LA-UR-12-24049

Approved for public release; distribution is unlimited.

| | |
|---|---|
| Title: | Wavelet-Smoothed Interpolation of Masked Scientific Data for JPEG 2000 Compression |
| Author(s): | Brislawn, Christopher M. |
| Intended for: | Project report |

# Wavelet-Smoothed Interpolation of Masked Scientific Data for JPEG 2000 Compression

Prepared for: DOE Office of Science ASCR

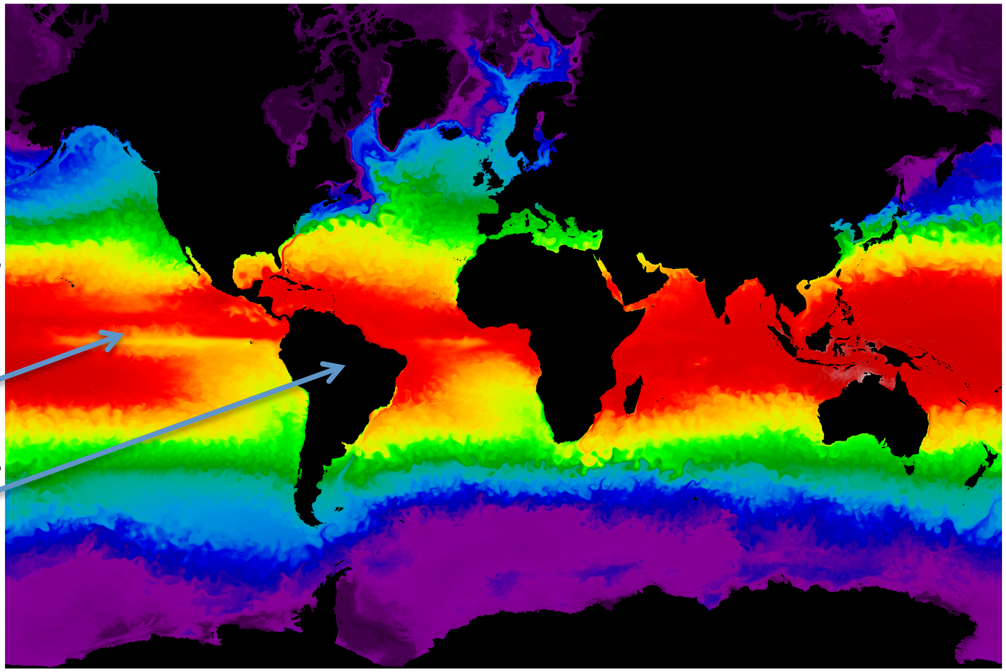
By: Christopher M. Brislawn
(brislawn@lanl.gov)

August 2012

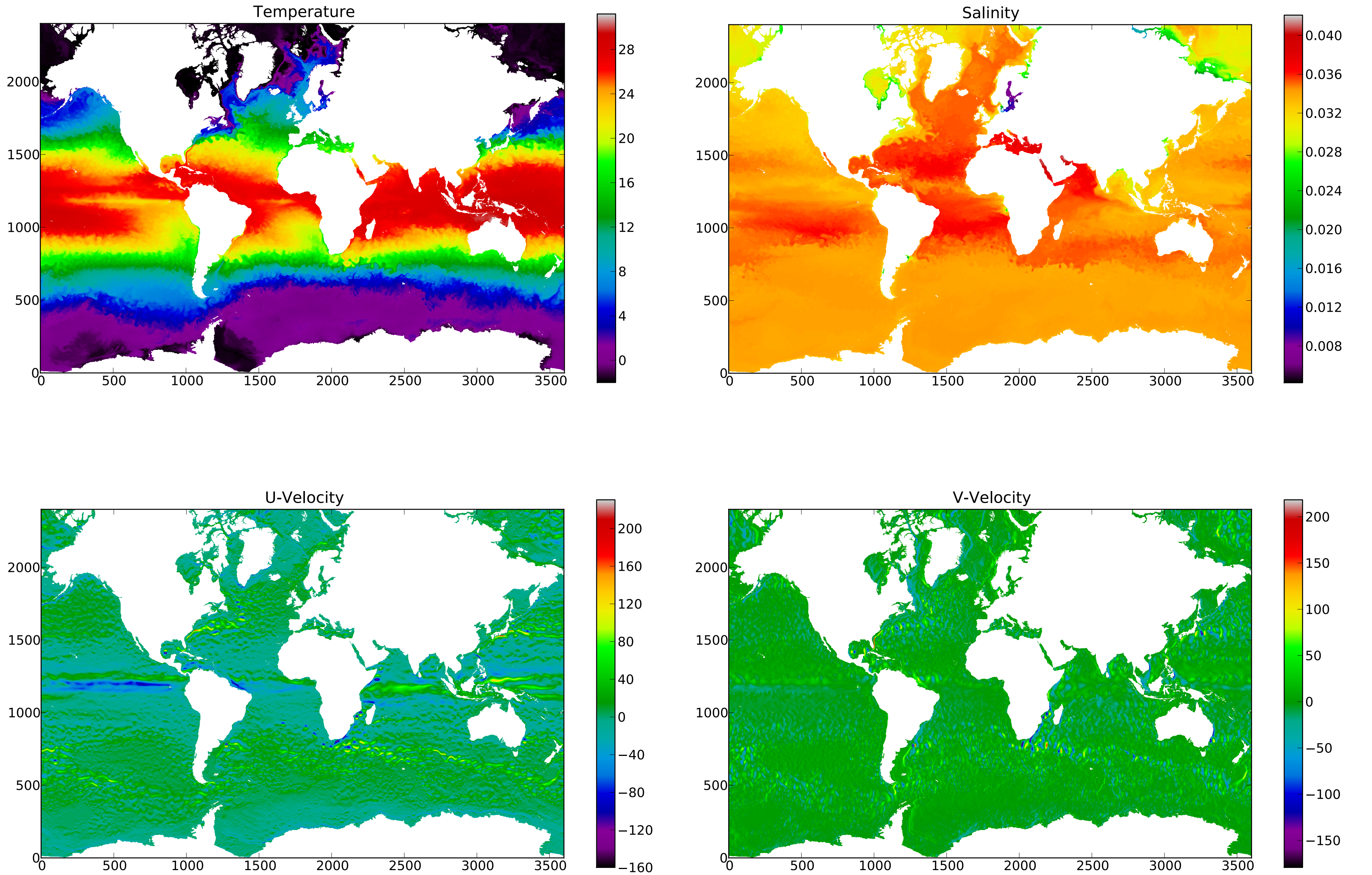
# Introduction

- How should we manage scientific data with “holes”?
  - Some applications, like JPEG 2000, expect logically rectangular data, but some sources, like the Parallel Ocean Program (POP), generate data that isn’t defined on certain subsets.
  - We refer to grid points that lack well-defined, scientifically meaningful sample values as “masked” samples.

• POP temperature field, 2400 x 3600 x 42 grid points.
•Unmasked samples carry meaningful values.
•Masked samples are filled with meaningless dummy values.

