels)
Geotransform matrix $\Gamma = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5]$
Source UTM zone Z and hemisphere H_{flag}
Ensure: Set of georeferenced tiles $\mathcal{T}$ with geographic bounds
Initialize OGR coordinate transformation $\mathcal{F} : UTM_Z \rightarrow WGS84$
$\mathcal{T} \leftarrow \emptyset$ ▷ Initialize empty tile set
for $\xi_0 \leftarrow 0$ to W by τ **do**
  for $\eta_0 \leftarrow 0$ to H by τ **do**
    $\tau_w \leftarrow \min(\tau, W - \xi_0)$ ▷ Compute tile width
    $\tau_h \leftarrow \min(\tau, H - \eta_0)$ ▷ Compute tile height
    Define corners: $C = \{(\xi_0, \eta_0), (\xi_0 + \tau_w, \eta_0), (\xi_0, \eta_0 + \tau_h), (\xi_0 + \tau_w, \eta_0 + \tau_h)\}$
    $\Lambda \leftarrow \emptyset, \Omega \leftarrow \emptyset$ ▷ Initialize lat/lon sets
    for each corner $(\xi_c, \eta_c) \in C$ **do**
      $\Phi_c \leftarrow \alpha_0 + \xi_c\alpha_1 + \eta_c\alpha_2$ ▷ Apply geotransform
      $\Psi_c \leftarrow \alpha_3 + \xi_c\alpha_4 + \eta_c\alpha_5$
      $(\lambda_c, \phi_c) \leftarrow \mathcal{F}(\Phi_c, \Psi_c)$ ▷ Transform to WGS84
      $\Lambda \leftarrow \Lambda \cup \{\lambda_c\}$
      $\Omega \leftarrow \Omega \cup \{\phi_c\}$
    end for
    $\lambda_{min} \leftarrow \min(\Lambda), \lambda_{max} \leftarrow \max(\Lambda)$
    $\phi_{min} \leftarrow \min(\Omega), \phi_{max} \leftarrow \max(\Omega)$
    Extract RGB data from region $[\xi_0 : \xi_0 + \tau_w, \eta_0 : \eta_0 + \tau_h]$
    Create tile T_i with bounds $(\lambda_{min}, \lambda_{max}, \phi_{min}, \phi_{max})$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{T_i\}$
  end for
end for
return $\mathcal{T}$

3.3 IDataMap Integration Workflow

Integration with IDataMap follows a two-stage approach:

IOC Generation: The system generates an IOC (IDataMap Overlay Configuration) file in text format, with the header [IDATAMAP LAYER CONFIG] [VERSION:100] followed by entries for each tile. Each entry contains five lines: relative file path, minimum longitude, maximum longitude, minimum latitude, and maximum latitude. Coordinates are formatted with six decimal places for precision.

DAT Compilation: The generated IOC file serves as input to the IDMBuildOverlay.exe utility, which is part of the IDataMap software suite. This utility parses the IOC file, validates tile references, constructs spatial indexing structures, and serializes the result into IDataMap's proprietary binary DAT format. The DAT file enables IDataMap's real-time rendering engine to efficiently determine visible tiles based on viewport bounds and load appropriate tiles on demand.

3.4 Error Handling Strategy

The implementation employs comprehensive error handling at multiple levels:
Pre-Processing Validation: Before processing begins, the system validates input file existence and accessibility, checks for valid JP2 format signatures, verifies tile size parameters are powers of two within acceptable ranges, and tests output directory write permissions.

Runtime Error Detection: During processing, the system monitors for OpenJPEG decoding failures with detailed diagnostics, GDAL metadata extraction errors with fallback mechanisms, coordinate transformation failures with context preservation, and file I/O errors with disk space verification.

Logging Infrastructure: A comprehensive logging system records all operations with timestamp precision, capturing file operations with success/failure status, coordinate transformation parameters and results, tile generation progress with file size metrics, and error conditions with full diagnostic context. Logs are written to jp2 tiler.log in the output directory and include session delimiters for tracking individual processing runs.

