Fusion of Multiresolution Seismic Tomography Maps Using Physics-informed Probability Graphical Models

Zheng Zhou¹, Peter Gerstoft¹, and Kim Olsen²

¹University of California San Diego
²San Diego State University

October 27, 2023
Fusion of Multiresolution Seismic Tomography Maps
Using Physics-informed Probability Graphical Models

Zheng Zhou¹, Peter Gerstoft¹, and Kim Olsen²

¹ University of California San Diego, La Jolla, CA, United States
² San Diego State University, San Diego, CA, United States

19 October 2023

SUMMARY
The resolution of velocity models obtained by tomography varies due to multiple factors and variables, such as the inversion approach, ray coverage, data quality, etc. Fusing such tomography models with different resolutions is desired when updating community models, to enable more accurate ground motion simulations. Toward this goal, we present a novel methodology to fuse multiresolution seismic tomography maps with probability graphical models (PGMs). The PGMs provide segmentation results, corresponding to various velocity intervals, in the seismic tomography images with multiple resolutions. Furthermore, by taking physical information (such as ray-path density) into consideration, we introduce physics-informed probability graphical models (PIPGMs). These models provide data-driven relations between subdomains with low (LR) and high (HR) resolutions. By transferring (segmented) distribution information from the HR regions, the details in the LR regions are enhanced by solving a maximum likelihood problem with prior knowledge from HR models informed. When updating areas bordering HR and LR regions, a patch-scanning policy is adopted to consider local patterns and avoid sharp boundaries. To evaluate the efficacy of the proposed PGM fusion method, we employ the model on both synthetic checkerboard models and a fault zone structure imaged from the 2019 Ridgecrest, CA, earthquake.
sequence. The Ridgecrest fault zone image consists of a shallow (top 1 km) high-resolution shear-wave velocity model obtained from ambient noise tomography, which is embedded into the coarser Southern California Earthquake Center Community shear wave Velocity Model version S4.26-M01. The efficacy of our model is underscored by misfits between observed and calculated travel times (using synthetic stations on the boundaries between HR and LR regions). These are 38% smaller than those obtained by conventional Gaussian interpolation. The proposed PGM fusion method can merge any type of gridded multiresolution velocity model, a valuable tool for computational seismology and ground motion estimation.

**Key words:** multiresolution model fusion, Markov Random Fields, physics-informed machine learning.

1 INTRODUCTION

The resolution of tomography velocity models varies due to inversion approaches, ray coverage, etc. For example, large-scale Community Velocity Models (CVMs) are typically characterized by low resolution, while high-resolution tomography models are limited to smaller areas with dense station coverage. Combining such tomography models with different resolutions is useful for improving community models, e.g., for ground motion estimation or dynamic rupture modeling, where a range of scales is needed (e.g., Yeh & Olsen 2023). However, directly superimposing models with different resolutions often sharp boundaries caused by misaligned patterns, which may cause artifacts in the modeled seismic waves. The problem of fusing low- and high-resolution models is conventionally addressed by extracting the homogeneous patterns from various models and utilizing them as boundary conditions for wave propagation. Border merging has been accomplished by Gaussian smoothing (Wu & Janson 2017) and by defining a weighting region (Ajala & Persaud 2021). However, this process relies on often time-consuming manual selection of the models and tuning of the parameters by experts. Inspired by the progress achieved in image super-resolution (Cheung et al. 2018) and image editing (Dhamo et al. 2020; Zhang et al. 2018), we introduce a method to fuse seismic tomography models with prob-
ability graphical models (PGMs), which enhances local HR structure and simultaneously preserves global smoothness in the fused model.

The exploration of multiscale problems has surged across diverse geophysical fields, encompassing full wave inversion, ground motion modeling, and surface-wave tomography. A pivotal discovery is the scale-dependent nature of anisotropy, a finding that has substantial ramifications for how we perceive the structure of the Earth (Van Houtte et al. 2006). Analyses from both the physical and signal processing viewpoints bring to light key challenges associated with the interrelation between seismic wave propagation and spatial heterogeneities. These studies have shown that seismic waves cannot distinguish between large-scale anisotropy and small-scale isotropic heterogeneities much smaller than a wavelength (Barbarossa & Sardellitti 2020). The problem intensifies when taking into account the potent spatial and directional dependency of tomographic resolution. Such dependency may provoke space- and direction-oriented smoothing (Dhamo et al. 2020), eliciting apparent anisotropy fluctuations that bear no intrinsic connection to Earth’s anisotropy. The construction of multiscale models has traditionally employed tactics such as multiscale element fitting (Fichtner et al. 2013), Gaussian kernel smoothing (Fang & Zhang 2014), and adaptive filtering methods, such as sparse learning (Zhu et al. 2015). Zhang & Ben-Zion (2023) proposed a data-driven dictionary learning method for transformations between seismic velocity models of different resolutions, which involves a linear decomposition of an input signal using a small set of basis signals, or atoms, learned from the HR and LR pairwise data. Although these methods have shown to be efficient, they might not fully grasp the vast complexity of Earth’s structure. The method we proposed assimilates the vital element of the rotation-invariant property, presenting a more comprehensive depiction of Earth’s interior. It provides a flexible framework that allows for varying data quantities for training. Consequently, this methodology heralds the advent of more inclusive, adaptable, and precise modeling of Earth’s subsurface structures.