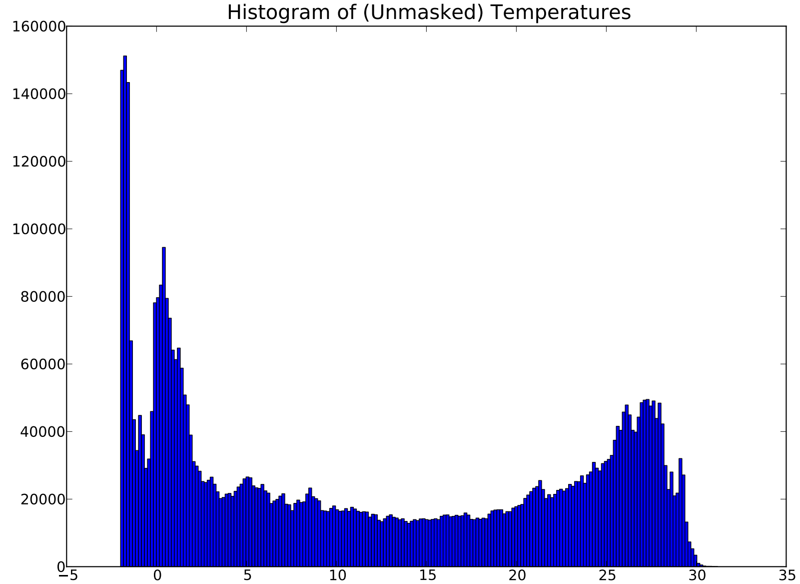


# POP Data Fields

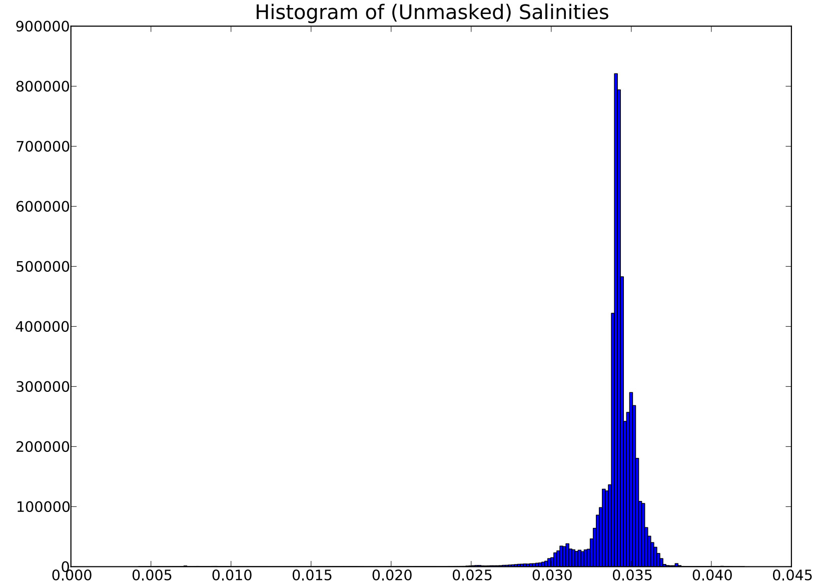


# POP Data Field Histograms

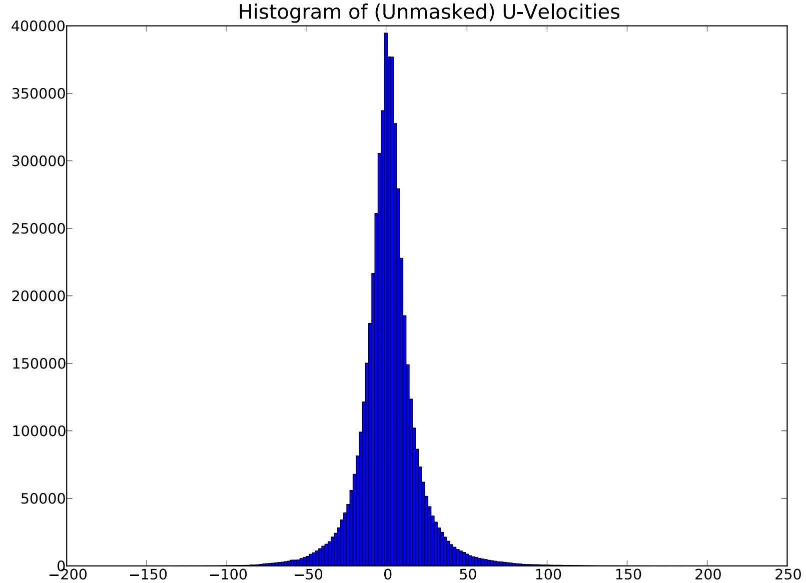
Histogram of (Unmasked) Temperatures



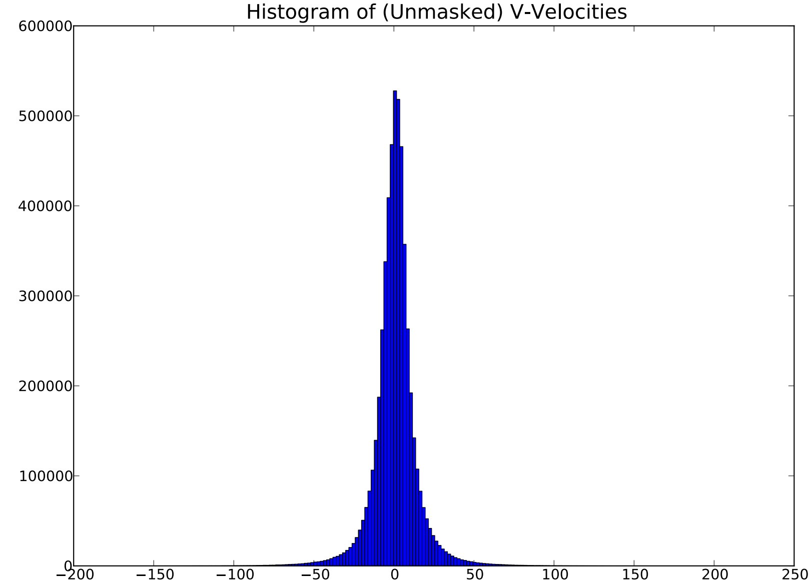

Histogram of (Unmasked) Salinities




Histogram of (Unmasked) U-Velocities

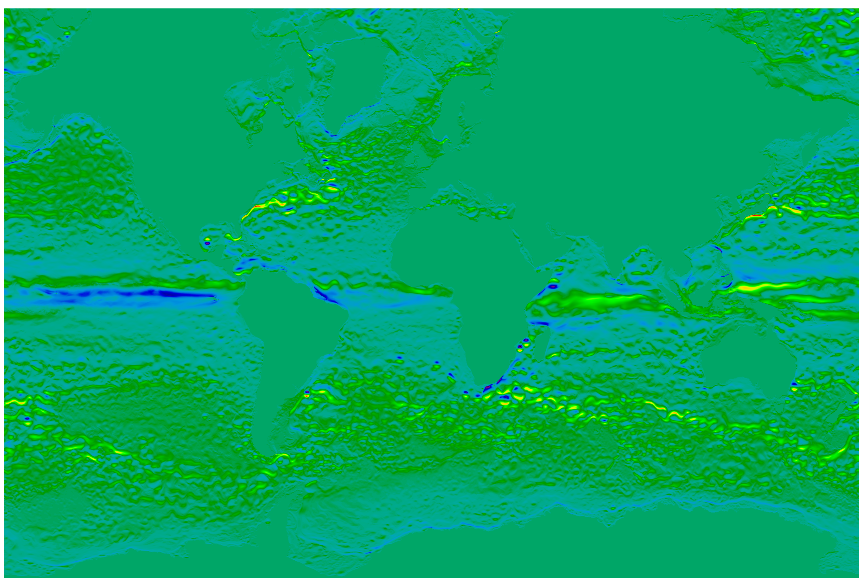



Histogram of (Unmasked) V-Velocities

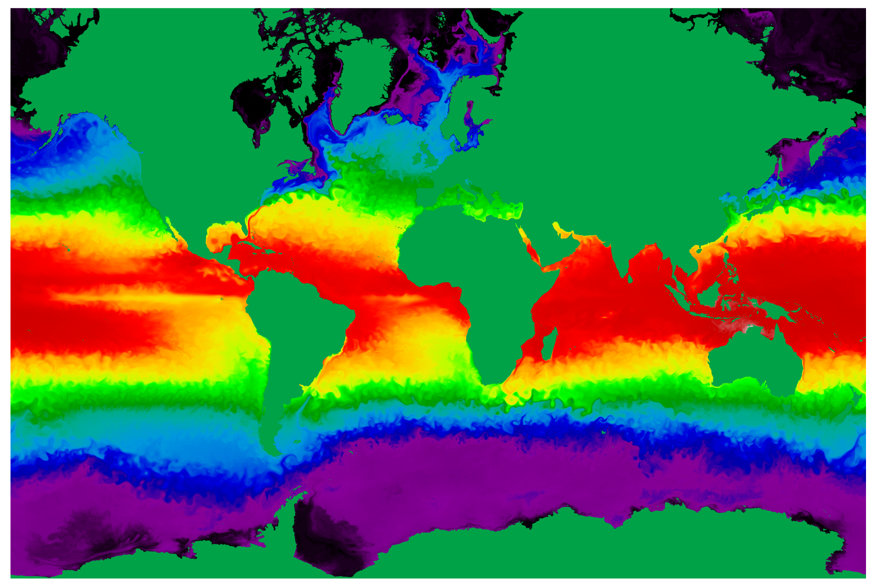


# First Idea: Fill with Global Mean

- Fill masked samples with a prudently chosen constant value like the global mean of the *unmasked* data.
  - The unmasked mean is easy to compute and may provide an adequate global approximation for data with unimodal, symmetric, sharply peaked distributions, such as the velocity components (L).
  - For other data, however, the mean may be a poor approximation globally (R), creating large discontinuities that cause severe artifacts when compressed data is reconstructed at reduced bit rates for applications like visualization.



U-velocity, mean-filled masked samples



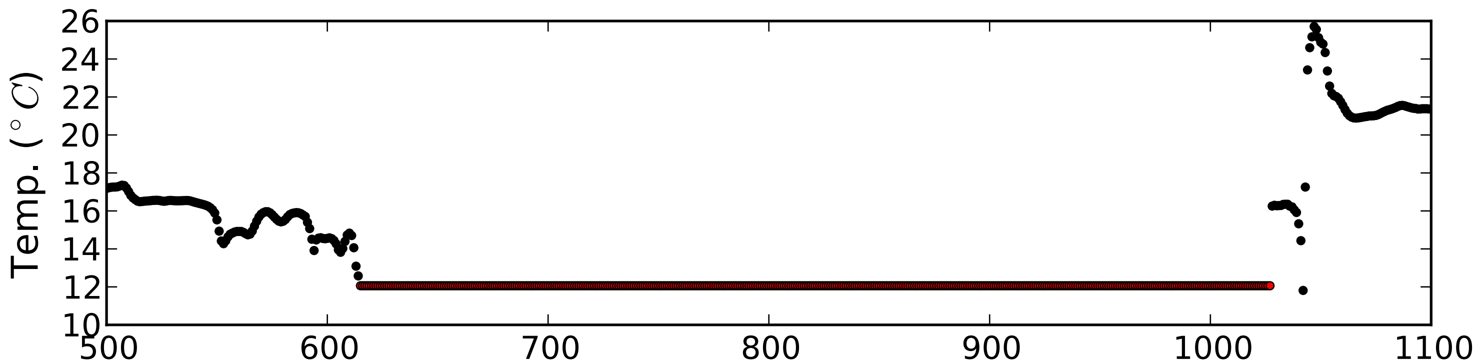
Surface temperature, mean-filled masked samples

# Interpolation Requirements

1. Since masked samples carry no meaningful scientific information, they can be interpolated arbitrarily as long as the *unmasked* samples are not modified.
2. Interpolating masked data for the sake of an application like JPEG 2000 should have a computational cost commensurate with the cost of the application and should require minimal hand-tuning for new data sets.
3. The interpolation should be chosen to minimize the introduction of artifacts into subsequent data analyses.
4. The interpolation should also minimize any collateral impact on subsequent data management, such as increased storage, transmission, and processing costs.