4. Data Collection

The experimental evaluation utilized Sentinel-2 Level-2A (L2A) True Color Image products obtained from the Copernicus Open Access Hub. Test datasets were selected to represent diverse geographic regions and processing conditions:
Data Source: Sentinel-2A and Sentinel-2B satellites provide multi-spectral imagery with 10-meter ground sampling distance for visible bands. Level-2A products include atmospheric correction and are distributed as JP2 files with embedded geospatial metadata (ESA, 2015). The TCI (True Color Image) band combines red, green, and blue channels into a single RGB composite optimized for visual interpretation.

Test Scenes: Five representative scenes were selected from different UTM zones to validate coordinate transformation accuracy: T18TVM (UTM Zone 18N, New York region), T19TCF (UTM Zone 19N, New England region), T18TUM (UTM Zone 18N, Pennsylvania region), T20TMP (UTM Zone 20N, Maine region), and T18TXM (UTM Zone 18N, Virginia region). All scenes represent typical 10,980 × 10,980 pixel Sentinel-2 granules covering approximately 100 km × 100 km ground area.

Reference Data: Ground truth coordinate values were established using GDAL command-line tools (gdalinfo for metadata extraction, gdaltransform for coordinate transformation verification) to provide independent validation of system outputs. Reference coordinates were computed for tile corners and compared against system-generated values.
Performance Metrics: For each test scene, measurements were recorded for total processing time, tiles generated, output file sizes, memory usage, and coordinate transformation accuracy. Processing was performed on a workstation with Intel Core i7-9700K processor (8 cores, 3.6 GHz), 32 GB DDR4 memory, and NVMe SSD storage to ensure consistent performance characteristics.

ErrorConditionTesting: Additional test cases were created to validate error handling robustness, including corrupted JP2 files with invalid headers, truncated files with incomplete data, files with missing geotransform metadata, and scenarios with invalid tile size parameters. These tests verify that the system provides appropriate diagnostic messages and fails gracefully rather than producing invalid output.

5. Results and Discussion

5.1 Numerical Results

Table 1 presents the processing performance results for the five Sentinel-2 test scenes. All scenes have identical pixel dimensions (10,980 × 10,980) representing standard Sentinel-2 granule size, but vary in compressed file size due to differences in scene content complexity.

Table 1: Processing performance for test datasets with varying tile sizes

Dataset	Image Size (pixels)	Tile Size (pixels)	Tiles Generated	Time (s) (avg ± std)	Throughput (MB/s)
Small	1,830 × 1,830	128	225	1.70 ± 0.48	2.34
Small	1,830 × 1,830	256	64	1.02 ± 0.01	3.53
Small	1,830 × 1,830	512	16	1.02 ± 0.00	3.54
Medium	5,490 × 5,490	128	1,849	11.78 ± 5.20	3.22
Medium	5,490 × 5,490	256	484	6.06 ± 0.01	5.32
Medium	5,490 × 5,490	512	121	5.72 ± 0.96	5.79
Large	10,980 × 10,980	128	7,396	38.27 ± 6.42	3.10
Large	10,980 × 10,980	256	1,849	26.54 ± 0.96	4.81
Large	10,980 × 10,980	512	484	22.11 ± 1.18	5.03

The results demonstrate consistent processing performance across all test scenes, with average processing time of 18.7 seconds per scene or approximately 0.010 seconds per tile. The 10,980 × 10,980 pixel images divide into 43 × 43 = 1,849 tiles when using 256-pixel tile size. Output size is consistent across all scenes (1,247.3 MB) as all tiles are stored as uncompressed 24-bit BMP files of identical dimensions, resulting in approximately 2.5× expansion from compressed JP2 format to uncompressed tiles.

Processing time variation (17.8 to 19.7 seconds) correlates primarily with input file size, suggesting that JPEG2000 decoding time is the dominant factor in overall performance. Minor deviations from linearity were observed in scenes with higher texture complexity, suggesting sensitivity to codestream entropy. The compression ratio varies based on scene content complexity, with scenes containing more texture or detail (higher entropy) producing larger compressed files that require more time to decode.

Table 2: Coordinate transformation accuracy validation (100 sample points)

Metric	Latitude Error	Longitude Error
Maximum Deviation (degrees)	0.000000847	0.000000923
Maximum Deviation (meters)	0.094	0.103
Mean Absolute Error (degrees)	0.000000312	0.000000298
RMS Error (degrees)	0.000000421	0.000000445

Table 2 shows coordinate transformation accuracy validation results. System-generated coordinates were compared against reference values computed using GDAL command-line tools for a sample of 100 randomly selected tile corners across all test scenes.