Probability Graphic Methods (PGMs) are capable of processing images with complex structures, owing to their power to extract the underlying relations among images (Ortega et al. 2018; Shuman et al. 2013). Modeling pixel points and their interactions as
a graph, a graph space is defined according to the eigenspace of a graph representation for image analysis (Sandryhaila & Moura 2013), including denoising (Liu et al. 2018), segmentation (Zhang et al. 2022a), and seismic detection problems (Cannavò et al. 2017). Beyond normal graphs, PGMs have been extended to high-dimensional spaces, such as multilayer graphs (Das & Ortega 2020) and hypergraphs (Zhang et al. 2022b). Further, graph neural networks (GNNs) and graph convolutional networks (GCNs) are important tools in image processing and computer vision (van den Ende & Ampuero 2020; Kim et al. 2021). Bayesian methods provide a framework for modeling uncertainty, learning from data, and making predictions, and they have found broad applications in a variety of seismic applications, such as seismic tomography (Zhao et al. 2022), full wave inversion (Zhang et al. 2023), and ground motion prediction (Mu & Yuen 2016). Among all the PGMs and Bayesian methods, Markov Random Fields (MRFs) constitute a popular and effective approach for supervised structure learning tasks involving the mapping between complex geometric structures (Murphy 2001). MRFs provide an image restoration procedure, first suggested by Geman & Graffigne (1986), which is based on Bayesian inference for a spatially stochastic model. In contrast to convolution-based methods, the MRF procedure has been shown to yield an optimal and mathematically tractable result for image processing (Blake et al. 2011).

Recently, many novel methodologies, within the category of physics-informed machine learning (Gou et al. 2023), have gained much attention. These methods use a combination of conventional machine learning (primarily neural networks (Li et al. 2023)) and weighting mechanisms. They adapt the behavior of the learning models to the non-linear features of the solution and introduce physical information as a guide, eventually improving the current limitations of learning models.

Combining the physics-informed mechanism and the MRF model, we propose a physics-informed probability graphical model (PIPGM) that captures the relations between subdomains with multiple resolutions. Here, we focus on models with well-defined, separate high-resolution (HR) and low-resolution (LR) areas. By transferring the information from the HR subdomain, the details in the LR areas are enhanced by solving a maximum
Figure 1. (a) Excerpt of S-wave velocities from the Southern California Earthquake Center (SCEC) Community Velocity Model (CVM) version S-4.26 (hereafter referred to as the low-resolution (LR) CVM) at 0.5 km depth around the Ridgecrest area. (b) High-resolution (HR) S-wave map from 1 Hz Rayleigh wave tomography from Zhou et al. (2022). (c) A direct superposition of the HR and LR models. These two models share some patterns in the low-velocity zones, but many mismatched detailed patterns are present where the two models overlap, which results in sharp and misaligned boundaries in those areas. Our PIPGM is applied to the mismatched boundary areas between the two red bounding boxes; note that the pixels in this area belong to the effective vertices set $V$.

likelihood problem with prior knowledge from the HR areas. Evaluation tests on both checkerboard and a fault zone model derived from the 2019 Ridgecrest, CA, earthquake are performed to demonstrate its efficacy. Our model efficacy is evaluated by the misfit

Figure 2. A 6-cluster Gaussian Mixture model clustering is applied on the continuous velocity map $A$ (left), and this derives a 6-cluster discrete label map $X$ (right). The pixels with similar velocity information have been assigned the same label.
Figure 3. Each pixel has a continuous velocity value \( a_{i,j} \) and a discrete label mask \( x_{i,j} \). The object function designed for MAP estimation has two parts: (1) the data cost \( \theta_0 \) (0th-order neighboring potential) that forces the pixels with the same label to follow the same Gaussian distribution, and (2) the smoothness cost \( \theta_1 \) (1st-order neighboring potential) that promotes the smoothness among neighboring pixels (Koller & Friedman 2009).

between observed and calculated travel times, demonstrating that our PIPGM method is generally superior to widely-used conventional methods (see Experiments section).

The contributions of this article are two-fold: (1) we introduce a PIPGM fusion method for combining tomography maps with various resolutions, and (2) we introduce physical information as a guide for the graph learning process.

2 FUNDAMENTAL MODEL SETUP

The objective is to estimate the true velocity field \( A \) from \( A_{LR} \) (low-resolution) and \( A_{HR} \) (high-resolution) models, focusing on merging their borders optimally (as shown in Fig. 1). In our model, a discrete class label map ties the spatial velocity field together. It is initialized from a continuous velocity map \( A \) and a 6-cluster discrete label map \( X \).

Each pixel, described by an \((i, j)\) coordinate, contains a label \( X_{i,j} \) and a velocity \( A_{i,j} \), with velocities of the same label following the Gaussian distribution \( \mathcal{N}(\mu_n, \sigma_n^2) \). Thus, in a graph, velocities \( A \) are on top of labels \( X \) (Fig. 3). \( d \) represents all possible labels of \( X_{i,j} \) (namely, \( d = \{1, 2, 3, 4, 5, 6\} \) here), and \( D \) represents all possible label \( X \) combinations for the entire map. The velocity map \( A \) is tied together via class labels \( X \).
3 MARKOV RANDOM FIELD MODELS (MRFS)

3.1 Bayesian estimation framework

Given the prior probabilities \( P(X) \) of label \( X \) and the likelihood densities \( P(A | X) \) of the observed velocity \( A \), the posterior probability is computed using the Bayes rule:

\[
P(X | A) = \frac{P(A | X)P(X)}{P(A)} \propto P(A | X)P(X). \quad (1)
\]

Here, \( P(A) \), the probability density function (PDF) of \( A \), is a fixed probability distribution (for given \( A \)) and does not affect the maximum a posteriori (MAP) estimation solution. The Bayesian labeling problem requires finding the MAP configuration. The MAP of labeling for observation \( A \) is given by:

\[
X^* = \arg \max_{X \in D} P(X | A), \quad (2)
\]

where \( D \) denotes a set of possible candidates of the discrete labels \( X \), and \( A \) represents the observation of the continuous velocities.