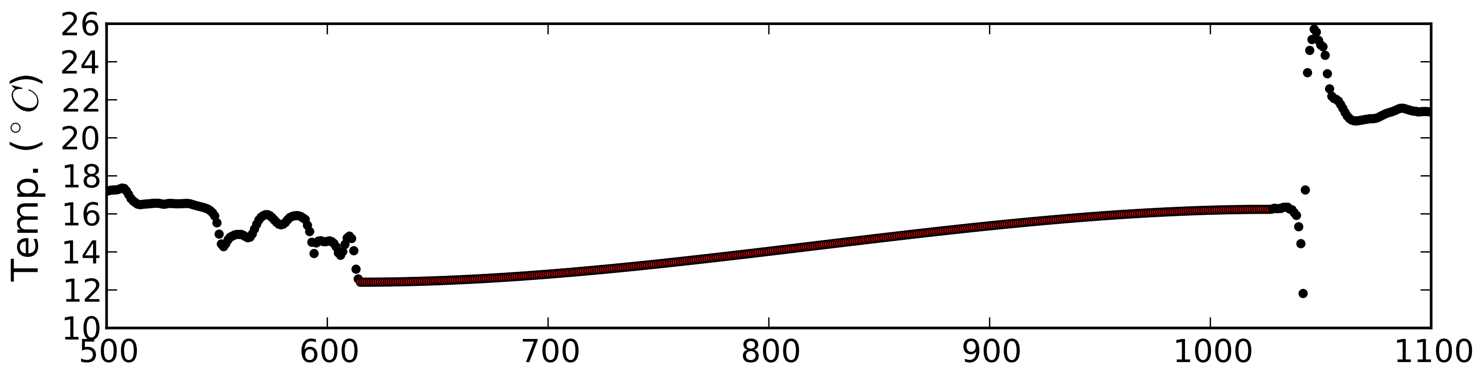
# What About Smooth Interpolation?

- 1-D slice of global mean interpolation across N. America:




  - The spike in temperature at offset 1050 is the Gulf Stream near Cape Hatteras, NC.

- Cubic interpolation is easy in 1-D but *much* harder in 2-3 dimensions, at exascale, with complicated boundaries,....



# Smooth Interpolation?

- Conventional interpolation (bicubic or tricubic B-splines, radial basis functions, kriging methods) can be expensive to compute and can require significant hand-tuning for each interpolation mask, so we are developing a highly scalable interpolation approach tailored for arrays being input to JPEG 2000 for scalable data compression.
- **Insight**: JPEG 2000 quantizes and entropy-encodes the discrete wavelet transform (DWT) representation of data so the relevant notion of "smoothness" is minimal wavelet-domain entropy rather than differentiability.
- **Runtime efficiency:** Our method takes advantage of the low computational complexity (linear cost scaling) of discrete wavelet transforms in multiple dimensions.

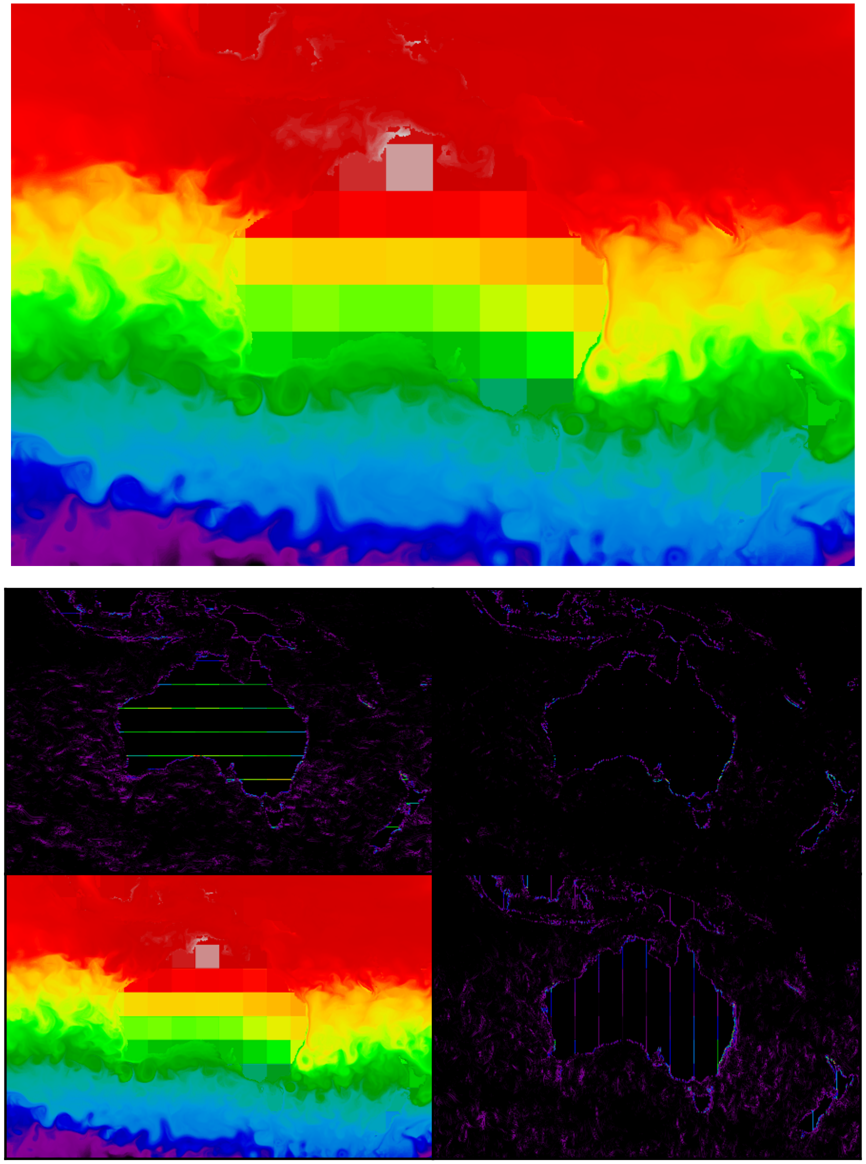
# Technical Approach

- Inspired by digital signal processing with 1-dimensional filter banks:  Use the degrees-of-freedom given by the *masked* samples to minimize the *local* (masked) entropy of DWT bandpass subbands.
  - This can be formulated as a least-squares minimum-norm optimization problem in the wavelet domain.
- Starts with a coarse block-based initialization of the masked samples in the spatial domain.
- Smoothing performed by *L* forward-inverse *L*-level DWTs.
  - No large unstructured iterative linear solvers!
- This quickly yields a smooth interpolant that preserves unmasked data exactly, with minimal wavelet-domain entropy in masked regions, and no mask-specific tuning.