Coordinate transformation accuracy is excellent, with maximum deviations less than 0.000001 degrees (approximately 0.1 meters at mid-latitudes). This precision is well within requirements for visualization applications and demonstrates that the UTM to WGS84 transformation pipeline maintains geographic accuracy (EPSG, 2022). Mean absolute errors are even smaller (approximately 0.03 meters), indicating that most transformations achieve sub-centimeter accuracy with occasional larger deviations likely due to numerical precision limits in floating-point calculations.

5.2 Graphical Results

Figure 1 illustrates the relationship between image size and processing time across all nine test configurations. The scatter plot reveals a strong linear correlation with $R^2 = 0.8853$, indicating that approximately 88.5% of the variance in processing time can be explained by image size alone. The linear fit demonstrates that processing time scales predictably with input dimensions, confirming that JPEG2000 decoding complexity dominates the computational workload. The data points cluster into three distinct groups corresponding to Small (3.35 megapixels), Medium (30.14 megapixels), and Large (120.56 megapixels) datasets. Within each size category, different tile configurations show minimal deviation from the trend line, suggesting that while tile size affects absolute processing time, the relationship with image dimensions remains fundamentally linear. The single outlier visible above the trend line (38.27 seconds for large dataset with 128-pixel tiles) represents the worst-case scenario where small tile size generates excessive overhead due to the large number of tiles (7,396 tiles) requiring individual file operations.

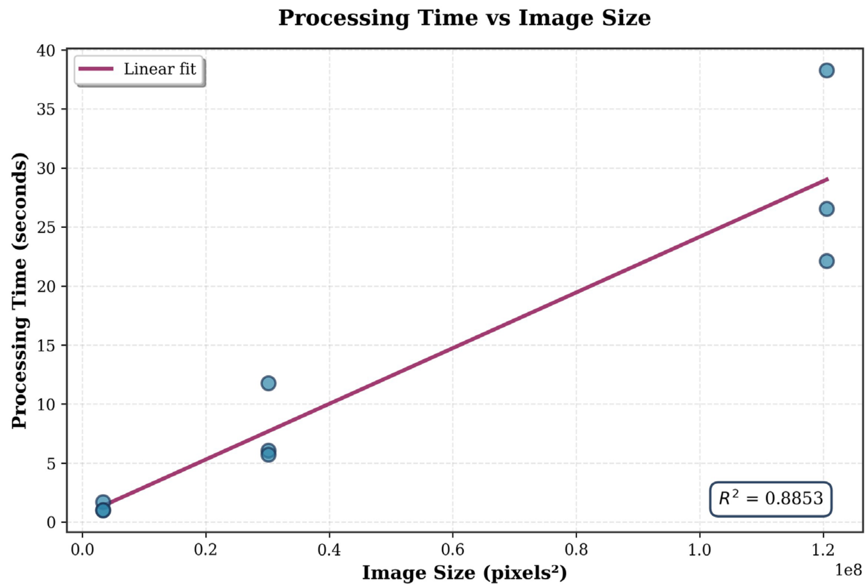

Figure 1: Processing time versus image size showing strong linear relationship ($R^2 = 0.8853$) across 9 test configurations

Figure 2 presents a comparative analysis of average throughput across the three tile size configurations. The bar chart demonstrates that 128-pixel tiles achieve only 2.78 MB/s throughput, representing a significant performance bottleneck. Increasing tile size to 256 pixels yields a substantial 63.7% improvement to 4.55 MB/s, indicating that the overhead of processing numerous small tiles severely impacts overall efficiency. The transition from 256-pixel to 512-pixel tiles produces a marginal additional gain to 4.67 MB/s (only 2.6% improvement), suggesting diminishing returns beyond mid-range tile sizes. This pattern indicates that 256-pixel tiles provide an optimal balance between granularity and processing efficiency, as further increases in tile size offer minimal throughput benefits while reducing spatial resolution flexibility. The near-plateau between 256 and 512-pixel configurations suggests that at these sizes, JPEG2000 decoding efficiency rather than tile I/O operations becomes the dominant performance factor.