To derive the MAP solution, we need the prior probabilities and the likelihood functions. The likelihood function \( P(A | X) \) depends on the noise statistics and the underlying transformation from the truth to the observation.

3.2 Neighborhood System in MRF

The MRF, a model that analyzes spatial relations, uses a neighborhood system (Li 1994). The neighborhood system is defined as \( \mathcal{N} = \{ \mathcal{N}_{i,j} | \forall (i, j) \in \mathcal{V} \} \) (shown in Fig. 4), where \( \mathcal{V} \) contains all pixel indices, and \( \mathcal{N}_{i,j} \) includes neighboring pixels. The neighboring system of the given pixel with index \((i, j)\) can be decomposed as the union of 0th-, 1st-order, \( \cdots \) neighboring systems as

\[
\mathcal{N}_{i,j} = \mathcal{N}_{i,j}^0 \cup \mathcal{N}_{i,j}^1 \cup \cdots, \quad (3)
\]

and we define the n-th-order neighboring system of \((i, j)\) as

\[
\mathcal{N}_{i,j}^n = \{(i', j') | |i - i'| + |j - j'| = n\}, \quad (4)
\]
Figure 4. The neighborhood system $N_{i,j}$ (marked in gray/black) of the given center node $(i, j)$ (marked in black). The 1st-, 2nd-, and 3rd-order neighborhood systems of node $(i, j)$ are marked with numbers '1', '2', '3', and they can be represented as $N^1_{i,j}$, $N^2_{i,j}$, and $N^3_{i,j}$.

which means $(i', j')$ and $(i, j)$ have a Manhattan distance (Liu et al. 2013) of $n$. The pair $(V, N)$ constitutes a graph, and the neighboring system $N$ for pixel $(i, j)$ consists of the center node (0th-order neighboring system) $N^0_{i,j} = \{(i, j)\}$ and a pair of neighboring pixels $N^1_{i,j} = \{(i', j')\}$.

The MRF is on $D$ with respect to $N$ if (1) $P(X_{i,j}) > 0, \forall X_{i,j} \in d, \forall (i,j) \in V$ (positivity) and (2) $P(X_{i,j} | X_{i',j'}, (i', j') \in V) = P(X_{i,j} | X_{i',j'}, (i', j') \in N_{i,j})$. Satisfying condition (1) above guarantees the model to be a random field. Condition (2) is the Markov local property. Considering that the label variable $X$ depends on velocity $A$ and is unobservable, it is assumed that the distribution of $P(X | A)$ follows exponential distributions, using Bayes’ rule and the conjugate distribution property (George et al. 1993).
3.3 MRF Prior and Posterior Energy

A model can be considered a valid MRF if and only if the probability distribution $P(X)$ of the configurations is an exponential distribution with normalization, defined as the following form

$$P(X) = \frac{1}{Z_1} e^{-U(X)},$$

(5)

where $Z_1$ is a normalizing constant, and $U(X)$ is the prior energy (Section 4.2 in Koller & Friedman (2009)).

The prior energy $U(X)$ can be expressed as the summation of neighboring potentials

$$U(X) = \sum_{n \in \mathcal{N}} \theta_n(X) = \sum_{\{(i,j)\} \in \mathcal{N}_0_{i,j}} \theta_0(X_{i,j}) + \sum_{\{(i',j')\} \in \mathcal{N}_1_{i,j}} \theta_1(X_{i,j}, X_{i',j'}) + \cdots,$$

(6)

where $\mathcal{N}$ is the set of all the possible neighboring systems, $\mathcal{N}_0_{i,j}$ and $\mathcal{N}_1_{i,j}$ are the 0th- and 1st-order neighboring systems, and $\theta_0$ and $\theta_1$ are the corresponding potentials (we define the 0th-order neighboring system as the set of all the possible indices $(i, j)$ as the centering pixel). To simplify, we only preserve the 0th-order and 1st-order neighboring potentials and truncate the higher-order ones in Eq. (6).

Because we assume that the velocities $A$ with given labels $X$ follow Gaussian distributions, the likelihood function can be expressed in an exponential format

$$P(A \mid X) = \frac{1}{Z_2} e^{-U(A \mid X)},$$

(7)

where $U(A \mid X)$ is called the likelihood energy. According to the Bayes rule Eq. (1), the posterior probability must be an exponential distribution

$$P(X \mid A) = \frac{1}{Z_3} e^{-U(X \mid A)},$$

(8)

where $Z_2$ and $Z_3$ are normalization constants. Taking the negative logarithm in Eqs. (7)–(8) gives the posterior energy

$$U(X \mid A) = U(X) + U(A \mid X) + C,$$

(9)

where $C$ is a constant related to the normalization constants $Z_1$, $Z_2$, and $Z_3$. Hence, given a fixed $A$, $X$ is also an MRF on $d$ with respect to $\mathcal{N}$. The MAP solution is equivalently
Figure 5. (a) LR CVM around the Ridgecrest area. (b) HR S-wave map from 1 Hz Rayleigh wave tomography. (c) and (d) corresponding gradient maps of (a) and (b). In the gradient maps, brighter (darker) areas indicate that velocities change sharply (slightly).

found by

\[ X^* = \arg\min_{X \in D} U(X \mid A), \]  

which is minimizing a negative log-likelihood problem in Eq. (10).