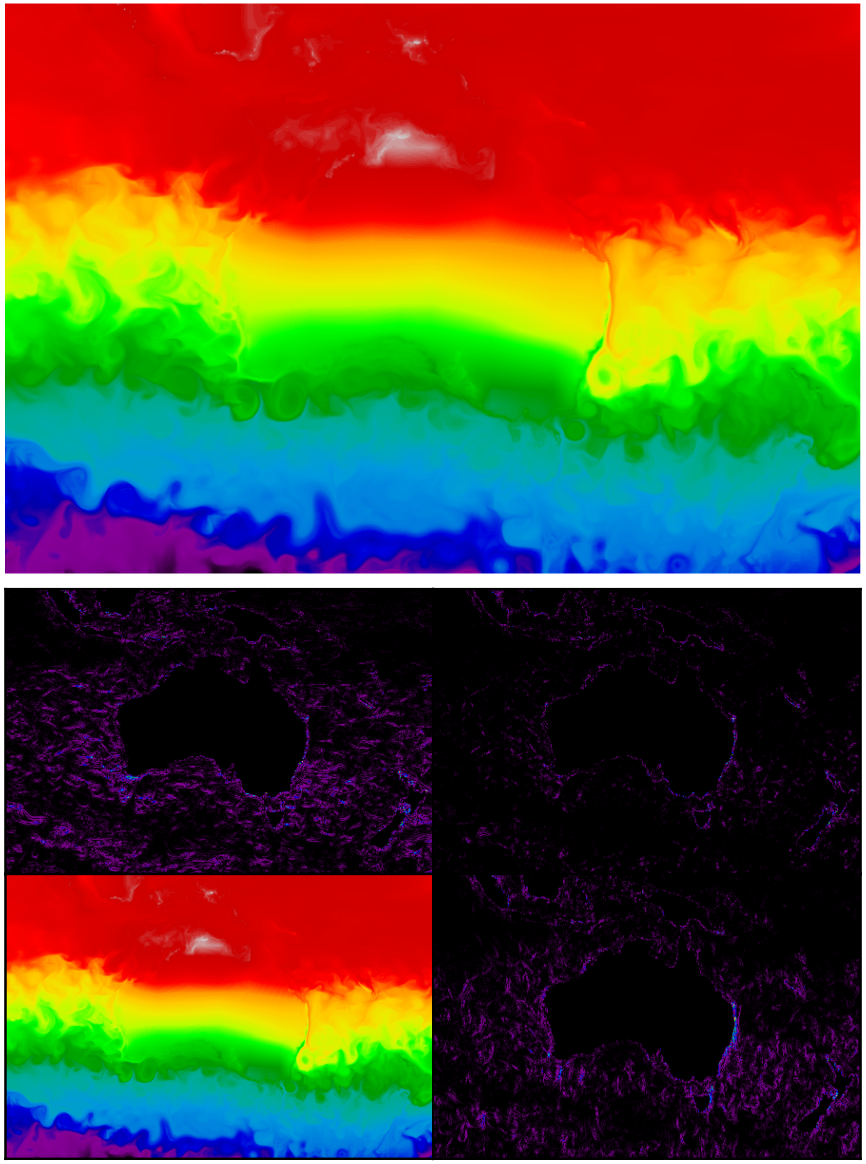
# 2-D POP Temperature Data Example

## 600 x 900-sample detail (Australia )

Block-filled initial array; first-level DWT subbands

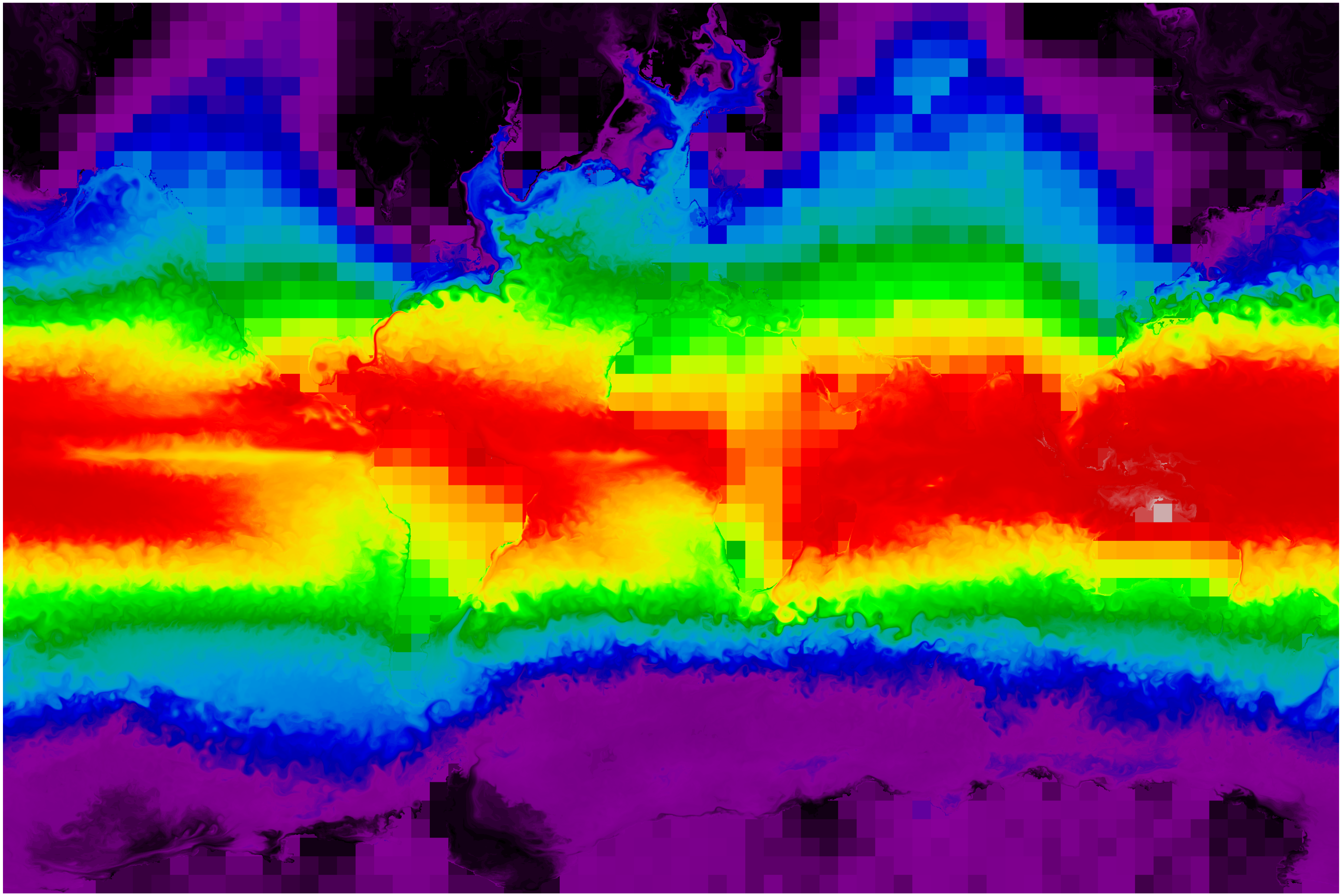


Wavelet-smoothed interpolant; first-level subbands



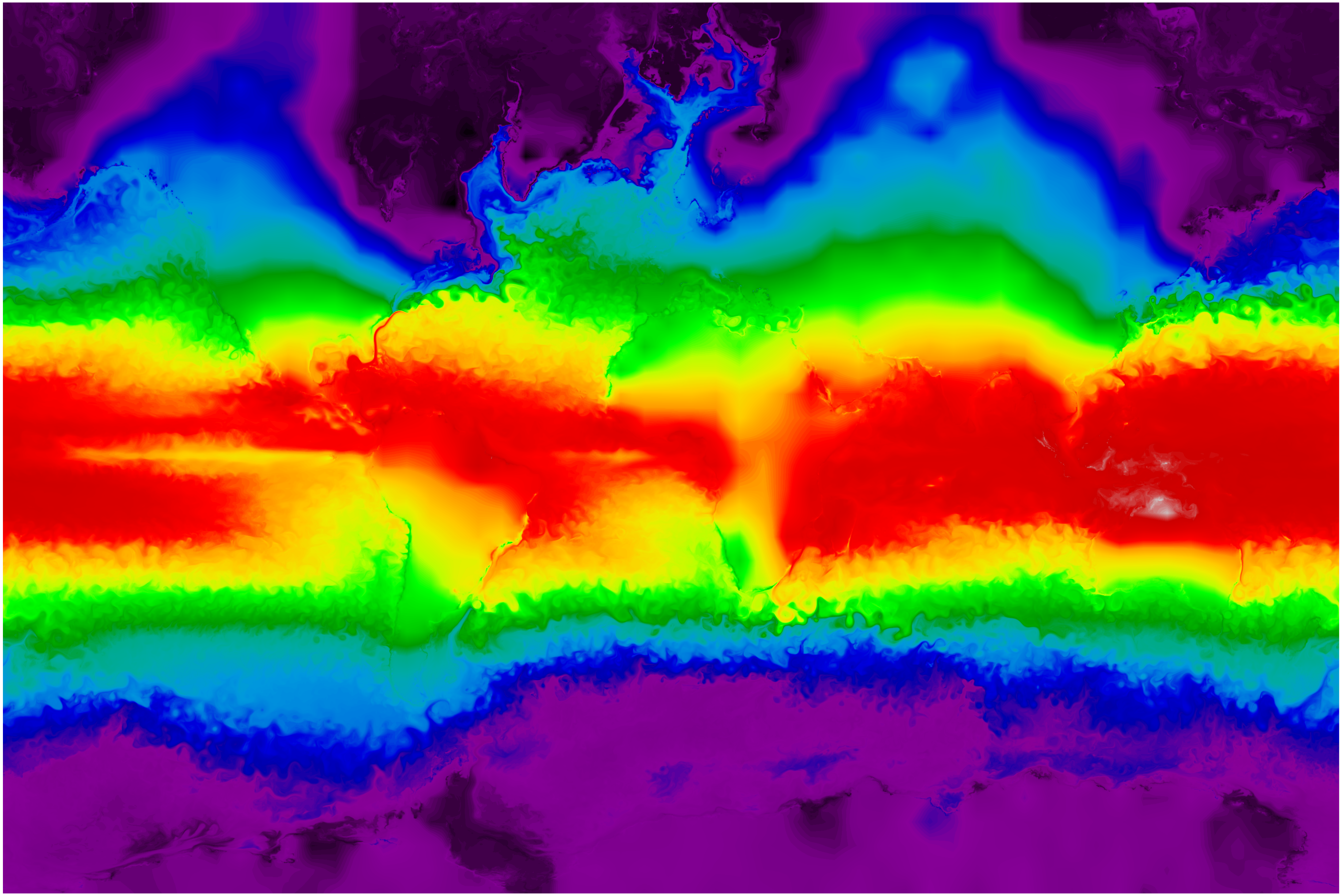
# Global POP Temperature Data

## Block-Filled Initial Array



# Global POP Temperature Data

## Wavelet-Smoothed Interpolant

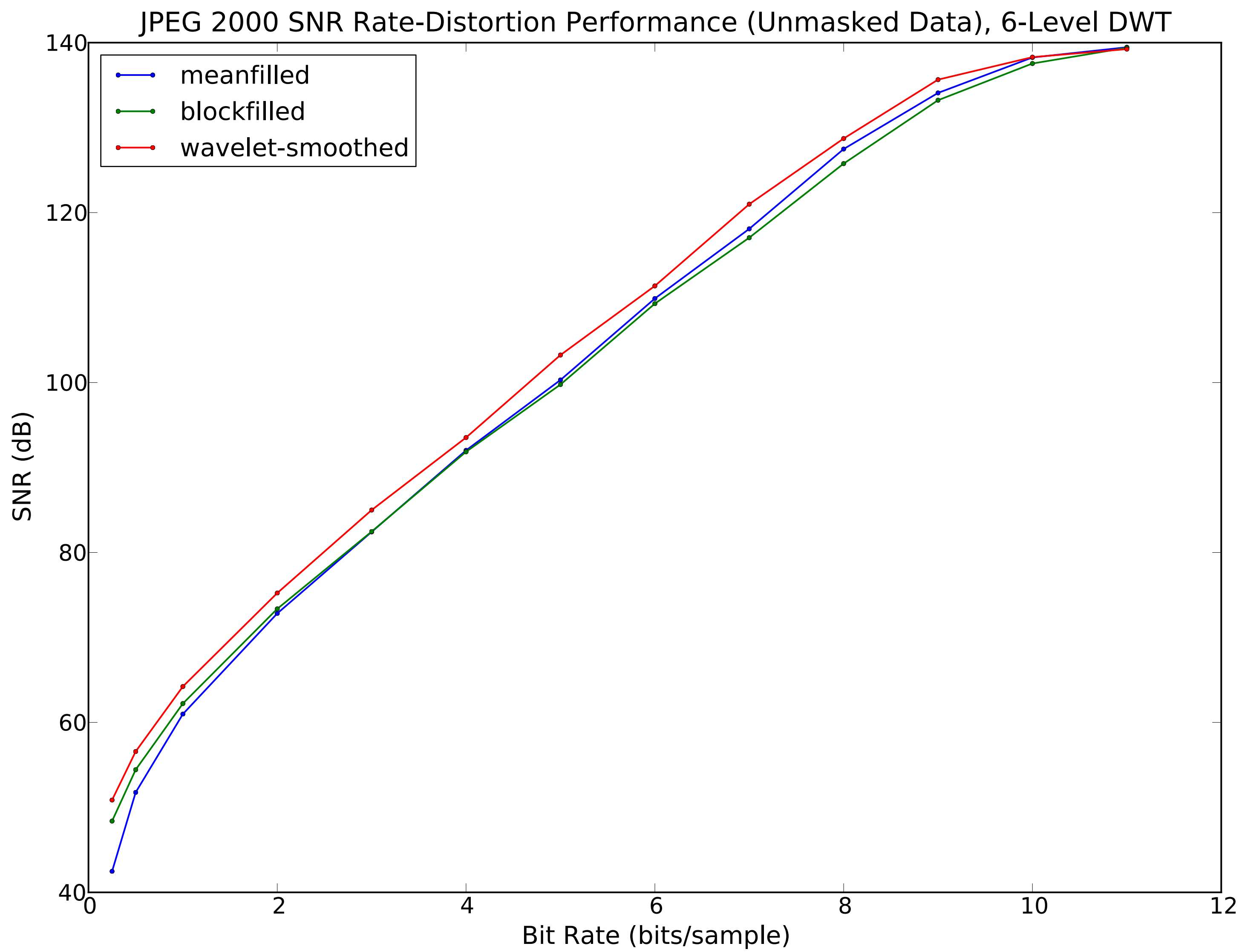


# 2-D POP Temperature Data Example

- 6-level DWT, 5-tap/3-tap irreversible wavelet filter bank
  - Uses the author's causal minimal realization of McMillan degree 2: the lifted filter bank requires just 1 mult and 2 adds per unit input. Symmetric boundary conditions per JPEG 2000.