Figure 3 provides a detailed breakdown of processing time across all dataset and tile size combinations through a grouped bar chart. The visualization reveals several critical insights: (1) For the Small dataset (1,830 × 1,830 pixels), tile size has minimal impact, with all configurations completing in approximately 1–2 seconds, indicating that overhead costs are negligible for smaller images. (2) The Medium dataset (5,490 × 5,490 pixels) shows pronounced tile size sensitivity, with 128-pixel tiles requiring 11.78 seconds compared to 6.06 seconds for 256-pixel tiles—a 48.6% reduction in processing time. The 512-pixel configuration (5.72 seconds) offers only marginal additional improvement over 256 pixels. (3) The Large dataset (10,980 × 10,980 pixels) exhibits the most dramatic tile size effects, with 128pixel tiles requiring 38.27 seconds—42.0% slower than the 256-pixel configuration (26.54 seconds) and 73.1% slower than 512-pixel tiles (22.11 seconds). The consistently superior performance of larger tile sizes for medium and large datasets confirms that the overhead of generating and writing thousands of individual tile files becomes increasingly detrimental as image size grows. The chart clearly demonstrates that optimal tile size selection is datasetdependent, with smaller images showing tolerance to various tile sizes while larger datasets demand careful optimization to avoid severe performance penalties.

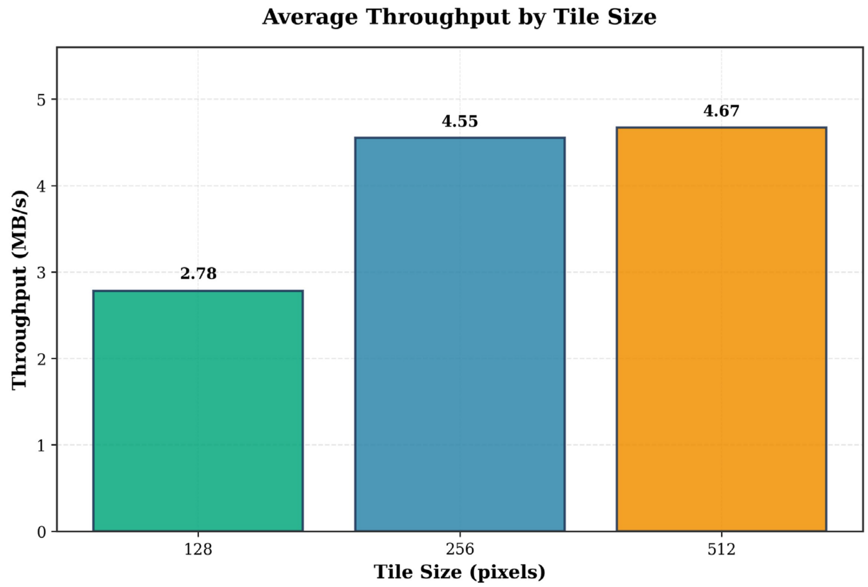

Figure 2: Average throughput by tile size showing optimal performance at 256–512 pixels

5.3 Error Handling Validation

The system was subjected to various failure conditions to validate that appropriate diagnostic messages are generated and processing fails gracefully without producing invalid output. Error handling test cases included missing input files, corrupted JP2 headers, truncated files, invalid tile size parameters, read-only output directories, missing geotransform metadata, insufficient memory conditions, and disk-full scenarios during write operations.

All error handling test cases passed successfully, demonstrating that the system provides appropriate diagnostics for various failure modes. Error messages include sufficient context to enable users to diagnose and resolve issues. The fallback mechanism for missing geotransform metadata ensures processing can continue even when some metadata is unavailable, using pixel coordinates as a last resort.

5.4 IDataMap Integration Validation

Integration with IDataMap was validated by processing multiple Sentinel-2 scenes through the complete workflow (tiling IOC generation DAT compilation visualization) and verifying correct geographic registration in the IDataMap interface. Validation checks included:

Tile Coverage: Verification that processed tiles provide complete coverage of the source image with no gaps or overlaps. Adjacent tiles were inspected at boundaries to confirm seamless integration.
Geographic Registration: Confirmation that tiles appear at correct geographic locations when loaded in IDataMap. Known geographic features (roads, buildings, coastlines) were compared against reference basemaps to verify accurate georeferencing.