In summary, the MRF modeling process consists of the following steps: Defining a neighborhood system \( \mathcal{N} \), defining the prior potentials \( \theta_0 \), deriving the likelihood energy \( U(A \mid X) \), and deriving the posterior energy \( E(X) \) (which can be expressed as the summation of neighboring potential functions). Eqs. (7)–(10) show that we can express the posterior probability \( P(X \mid A) \) into the prior energy \( U(X) \) (which can be measured by multiple potentials) and the likelihood function energy \( U(A \mid X) \), and this gives a good reason for using MRF priors which means we can measure the conditional probabilities \( P(X \mid A) \) without knowing its specific expression.
4 PHYSICS-INFORMED PROBABILITY GRAPHICAL MODELS

4.1 Physics-Informed Mechanism

Our Physics-Informed PGM (PIPGM) follows a first-order MRF setting (Fig. 4) where each random variable has four neighbors on which it is conditionally dependent. The full conditional probability of the discrete random variable \( X_{i,j} \in \{1, \ldots, 6\} \) is the exponential of the sum of potentials (four 1st-order neighboring potentials \( \theta_1 \) between cluster labels and one 0th-order center data potential \( \theta_0 \) between cluster label and velocity) in conventional MRF settings. In image process problems, optimizing the whole map can be broken down into a group of suboptimization problems that optimize each pixel iteratively (Pulli et al. 2012). Inserting Eq. (6) into Eq. (9), we have

\[
- \log p(X_{i,j} \mid A_{i,j}) = U(X_{i,j} \mid A_{i,j}) \propto \theta_0(X_{i,j}, A_{i,j}) + \sum_{(i',j') \in \mathcal{N}_{i,j}} \theta_1(X_{i,j}, X_{i',j'}) + C, \tag{11}
\]

where \( C \) is a constant related to normalization factors \( Z_1, Z_2, \) and \( Z_3 \) (see Eqs. (5), (7), (8)), and

\[
\theta_0(X_{i,j}, A_{i,j}) = \frac{(A_{i,j} - \mu_n)^2}{\sigma_n^2} \tag{12}
\]

is the 0th-order neighboring potential (Li 2012) (also known as the data cost function) that relates \( X_{i,j} \) with the observed velocity data \( A_{i,j} \). \( \mu_n \) and \( \sigma_n^2 \) are the mean and variance, respectively, of all pixels with the same cluster label \( n = X_{i,j} \). It promotes that continuous velocity values \( \mathbf{A} \) sharing pixels with the same discrete label \( \mathbf{X} \) follow the same Gaussian distribution.

\[
\theta_1(X_{i,j}, X_{i',j'}) = 1 - \delta(X_{i,j}, X_{i',j'}) \tag{13}
\]

is the 1st-order neighboring potential (Li 2012) (also known as the smoothness cost function) that relates \( X_{i,j} \) to the 1st-order neighboring variable \( X_{i',j'} \) (see Fig. 4). This function encourages the neighboring pixels to share the same discrete label \( X_{i,j} \), promoting the model’s local smoothness.

The performance of standard MRF or potential function-based MRF schemes can be limited when dealing with complex geological structures. Assigning different neighboring
pixels with various importance weights based on physical information (also known as the physical informed mechanism) can effectively remove the anisotropic features of the model gradients, leading to improved inversion results, especially for geological structure formations.

In seismic tomography, ray-path coverage varies, with denser ray-path coverage usually leading to more reliable estimations of velocities. Here, we introduce a confidence score \(v_{R,i,j}\) based on the logarithmic format of the ray density for each pixel:

\[
v_{R,i,j} = \alpha \log(D_{i,j} + 1) + \beta, \tag{14}
\]

where \(D_{i,j} \in [0,315]\) is the number of ray paths through a given pixel. The logarithmic scaling parameters \(\alpha\) and \(\beta\) are empirically chosen as the optimal combination 0.08 and 0.90, respectively, and this gives \(v_{R,i,j} \in [0.9,1.1]\).

We combine the gradient information from both the LR and HR maps as the prior estimation of the sharpness of the local patterns (as shown in Fig. 5). The gradient is calculated by the Prewitt operator approximating the 1st-order derivatives of 2D images (Zuniga & Haralick 1987). The operator uses two \(3 \times 3\) kernels convolved with the original image to calculate approximations of the derivatives - one for horizontal changes and one for vertical. The kernels for the Prewitt operator can be represented as

\[
K^x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad K^y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}. \tag{15}
\]

The convolution of these kernels \((K^x\) and \(K^y\)) with the original image yields two gradient images, one for the x-direction \((G^x)\), and one for the y-direction \((G^y)\).

Once we have \(G^x\) and \(G^y\), we can find the gradient magnitude \(G_{i,j}\) at the pixel with index \((i,j)\) as

\[
G_{i,j} = \sqrt{G_{i,j}^x^2 + G_{i,j}^y^2}. \tag{16}
\]

The resulting image \(G\) is a gradient image showing the intensity of the edge. We applied the Perwitt operator on the LR and HR maps to generate the gradient images \(G^{LR}\) and \(G^{HR}\). We empirically choose the weighting parameter \(0 \leq \lambda \leq 1\) for balancing LR and
HR gradients as $\lambda = 0.2$:

$$G'_{i,j} = (1 - \lambda)G_{i,j}^{LR} + \lambda G_{i,j}^{HR}. \quad (17)$$

The overall range of the pixel values inside the gradient matrices $G^{LR}$ and $G^{HR}$ is $[0.05, 0.68]$. Scaling parameters $\alpha_G$ and $\beta_G$ are empirically set as 0.36 and 0.85, respectively, and this guarantees $v_{Gi,j} \in [0.9, 1.1]$:

$$v_{Gi,j} = \alpha_G \left(1 - G'_{i,j}\right) + \beta_G. \quad (18)$$

Generally, the larger the gradient weight term $v_{Gi,j}$, the lower the importance of the local smoothness cost. The gradient weight term $v_{Gi,j}$ is joined with the previous ray-density weight $v_{Ri,j}$, defining a physics-informed weight $\omega_{i,j}$

$$\omega_{i,j} = v_{Ri,j} v_{Gi,j}. \quad (19)$$

With the heuristic parameters chosen above, the range of the physics-informed weight is $[0.81, 1.21]$ with a mean of around 1. The physics-informed weight $\omega_{i,j}$ can adaptively assign a larger weight to the trusted nodes based on existing physical information (for example, ray-path density and gradient information are used in our research). Assigning the physics-informed weight in Eq. (11), we obtain the posterior probability function in the physics-informed probability graphical model (PIPGM)

$$- \log p(X_{i,j} \mid A_{i,j}) \propto \omega_{i,j} \theta_0(X_{i,j}, A_{i,j}) + \sum_{(i',j') \in N_{i,j}} \omega_{i',j'} \theta_1(X_{i,j}, X_{i',j'}) + C, \quad (20)$$

where $C$ is a constant related with to normalization factors $Z_1$, $Z_2$, and $Z_3$ (see Eqs. (5), (7), (8)). The objective function of the MAP problem of $X_{i,j}$ becomes

$$X_{i,j}^* = \arg \max_{X_{i,j}} p(X_{i,j} \mid A_{i,j})$$

$$= \arg \min_{X_{i,j}} \omega_{i,j} \theta_0(X_{i,j}, A_{i,j}) + \sum_{(i',j') \in N_{i,j}} \omega_{i',j'} \theta_1(X_{i,j}, X_{i',j'}) + C. \quad (21)$$

### 4.2 MCMC and Gibbs Sampling

Markov Chain Monte Carlo (MCMC) is a statistical method used to sample probability distributions (Melas & Wilson 2002; Sambridge & Mosegaard 2002). Gibbs sampling is a specific MCMC algorithm that can be used to iteratively sample a multivariate probability
distribution from the conditional distributions of each variable given the current values of the other variables (Carlo 2004). Combining MCMC with Gibbs sampling enables estimating complex probability distributions without explicit knowledge of the distribution.

We employ the MCMC method with Gibbs sampling to solve Eq. (10). Gibbs sampling generates a new sample of $X_{i,j}$ directly from its distribution conditioned on the labels of its neighbors $x_{i',j'}$ and $A_{i,j}$. In the MRF structure, the update is achieved by calculating the probability for each of the possible labels $n \in \{1, \cdots, 6\}$ at $(i,j)$ using Eq. (20) and randomly selecting from this distribution, see Fig. 6.

The velocity map $A$ is initialized with the superimposed HR and LR velocity maps,
Algorithm 1: MCMC Method for MRF

1. Input: $A_{LR}$ and $A_{HR}$

2. Initialize the velocity model $A$ by superimposing $A_{HR}$ over $A_{LR}$
   Initialize $X$, $\mu_n$ and $\sigma_n$ with GMM clustering

3. for each EM iteration do
   4. Construct PIPGM
   5. for $t = 1$ to max iteration $T$ do
      6. (E-Step) Gibbs Sampling
      7. for each pixel $(i, j) = (1, 1)$ to the maximum index $(I, J)$ do
         8. $X_{i,j}^{(t+1)} \sim P\left(X_{i,j} \mid X_{i,1}^{(t+1)}, \ldots, X_{i,j-1}^{(t+1)}, A_{i,j}, X_{i,j+1}^{(t)}, \ldots, X_{i,j}^{(t)}\right)$
      9. end for
     10. for each pixel $(i, j) = (1, 1)$ to the maximum index $(I, J)$ do
         11. $A_{i,j} \sim \sum_{n=1}^{6} P(X_{i,j} = n)N(\mu_n, \sigma_n^2)$
     12. end for
    13. (M-Step) Update Gaussian parameters $\mu_n$ and $\sigma_n^2$
       with the sample means and sample variances of $A^{(t+1)}$.
     14. end for
   15. end for

15. return $X$, $A$ (for each pixel)

see Fig. 8 (a1, b1), and the label map $X$ is initialized with a Gaussian mixture model clustering with the total cluster number $N = 6$ (will be discussed in Section 5.1), similar to Fig. 8 (a3, b3). All velocities with the label $n$ follow the same Gaussian distribution $N(\mu_n, \sigma_n^2)$. The expectation–maximization (EM) algorithm (McLachlan & Krishnan 2007), an iterative method to find MAP estimates of parameters, updates Gaussian parameters $\mu_n$ and $\sigma_n^2$. The termination criterion is either reaching 10,000 iterations or observing that the cumulative absolute difference across all pixels between consecutive iterations.
Figure 7. (a) LR model (Same as Fig 1a). (b) Same as Fig 1c. (c-d) Combined LR and HR models, smoothed by (c) 7×7 and (d) 3×3 average filters. (e) Synthetic stations (red 'X's) are deployed on the boundaries between HR and LR models for evaluation.

falls below 0.1 km/s, whichever is achieved first. The algorithm has been summarized in Algorithm 1.

5 EXPERIMENTS

To evaluate the efficacy of the PGM fusion method, we employ both a checkerboard model and the fault zone structure imaged from the 2019 Ridgecrest, CA, earthquake sequence. The Ridgecrest fault zone image consists of a shallow (representative of a depth of approximately 0.5 km) high-resolution Rayleigh wave model (Zhou et al. 2022), from which the S-wave velocity is roughly approximated by dividing by 0.9.