  - In-place implementation: filtered data overwrites input in 2-D interleaved JPEG 2000 format; no additional memory required.
- JPEG 2000 compression of interpolated arrays
  - Floating point data prequantized to 27-bit precision (167 dB SNR) for input to JPEG 2000; floating point I/O is also possible.
  - Codestream optimized for decoding at preset entropies ("quality layers") of 0.25, 0.5, 1, 2, 3, 4, …, 12 bits/sample.
  - Data for each interpolation encoded once with JPEG 2000, then decoded at all preset rates from the same compressed file to make distortion measurements. ("encode once, decode many")

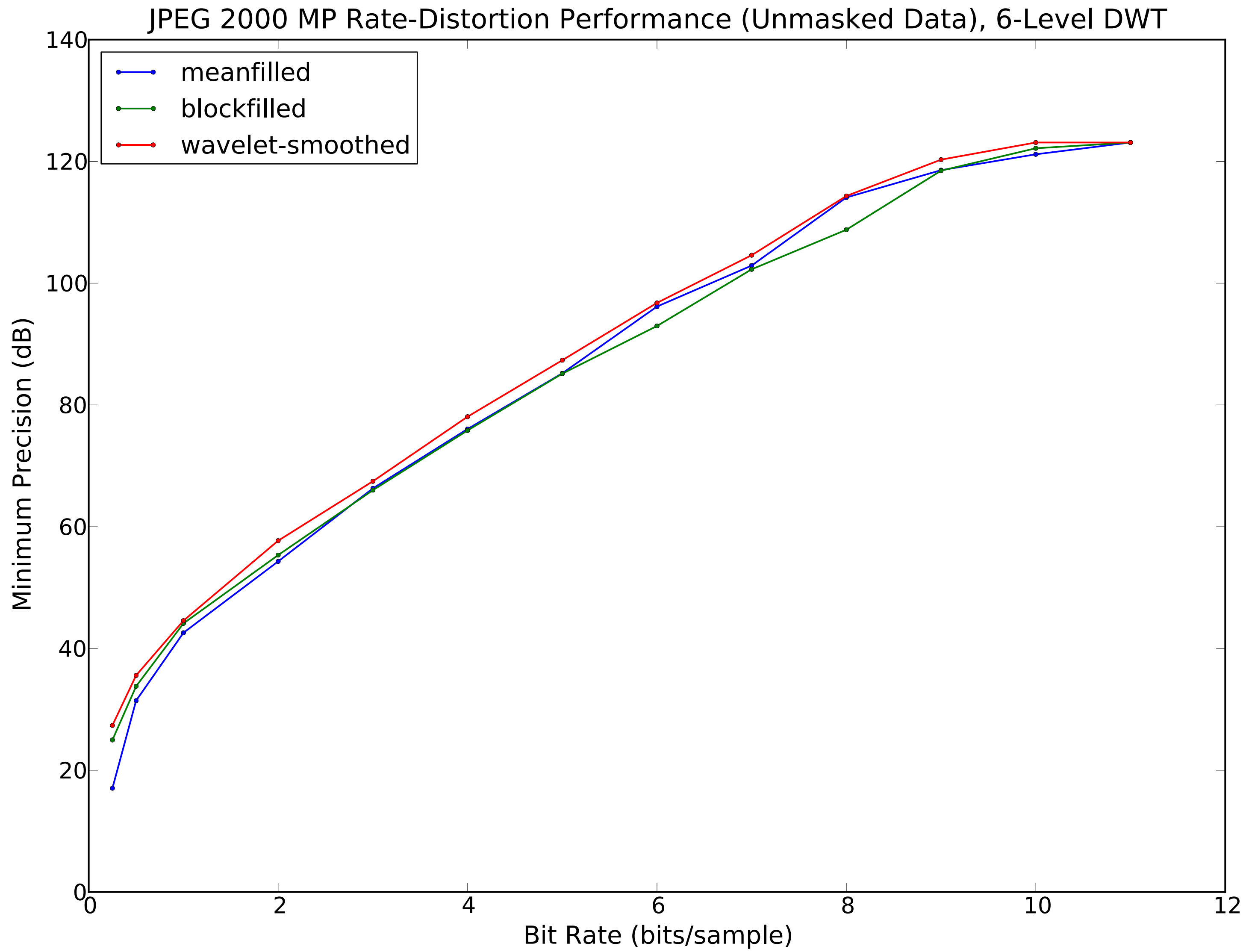
# Signal-to-Noise-Ratio Performance

- Error taken over unmasked (meaningful) temperature data only.