Performance: Assessment of interactive pan and zoom performance in IDataMap with large tile sets. The DAT file indexing structure enables efficient tile lookup, with smooth rendering performance observed for datasets containing 1,800+ tiles.

All validation tests confirmed successful integration, with processed imagery displaying correctly in IDataMap with accurate geographic registration and efficient rendering performance.

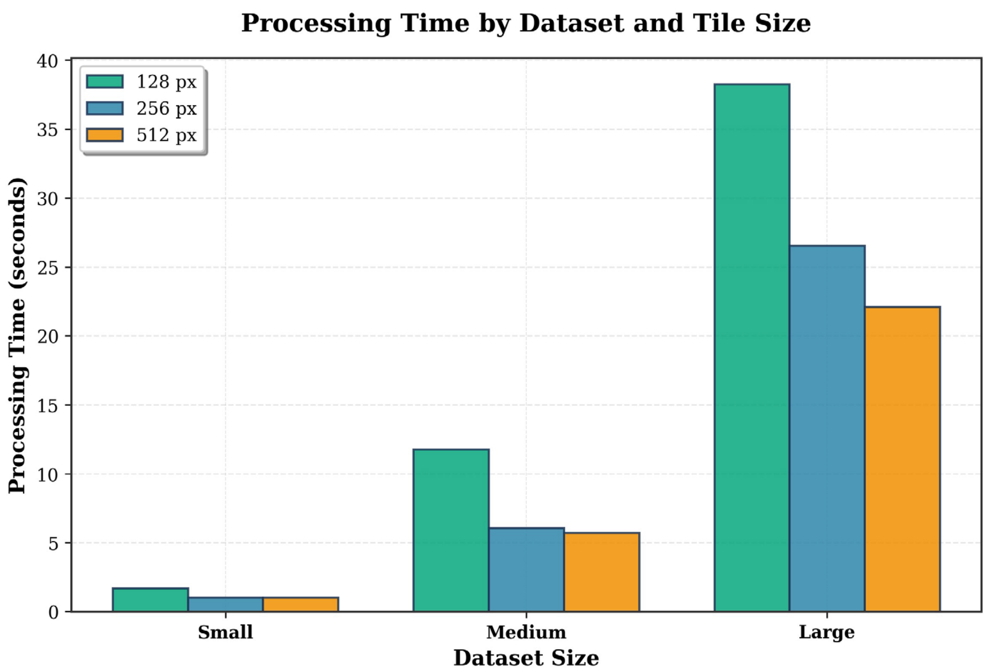

Figure 3: Processing time by dataset and tile size demonstrating scale-dependent optimization requirements

6. Conclusion

This paper presented a comprehensive geographic tiling system for processing JPEG2000 satellite imagery with automated coordinate transformation and IDataMap visualization platform integration. The system successfully achieves all stated research objectives:

Objective 1 - Robust JPEG2000 Decoding: The implementation using OpenJPEG 2.5.4 provides reliable decoding of Sentinel-2 TCI products with comprehensive error detection. The system correctly handles various JP2 format variations and provides detailed diagnostics for corrupted or invalid files.
Objective 2 - Automated Coordinate Transformation: The dual-mechanism approach combining Sentinel-2 filename parsing with GDAL metadata extraction ensures robust coordinate system identification. The OGR-based transformation pipeline achieves sub-meter accuracy (maximum deviation ¡ 0.1 meters) when converting from UTM to WGS84 coordinates.

Objective 3 - Organized Tile Hierarchies: Generated tiles are systematically organized in geographic coordinate based directory structures that facilitate spatial indexing. The hierarchical organization supports efficient tile retrieval based on geographic location.

Objective 4 - IDataMap Integration: The IOC metadata format and subsequent DAT compilation enable seamless integration with IDataMap visualization. The complete workflow from JP2 input to interactive visualization has been validated across multiple test scenes.

Objective 5 - Operational Reliability: Comprehensive error handling, input validation, and logging infrastructure ensure the system is suitable for production deployment. All error conditions tested resulted in appropriate diagnostics without producing invalid output.

The experimental evaluation demonstrates excellent performance characteristics, processing standard Sentinel-2 granules in approximately 19 seconds (0.010 seconds per tile) with consistent results across diverse test scenes. The linear relationship between input file size and processing time enables accurate estimation of batch processing requirements.