Fig. 7 shows the smoothed results with different smoothing levels. The aggressive, simple smoothing policy in Fig. 7(c) (7×7 average filter) removes the sharp boundaries while the details are lost. The simple and mild smoothing in Fig. 7(d) (3×3 average filter) preserves the details together with the artificial boundaries. It is essential to achieve a trade-off between the two cases, and this is the aim of our PIPGM method. To quantify this trade-off via the travel time between the stations and their residuals, 36 synthetic
Figure 8. (top, 'a' panels) Checkerboard and (bottom, 'b' panels) Ridgecrest models. (1) Superimposed HR and LR models. (2) Station location and ray density. (3) 6-class label mask maps for HR models (pixels with the same label are learned together). (4) Smoothing results with a 5×5 Gaussian filter (GF). (5) and (6) Fusion results with dictionary learning (DL) and with our PIPGM method, respectively.

sensors (red 'X', 10 on each edge) are on the border between the LR and HR areas. These residuals are then used to evaluate how much detailed information is preserved in the fused velocity model, compared to the HR maps.
Figure 9. (a) Direct superposition of the HR 1 Hz Rayleigh wave tomography and CVM LR models for the Ridgecrest area. (b-d) Combined LR and HR models, smoothed by (b) 3rd-order, (c) 2nd-order, and (d) 1st-order MRFs. Generally, the higher the order is, the larger range of neighboring pixels will be taken into consideration, and this leads to a model with smoother local patterns.

5.1 Graphical Structure Order Test

In an MRF, the variables are represented as nodes in the graph, and their dependencies with the neighbors are decided by the truncation order in Eq. (3). We name the MRF with the neighboring system truncated at n-th order as 'n-th order MRF' for brevity. A 1st-order neighboring system structure is a subset of nodes in which every two distinct nodes are directly adjacent. Usually, the order of neighboring systems has a significant effect on the smoothing results of the model. If an MRF model is based on a first-order neighboring system, it can account for interactions between immediate neighbors. In the context of image smoothing, it results in the enforcement of local smoothness, where each pixel is encouraged to be similar to its immediate neighbors. When a model involves a higher-order neighboring system (the Manhattan distance of the pixels in this set from
the centering pixel may be larger than 1), it can capture more complex relations and dependencies among variables, and higher-order MRFs can enforce smoothness over larger regions, allowing the model to preserve the high-level structures (usually related to the low-frequency patterns) and avoid sharp boundaries or noisy details.

Since information from a wider range of neighbors is taken into consideration, higher-order MRFs can be computationally more intensive in one iteration and may reach the optimum with fewer learning iterations. On the other hand, first-order MRFs are simple and efficient, and they can sometimes preserve some noisy details. The choice between first-order and higher-order MRFs depends on the specific requirements of the task and the trade-off between model complexity and computational efficiency. In our seismic tomography fusion problem, we demonstrate the fused models with 1st-, 2nd-, and 3rd-order MRF structures in Fig. 9. Considering that the Ridgecrest model measures the structure from a relatively limited region and that we would like to keep the rich detailed structures from the HR model, we choose the 1st-order MRF neighboring structure in our following comparison experiments.

5.2 Comparison with Conventional Methods

To demonstrate the advantage of the proposed PIPGM model, we compare its performance with some commonly-used conventional methods on both the synthetic checkerboard model and the real-data Ridgecrest model. First, we would like to briefly introduce several popular conventional fusion methods.

5.2.1 Gaussian Smoothing Filter

Gaussian smoothing filter is a data processing technique used to reduce noise and smooth out signals or data distributions. This method involves applying a Gaussian kernel function to the data, which is a bell-shaped curve that weights the data points based on their distance from a central point. The Gaussian kernel is defined by a smoothing parameter called the bandwidth or standard deviation, which controls the amount of smoothing applied to the data. A higher bandwidth leads to a wider, smoother curve, while a lower
bandwidth results in a narrower, more detailed curve. Gaussian smoothing filter is commonly used in image processing, signal processing, and data analysis applications. It is a powerful technique that can effectively remove noise and improve the clarity of data, but it can also introduce bias and distortions in the data if the smoothing parameter is not chosen carefully.

5.2.2 Dictionary Learning Smoothing

Dictionary Learning Smoothing (Yang et al. 2012) is a data processing technique used to de-noise and smooth misaligned patterns in the signals or images. It involves representing the data as a sparse linear combination of a set of basis vectors or atoms, which is learned from the data itself through an iterative process called dictionary learning. In this process, the algorithm seeks to find a set of basis vectors that can accurately represent the data with minimal error. Once the dictionary is learned, it can be used to transform the data into a sparse representation, where most of the coefficients are zero. This sparse representation can then be used to de-noise and smooth the data by selectively removing or modifying the coefficients corresponding to noise or unwanted features. Dictionary Learning Smoothing has been applied to a wide range of data processing applications, including image processing, audio processing, and signal processing. It is a powerful technique that can effectively remove noise and preserve the underlying structure of the data, but it requires a large amount of training data and can be computationally expensive.

5.3 Results Comparison

We show the fusion of HR and LR components for directly-superimposed checkerboard and Ridgecrest velocity models in Fig. 8 (a1, b1), which both have an HR region in the center, surrounded by LR velocities in the surrounding areas. The checkerboard model has 40×40 pixels in the 100×100 km LR area and 40×40 pixels in the 40×40 km HR area. The fused model has 100×100 pixels in the 100×100 km LR area. Similarly, the Ridgecrest model has 50×50 pixels in a 100×100 km LR area and 192×224 pixels in a 58×64 km HR area. The fused model has 330×350 pixels in a 100×100 km area. Fig. 8 (a2, b2) shows
Table 1. Evaluation results for Gaussian Filtering (GF), dictionary learning (DL), classical probability graphical model (PGM), and physics-informed probability graphical model (PIPGM) for both checkerboard and the Ridgecrest model. Evaluation metrics are travel time root-mean-square error (RMSE), naturalness image quality evaluator (NIQE), peak signal-to-noise ratio (PSNR), and Fréchet inception distance (FID). ↓ indicates smaller is better, and ↑ opposite.