# Minimum Precision Performance

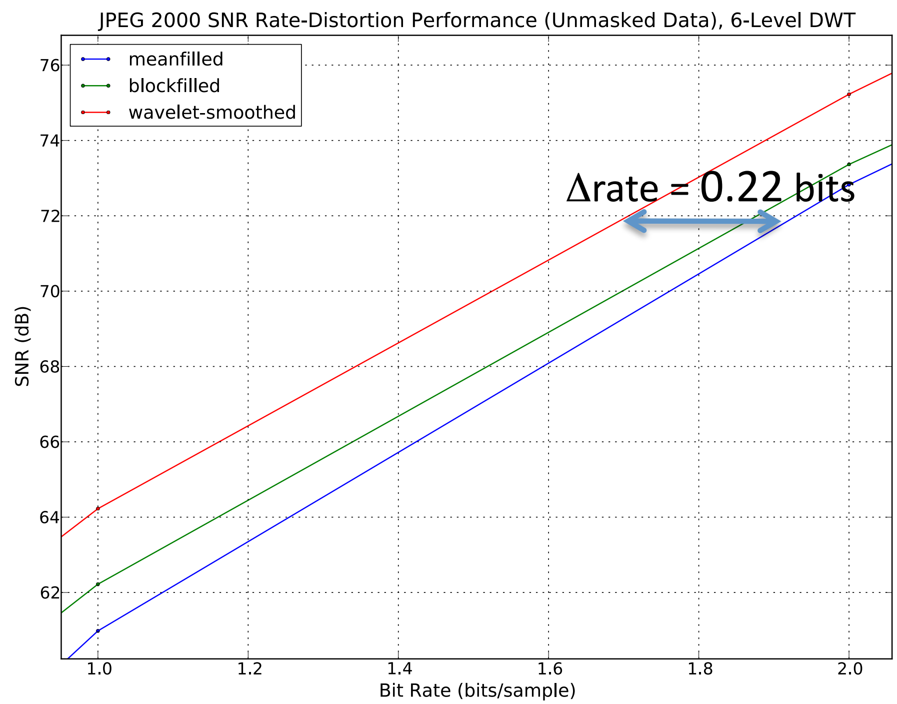
- Minimum Precision = $20*\log_{10}(\text{std. dev.} / \text{max. error})$ dB



# How Much Improvement Is This?

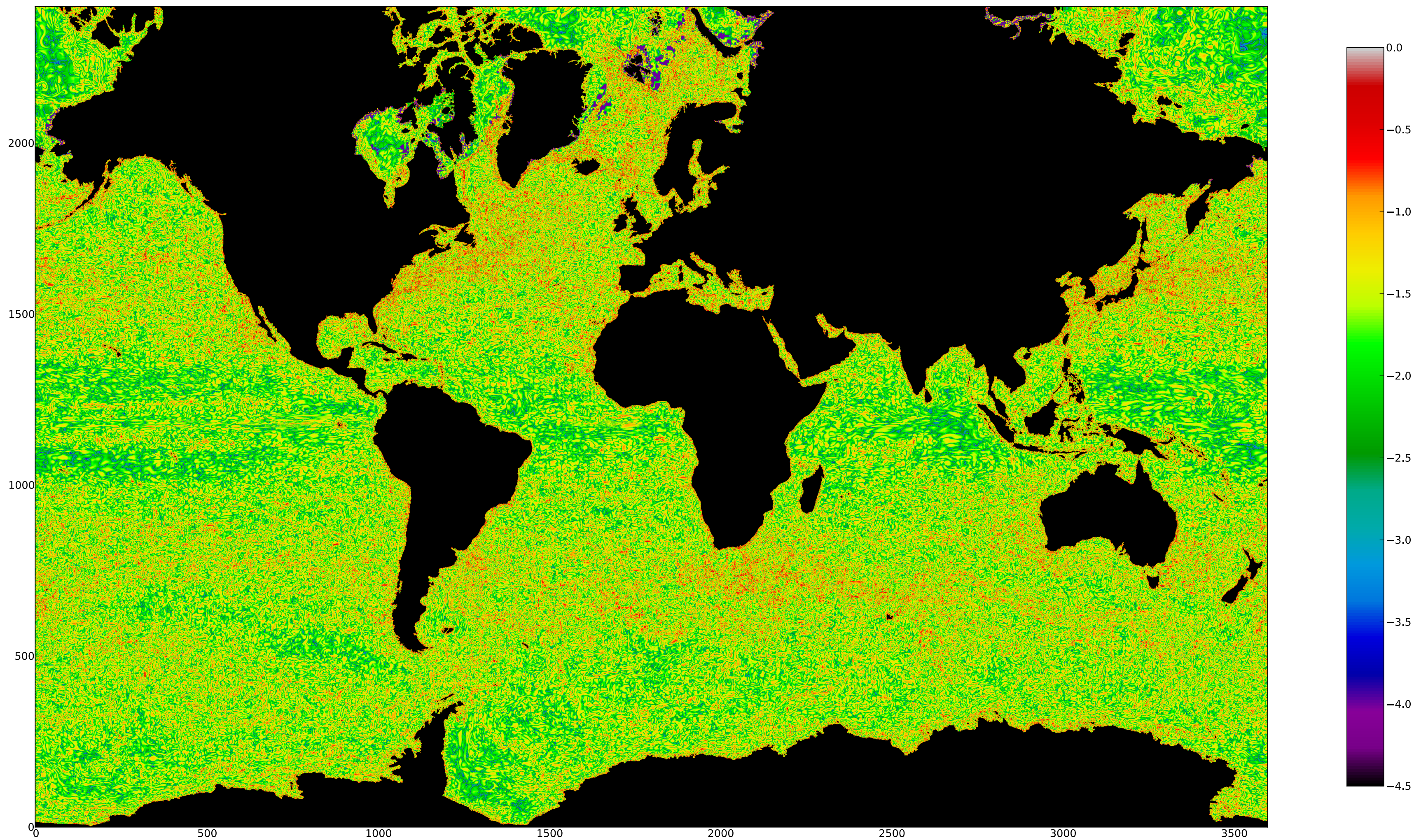
- Average SNR improvement (across all rates tested) using wavelet-smoothed vs. mean-filled interpolation of POP temperature data was 2.5 dB.
- Average improvement in Minimum Precision was 2.2 dB.
- Differences of a few tenths of a dB are considered visually significant in the image processing literature.

For a rate-centric perspective, in a POP visualization application requiring 72 dB unmasked SNR (12-bit nominal fidelity), a coding difference of 0.22 bits/sample translates into a bandwidth reduction of over 10% (from 1.93 to 1.71 bits per sample), due solely to improved data interpolation across the *masked* (meaningless) portions of the array.



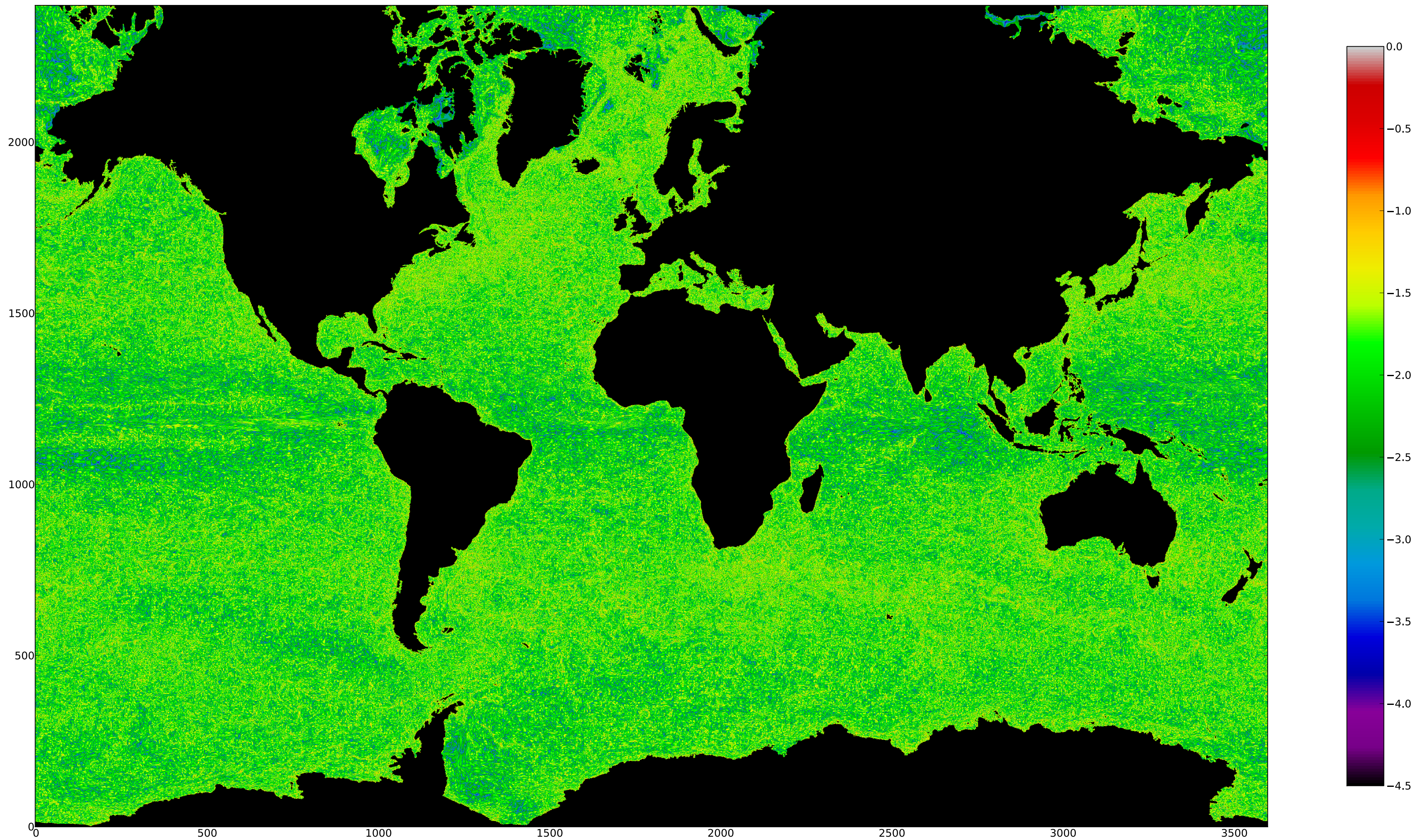
# Difference Images @ 0.25 bits/sample

- Logarithmic-scaled error images with identical color maps.
  - Note concentration of large errors (i.e., coding artifacts) near coastlines in reconstructed temperature data when using mean-filled interpolation:



# Difference Images @ 0.25 bits/sample

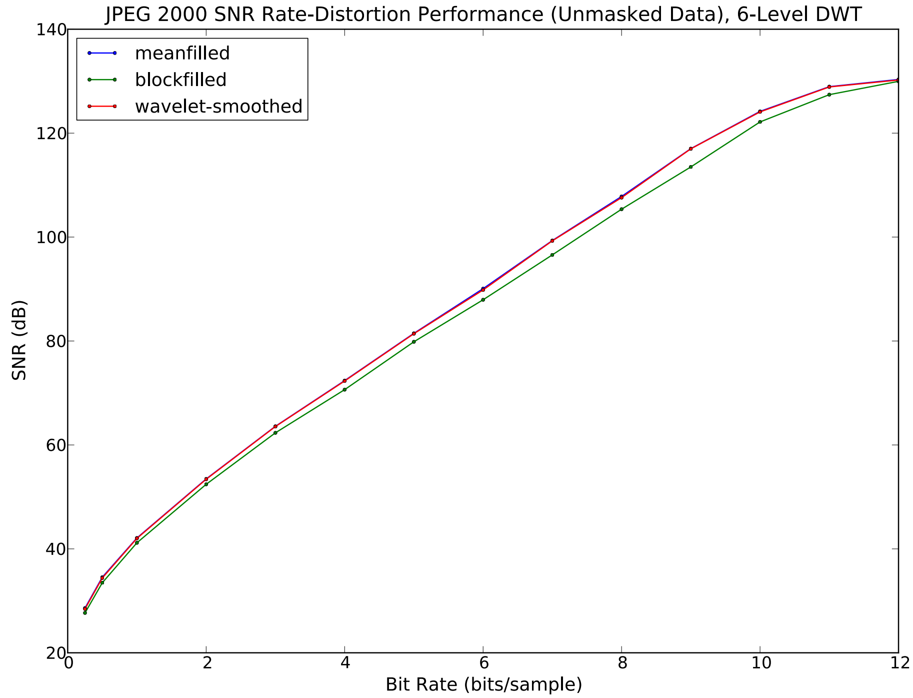
- Greatly reduced concentration of large errors near coastlines when using wavelet-smoothed interpolation on temperature data:



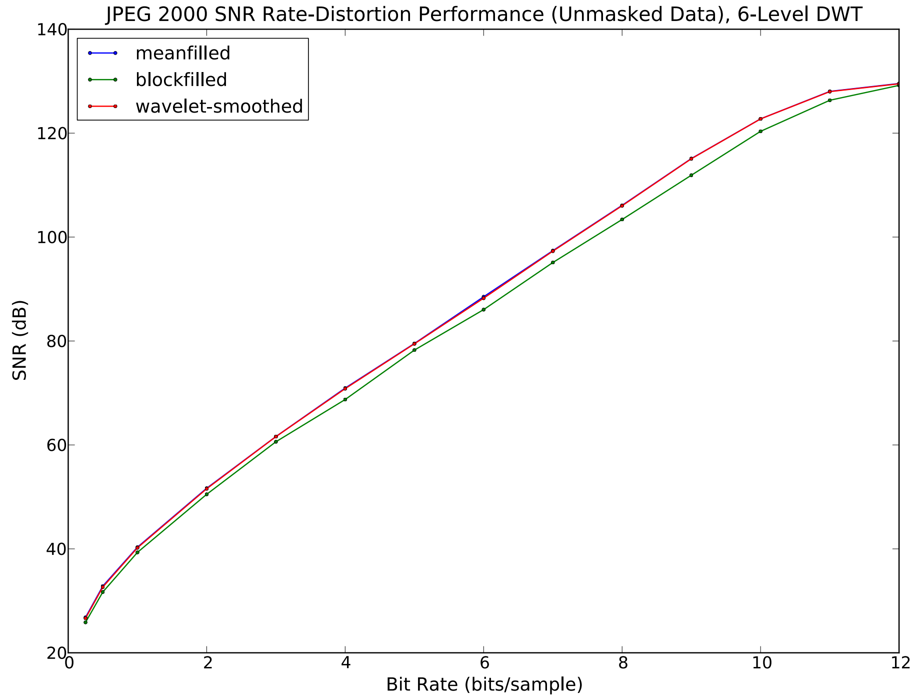
# Velocity Data Components

- As expected from the histograms, wavelet smoothing provides no improvement in coding performance over meanfilled interpolation, although, interestingly, both outperform block-filled interpolation.

U-Velocity SNR Performance:



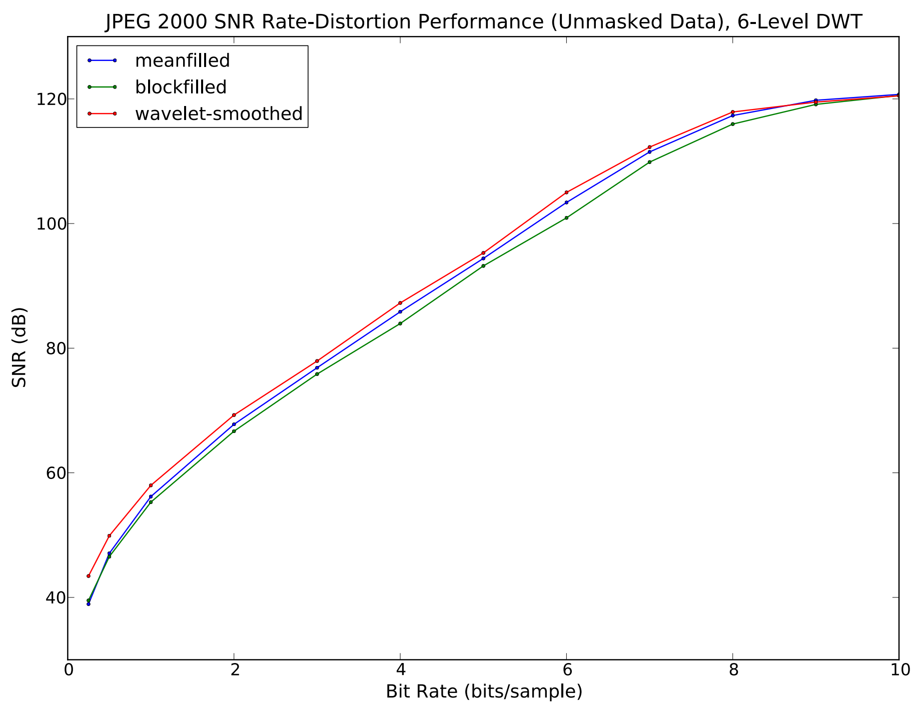

V-Velocity SNR Performance:

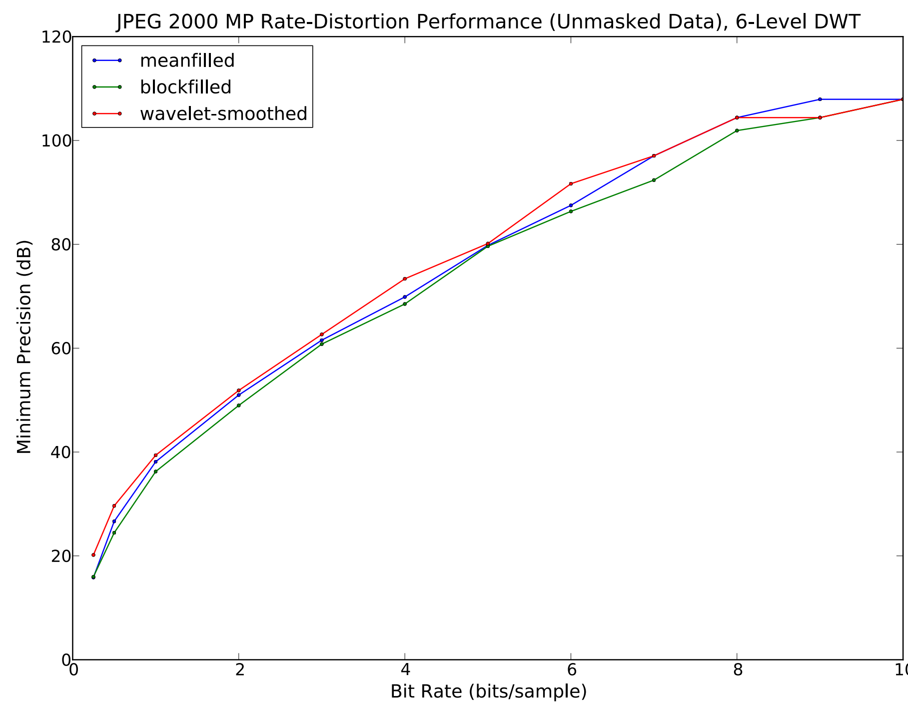



Minimum Precision behaves similarly for U- and V-velocity.