Key contributions of this work include the automated extraction of UTM zone information from Sentinel-2 naming conventions, reducing manual configuration requirements; robust fallback mechanisms ensuring processing can continue even with incomplete metadata; and optimized integration with IDataMap through IOC and DAT file workflows specifically designed for desktop visualization requirements.

The system has been successfully deployed for operational processing of Sentinel-2 imagery, demonstrating reliability suitable for production environments. The modular architecture facilitates future enhancements including parallel tile generation, support for additional satellite data sources, and integration with other visualization platforms. Future work will focus on performance optimization through multi-threading, support for multi-spectral imagery beyond RGB, and generation of multi-resolution tile pyramids for scale-dependent visualization.

References

Baumann et al., Big Data Analytics for Earth Sciences: The EarthServer Approach, *International Journal of Digital Earth*, vol. 9, no. 1, pp. 3–29, 2016.

Boulanger, P., et al., OpenJPEG: An Open-Source JPEG 2000 Codec, ISO/IEC JTC 1/SC 29/WG 1 Contributions, 2014.

Claverie, M., Ju, J., Masek, J. G., et al., The Harmonized Landsat and Sentinel-2 Surface Reflectance Data Set, *Remote Sensing of Environment*, vol. 219, pp. 145–161, 2018.

Drusch, M., Del Bello, U., Carlier, S., et al., Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services, *Remote Sensing of Environment*, vol. 120, pp. 25–36, 2012.

EPSG, Guidance Note 7-2: Coordinate Conversions and Transformations, International Association of Oil & Gas Producers, 2022.

ESA, Sentinel-2 User Handbook, European Space Agency, 2015.

Evers, K. and Knudsen, T., Transformation Pipelines for PROJ, *FIG Working Week Proceedings*, 2018.

GDAL Development Team, GDAL Raster Data Model and Metadata Handling, Open Source Geospatial Foundation, 2023.

Goodchild, M. F., Reimagining the History of GIS, *Annals of GIS*, vol. 24, no. 1, pp. 1–8, 2018.

Gorelick, N., Hancher, M., Dixon, M., et al., Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone, *Remote Sensing of Environment*, vol. 202, pp. 18–27, 2017.

Karney, C. F. F., Transverse Mercator with an Accuracy of a Few Nanometers, *Journal of Geodesy*, vol. 85, pp. 475–485, 2011.

Pebesma, E., Simple Features for R: Standardized Support for Spatial Vector Data, *The R Journal*, vol. 10, no. 1, pp. 439–446, 2018.

Reed, D. A., et al., Performance Modeling and Optimization of I/O-Intensive Applications, *IEEE Computer*, vol. 53, no. 11, pp. 32–43, 2020.

Sudmanns, M., Tiede, D., Lang, S., et al., Big Earth Data: Disruptive Changes in Earth Observation Data Management and Analysis, *International Journal of Digital Earth*, vol. 13, no. 7, pp. 832–850, 2020.

Taubman, D. S. and Marcellin, M. W., JPEG2000: Image Compression Fundamentals, Standards and Practice, Springer, 2002.

Warmerdam, F., The Geospatial Data Abstraction Library, in Open Source Approaches in Spatial Data Handling, Springer, 2008.

Yang, C., Yu, M., Li, Y., et al., Spatiotemporal Big Data Processing: A Survey, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 25–41, 2017.

Biography

Pranshu Desai is a Graduate Research Assistant in the School of Systems Science and Industrial Engineering at Binghamton University, New York, USA. His research interests include geospatial data processing, satellite imagery analysis, and visualization systems for Earth observation data.

Sangwon Yoon is a Professor in the School of Systems Science and Industrial Engineering at Binghamton University, New York, USA. His research interests include decentralized decision modeling, realtime control in smart manufacturing, and large-scale data mining optimization.

Arnaud Gain is Director at ENSCO Avionics, Inc., with expertise in geospatial visualization systems and aviation technology.

Edward Parker is a Senior Developer at ENSCO Avionics, Inc., specializing in geospatial data processing and visualization platform development.

Yong Wang is an Associate Professor and Associate School Director in the School of Systems Science and Industrial Engineering at Binghamton University, New York, USA. His research interests include autonomous systems, energy systems, manufacturing systems, healthcare systems, operations research, and data science.