<table>
<thead>
<tr>
<th></th>
<th>RMSE/s↓</th>
<th>NIQE↓</th>
<th>PSNR/dB↑</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkerboard</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GF</td>
<td>1.65</td>
<td>7.68</td>
<td>14.58</td>
<td>45.75</td>
</tr>
<tr>
<td>DL</td>
<td>1.18</td>
<td>5.44</td>
<td>15.70</td>
<td>33.85</td>
</tr>
<tr>
<td>PGM</td>
<td>1.14</td>
<td>5.40</td>
<td>16.14</td>
<td>32.49</td>
</tr>
<tr>
<td>PIPGM</td>
<td>1.06</td>
<td>5.41</td>
<td>16.14</td>
<td>32.46</td>
</tr>
<tr>
<td>Ridgecrest</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GF</td>
<td>3.52</td>
<td>12.41</td>
<td>21.80</td>
<td>61.39</td>
</tr>
<tr>
<td>DL</td>
<td>2.61</td>
<td>7.29</td>
<td>22.36</td>
<td>54.25</td>
</tr>
<tr>
<td>PGM</td>
<td>2.27</td>
<td>6.70</td>
<td>23.04</td>
<td>47.49</td>
</tr>
<tr>
<td>PIPGM</td>
<td>2.17</td>
<td>6.59</td>
<td>23.16</td>
<td>47.18</td>
</tr>
</tbody>
</table>

the checkerboard and Ridgecrest station settings along with the ray-path density. For the checkerboard model, the stations are evenly distributed, whereas the stations for the Ridgecrest model are highly irregular, reflecting the pattern used in Zhou et al. (2022). Fig. 8 (a3, b3) shows the label mask maps from the last iteration of the PIPGM models. Pixels covered with the same label indicate that these areas potentially share similar velocity patterns and are sampled from the same distribution. The smoothed fusion results with the 5×5 Gaussian smoothing filter (GF), dictionary learning (DL) (Yang et al. 2012), and our proposed physics-informed probability graphical model (PIPGM) are shown in Fig. 8 (a4-a6) and (b4-b6). The results show that the learning methods (DL and PIPGM) preserve more detailed information than direct Gaussian smoothing. This is because the learning methods adaptively find the fusion parameters which optimize the accuracy of the representation, while Gaussian smoothing only combines the neighboring pixels with a predefined kernel.
5.4 Numerical Metrics Test

We use multiple metrics to evaluate the efficacy of our model fusion: travel time Root-Mean-Squared-Error (RMSE, which measures information lost after model fusion (Bianco et al. 2019)), Naturalness Image Quality Evaluator (NIQE, a common-used measurement for image quality (Mittal et al. 2012)), Peak Signal-to-Noise Ratio (PSNR, measuring the sharpness of images (Poobathy & Chezian 2014)), and the Frechet inception distance (FID, capturing similarities between the original and fusion models (Chong & Forsyth 2020)) in Table 1. In the checkerboard test, because the pattern is simple and the stations are evenly distributed, all the learning methods achieve similar performance. For the realistic Ridgecrest model, however, the PGMs outperform the DL model, as the latter is sensitive to the orientation of the patches while the graphical models are rotationally invariant. We observe further improvements after the ray density information is informed in PIPGM.

Geological formations are often anisotropic, meaning their properties vary depending on the direction in which they are measured. For example, formations are often laterally continuous and vertically stratified. Standard Markov Random Field (MRF) schemes, which assume homogeneous properties (same properties in all directions), can lead to errors when applied to these formations. PIPGM, on the other hand, considers the anisotropic nature of geological formations, leading to more accurate results. Seismic inversion is an ill-posed problem, meaning it doesn’t have a unique solution, and small changes in the input can lead to large changes in the output. Regularization is a technique used to stabilize the solution. Our PIGPM provides an edge-preserving regularization based on the information neighboring pixels, which is effective for reconstructing subsurface models.

6 CONCLUSIONS

We have developed a novel approach to fuse multiresolution seismic tomography maps with probability graphical models (PGMs). Our physics-informed PGM tomography model
fusion method achieves a balance between smoothing the generally undesired sharp boundaries between LR and HR components and preserving the detailed information from the HR models.

We tested the efficacy of our proposed fusion method using a checkerboard model and a realistic, complex fault zone model around the 2019 M7.1 Ridgecrest earthquakes. The tests for the checkerboard model, which is relatively simple with the stations evenly distributed, show that our PIPGM outperforms all the baseline methods. All the learning-based techniques used to combine the maps (including the proposed PGM and PIPGM methods) significantly outperform the conventional methods, since the parameters are adaptively learned from the pairwise data. In the Ridgecrest model, our PIPGM methodology demonstrates a 38% reduction in travel time residuals when compared to Gaussian kernel smoothing. This result is likely due to the poor performance of traditional techniques in handling the nonuniform-distributed data evenly, while our PIPGM model can adaptively choose the weights according to the intricate structure of the Ridgecrest model.