# Salinity Data

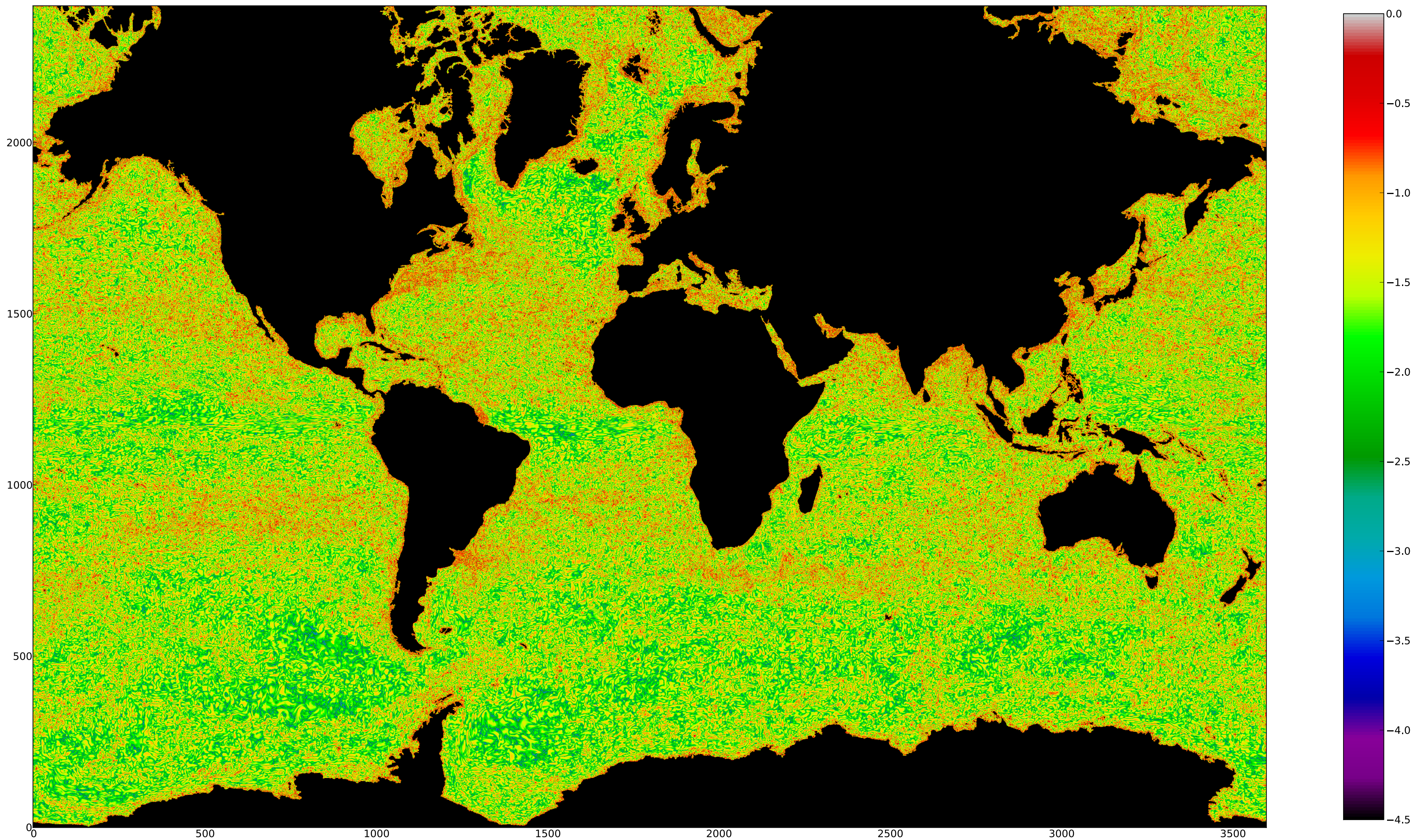
- The salinity histogram lies somewhere in between the generalized Gaussian shapes of the u- and v-velocity histograms and the multi-modal shape of the temperature histogram.
- Average improvement in SNR across all rates tested for wavelet-smoothed vs. mean-filled interpolation was 1.4 dB.
  - Unlike the case of temperature data, the block-filled scheme for salinity data doesn't outperform mean-filled interpolation at *any* of the rates tested.



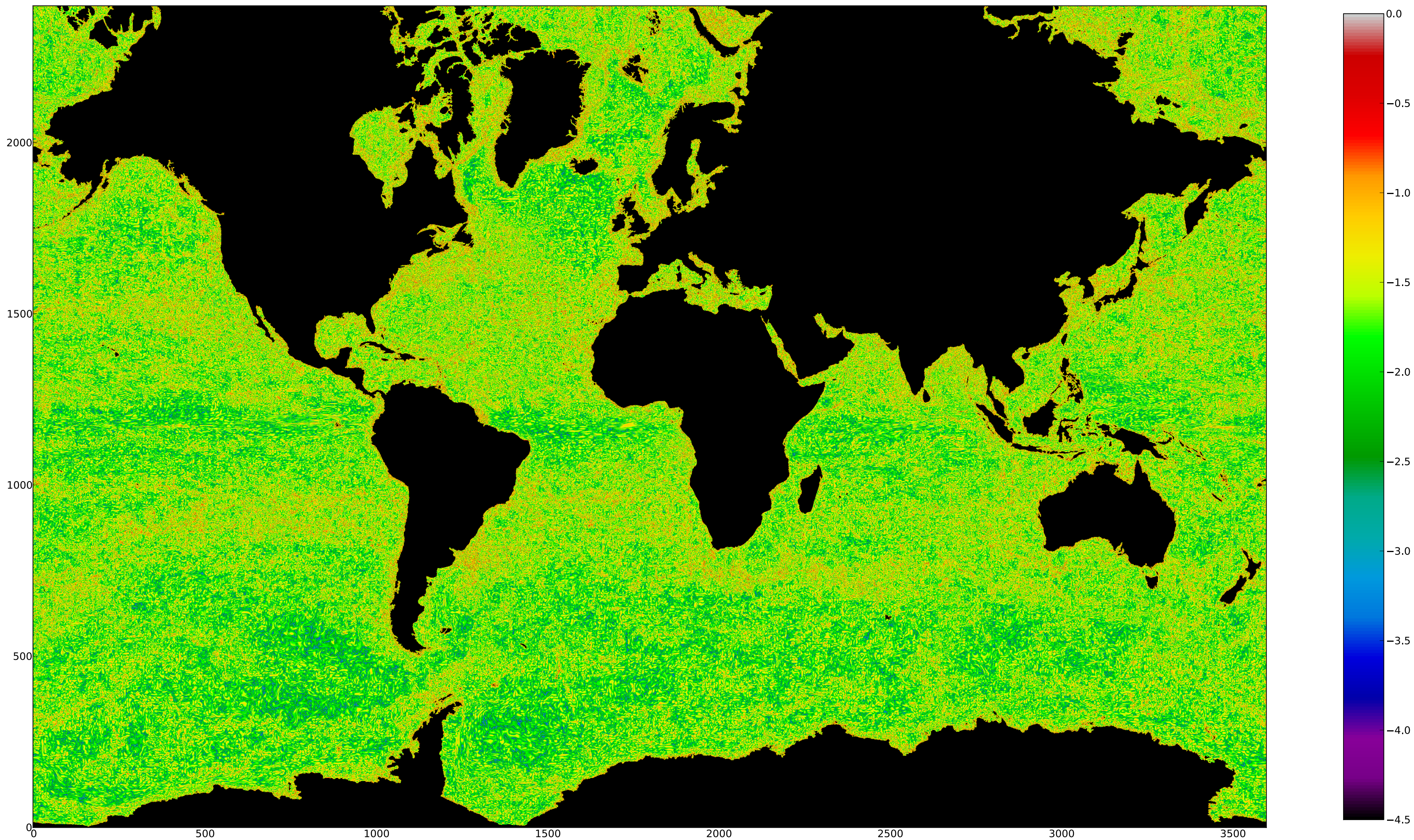
# Salinity Difference Images

- Mean-filled interpolation, reconstructed at 0.25 bits/sample:



# Salinity Difference Images

- Wavelet-smoothed interpolation reconstructed at 0.25 bits/sample



# Summary of Results

- Wavelet-smoothing is a highly scalable interpolation scheme for regions with complex boundaries on logically rectangular grids.
  - Computation is based on forward/inverse discrete wavelet transforms, so runtime complexity and memory scale linearly with respect to sample count.
    - Efficient state-of-the-art minimal realizations yield small constants (O(10)) for arithmetic complexity scaling, and in-situ implementation techniques make optimal use of memory.
  - Implementation in two dimensions using tensor product filter banks is straightforward and should generalize routinely to higher dimensions.
  - No hand-tuning required when the interpolation mask changes, making the method attractive for problems with time-varying masks.
  - Well-suited for interpolating undefined samples prior to JPEG 2000 encoding.
    - The method outperforms global mean interpolation, as judged by both SNR rate-distortion performance and low-rate artifact mitigation, for data distributions whose histograms do *not* take the form of sharply peaked, symmetric, unimodal probability density functions.
    - These performance advantages can hold even for data whose distribution differs only moderately from the peaked unimodal case, as demonstrated by POP salinity data.
  - The interpolation method is very general and is not tied to any particular class of applications, could be used for more generic smooth interpolation.

# Next Steps

- Understand the mathematical underpinnings of the method in terms of overdetermined linear systems, $\mathbf{A}x = b$, and compare the current computational algorithm to iterative linear solvers for the corresponding least-squares minimum-norm problem.
  - How close to optimal is the performance of the current algorithm?
- Implement the algorithm for 3-dimensional data and test on 3-D POP simulations as well as on a wider variety of simulation data.
- Implement the algorithm in C/C++ and enable frame-by-frame JPEG 2000 compression of simulation data as it is generated.
- Extend the method for time-varying masks, non-rectangular grids.
- For more general interpolation problems (not JPEG 2000 related):
  - Investigate differentiability and rate-of-convergence properties for the method when used to interpolate data sampled from smooth functions.
  - Does the method converge faster or yield smoother interpolants if used with "interpolating" wavelet filter banks?