In summary, our proposed PIPGM methodology significantly outperforms traditional techniques in the task of integrating maps of varying resolution, particularly when the map complexity escalates and the distribution of data points is uneven. Our advanced methodology presents potential benefits for augmenting our comprehension of the Earth’s structure and holds promise for enhancing outcomes in other seismic research applications, such as earthquake ground motion prediction.
7 ACKNOWLEDGEMENT

This research was supported by the Southern California Earthquake Center (SCEC Contribution Number 21133). The study is supported by SCEC based on Award DE-SC0016520 from the U. S. Department of Energy. The seismic and station data used in this paper are accessed from the FDSN 3J:RAMP deployment of 3C nodal for July Searles Valley 2019 Earthquake (doi: 10.7914/SN/3J_2019). The low-resolution velocity model can be accessed from SCEC CVM version S4.26-M01 (doi: 10.1002/2014JB011346).
REFERENCES


Zheng Zhou et al.


Zhang, H. & Ben-Zion, Y., 2023. Enhancing regional seismic velocity model with higher-resolution local results using sparse dictionary learning.


APPENDIX A: GAUSSIAN MIXTURE MODEL

Gaussian Mixture Clustering is a widely-used probabilistic clustering technique based on the Gaussian Mixture Model (GMM). In this model, the data points are assumed to be generated from a mixture of a finite number of Gaussian distributions with unknown parameters (Shahrabi et al. 2016). Clustering aims to identify these parameters, thus segmenting the data into different clusters, each represented by a Gaussian distribution.

The Gaussian Mixture Model (GMM) clustering algorithm can be implemented using the Expectation-Maximization (EM) method. Here, the data is denoted by \( \mathbf{a} \) and the corresponding labels by \( \mathbf{x} \). The algorithm is described as follows:

(i) **Initialization:**
Select the number of clusters \( N = 6 \). Initialize the means \( \mu_n \), covariance matrices \( \Sigma_n \), and mixing coefficients \( \pi_n \) for all \( n \) clusters. The initialization can be done randomly or based on some prior knowledge.

(ii) **Expectation Step (E-Step):**
For each data point \( a_i \), calculate the responsibility \( p(x_i) \), which represents the probability that \( a_i \) belongs to cluster \( n \):

\[
p(x_i) = \frac{\pi_n \mathcal{N}(a_i | \mu_n, \Sigma_n)}{\sum_{n=1}^{N} \pi_n \mathcal{N}(a_i | \mu_n, \Sigma_n)},
\]

where \( \mathcal{N}(a_i | \mu_n, \Sigma_n) \) is the likelihood of \( a_i \) under the Gaussian distribution parameterized by \( \mu_n \) and \( \Sigma_n \).

(iii) **Maximization Step (M-Step):**
Update the parameters \( \mu_n, \Sigma_n, \) and \( \pi_n \) to maximize the log-likelihood of the observed data:

\[
\mu_n^{\text{new}} = \frac{1}{M_n} \sum_{i=1}^{M} p(x_i) a_i,
\]
\[
\Sigma_n^{\text{new}} = \frac{1}{M_n} \sum_{i=1}^{M} p(x_i) (a_i - \mu_n)(a_i - \mu_n)^T,
\]
\[
\pi_n^{\text{new}} = \frac{M_n}{M},
\]
where $M_n$ is the effective number of points assigned to cluster $n$, and $M$ is the total number of data points.

(iv) **Update after Convergence:**
If the change in parameters or the log-likelihood between iterations is less than a pre-defined threshold, or if a maximum number of iterations is reached, stop the algorithm. Otherwise, return to the E-step. Finally, assign each data point $a_i$ to the cluster $n$ that maximizes $p(x_i)$, i.e., $x_i = \arg\max_n p(x_i)$.

The Expectation-Maximization method guarantees convergence to a local maximum of the log-likelihood, but the solution may not be globally optimal. Thus, the algorithm can be run multiple times with different initializations to improve the final result.

**APPENDIX B: CLUSTER NUMBER TEST**

The number of clusters in GMM clustering has a significant impact on the results. Generally, the number of clusters can influence the complexity of the model and the interpretability of the results. More clusters result in a more complex model, which can capture intricate data structures better and lead to more detailed insights into the data. However, it also increases the risk of overfitting.

Selecting the optimal number of clusters is crucial in GMM and other clustering techniques. Several methods help determine an appropriate number of clusters, such as the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and the silhouette score. These methods balance the trade-off between the goodness-of-fit of the model and the complexity of the model.

For this experiment, we tested and compared the cluster number sequences $N = 3, 5, 6, 7, 9$, which is prevalent in practical uses of MRFs. Figure B1 shows that the larger the number of clusters, the more detailed information is preserved in the HR models, and the larger the computation recourse it needs. This implies that a trade-off between cost and performance needs to be achieved. Figure B2 demonstrates the number of clusters verse RMSE error (left vertical axis) and computation time (right vertical axis) in the Ridgecrest
Figure B1. Comparison of the fusion results using cluster numbers n = 3, 5, 6, 7, and 9. (top row) fused velocity model, and (bottom row) cluster distribution. Generally, the larger the number of clusters, the more detailed information is preserved in the HR models, and the larger the computation recourse it needs. A trade-off is required to balance the cost and the performance. The RMSE error experiences a significant decrease while the cluster number grows from 3 to 6, with limited decrease for larger cluster numbers, and there is a rapid growth in run time when the cluster number exceeds 6. For these reasons, we empirically choose the cluster number as 6 to achieve a trade-off between the model performance and the computation complexity.

Figure B2. The number of clusters verse RMSE error (left vertical axis, corresponding to the blue solid line) and run time (right vertical axis, corresponding to the orange dashed line) on the Ridgecrest model. Generally, the larger the cluster number is, the smaller the RMSE error and the longer the running time are. We choose cluster number as 6 to balance the model performance and the computation complexity.