Statistica Sinica Preprint No: SS-2025-0120	
Title	Estimation of Piecewise Continuous Regression Function in Finite Dimension using Oblique-axis Regression Tree with Applications in Image Denoising
Manuscript ID	SS-2025-0120
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0120
Complete List of Authors	Subhasish Basak, Anik Roy and Partha Sarathi Mukherjee
Corresponding Authors	Partha Sarathi Mukherjee
E-mails	psmukherjee.statistics@gmail.com
Notice: Accepted author version.	

Statistica Sinica

Estimation of Piecewise Continuous Regression Function in Finite Dimension using Oblique-axis Regression Tree with Applications in Image Denoising

Subhasish Basak[1], Anik Roy[2], Partha Sarathi Mukherjee[1*]

[1]*Indian Statistical Institute, Kolkata, India*
[2]*Emory University, Atlanta, USA*

*Corresponding Author

Abstract: Decision trees are one of the most widely used nonparametric methods for regression and classification. In existing literature, decision tree-based methods have been used for estimating continuous functions or piecewise-constant functions. However, they are not flexible enough to estimate the complex shapes of jump location curves (JLCs) in two-dimensional regression functions. In this article, we explore the Oblique-axis Regression Tree (ORT) and propose a method to efficiently estimate piece-wise continuous functions in a general finite dimension with fixed design points. The central idea involves clustering the local pixel intensities by recursive tree partitioning and using the local leaf-only averaging for estimation of the regression function at a given pixel. The proposed method can preserve complex shapes of the JLCs well in a finite-dimensional regression function. Due to a different set of assumptions on the underlying regression function, the overall framework of the proofs is different from what is available

in the literature on regression trees. Theoretical analysis and numerical results, particularly on image denoising, indicate that the proposed method effectively preserves complicated edge structures while efficiently removing noise from piecewise continuous regression surfaces.

Key words and phrases: CART, Image denoising, Jump location curves, Oblique regression trees, p-dimension, Local-leaf-only-averaging.

1. Introduction

Tree-based methods hold a pivotal place in modern statistics and machine learning. Classification and Regression Trees (CART) [Breiman (1984)] are recognized as very powerful and widely used tools among supervised learning algorithms. Over time, ensemble learning techniques that combine multiple trees, such as Random Forest (RF) [Breiman (2001)], bagging, and boosting, have become increasingly popular for classification and regression applications. Decision trees are the building blocks of all these algorithms. The popularity of the techniques based on decision trees is attributed to their intuitive construction and appealing interpretability. However, it has been noticed from the time when CART was first proposed using marginal variables as splitting rules can pose challenges in preserving complex structures of the data while smoothing. This issue can be well addressed by taking a linear combination of the predictors as the splitting rule instead

of relying on marginal variables. This procedure is popularly known as the oblique regression tree (ORT) algorithm. In literature, various versions of the ORT algorithms [Cattaneo et al. (2024)] are available. However, nearly all of the existing methods assume that the true functional relationship is continuous in nature. However, in practice, there are many situations where the true regression function is clearly discontinuous. Such situations can be recognized when the response variable changes abruptly with small variations in predictor variables. For example, in materials science, rapid phase changes are common during phase transitions. In geostatistics, sensing data from rock strata often exhibits abrupt changes near sediment layers. Similarly, in image analysis, the intensity function of an image changes abruptly around object boundaries or edges. Recently, due to easy availability of various imaging modalities, image monitoring or surveillance has received considerable attention in manufacturing industries, medical science, earth surface surveillance, and so forth. Image denoising is a pivotal pre-processing step in most applications of image monitoring [Roy and Mukherjee (2024, 2025)]. As a result, jump-preserving smoothing of regression functions is an extremely important problem with numerous applications across engineering and scientific fields. In this paper, we introduce a nonparametric smoothing technique based on ORT, designed for scenarios where the true

regression function is piecewise continuous. Given that a grayscale image intensity function can be represented as a piecewise continuous regression function, we demonstrate the application of the proposed algorithm on the problem of grayscale image denoising in the latter part of the paper.

1.1 Literature review on related smoothing methods

So far in the literature, there has been a very limited discussion on the estimation of discontinuous regression functions. Proposed by Breiman (1984), CART is one of the most popular nonparametric regression methods that fits the piecewise constant function quite well. However, it fails to capture the complex shape, even if it forms a tree with high depth. A similar nonparametric regression approach in the literature is dyadic regression tree (DRT), proposed by Donoho (1997). The central idea is to split the input space at the midpoint along a dimension. The major difference between traditional CART and the DRT is that it allows a recursive splitting at any point of the input space. Although it has computational advantages, and it fits piecewise constant functions reasonably well, it faces similar issues when we are interested in accommodating complex shape boundaries (Chaudhuri and Chatterjee, 2023). Recently, ORT-based piecewise continuous function estimation and its theoretical properties have become an

active research area (Cattaneo et al., 2024). ORT-based decision tree is a natural extension of CART and DRT to accommodate the boundaries of the complex shape. Nowadays, ORT-based decision trees have received remarkable attention among the research communities. Zhan et al. (2024) shows the consistency of the ORT-based algorithm under the assumption that the underlying regression function is continuous. Additionally, several studies in the Gaussian Process (GP) literature are also useful for estimating piecewise continuous regression functions. These methods involve partitioning the input domain into multiple regions and modeling the data in each region with a separate GP model. A Bayesian tree-based GP, proposed by Gramacy and Lee (2008), uses a dyadic treed partitioning to split the input domain along axis-aligned directions, fitting a stationary GP model to the data in each tree leaf. Similarly, Konomi et al. (2014) use a Bayesian CART tree to partition the input domain. However, one major limitation of these piecewise GP models is that their partitioning schemes, such as axis-aligned partitions or triangulations, are too restrictive to capture the complex, curvy boundaries often present in various real-world images.

Since 2D grayscale image surfaces are discontinuous in nature, techniques for estimating piecewise-continuous functions have direct applications in image denoising. A thorough analysis of the history of image de-

noising can be found in Gonzalez and Woods (2018). Most of the image denoising methods available in current literature can be grossly classified into two types: (i) Some methods denoise without explicitly estimating underlying edge-structure, whereas (ii) other methods estimate these structures first, and then proceed to denoise. Qiu (2005) provides an elaborated overview of an approach known as Jump Regression Analysis (JRA). 2-D jump regression analysis (Qiu, 2005) estimates the regression surface from the noisy data, while preserving the boundary of the image object. Qiu (2009) proposes an edge-preserving image denoising method which splits each neighborhood into two different parts using edge information and uses the best fitting estimates on these regions. However, explicit edge detection-based denoising methods often provide poor results on images with low resolution. To overcome this, Mukherjee and Qiu (2011) develop a clustering-based approach where local neighborhoods around each pixel are split into two clusters using intensity values. There are many other approaches that do not detect the edge structures explicitly. Methods such as bilateral filtering proposed by Chu et al. (1998) calculate weights using underlying edge information to calculate local averages for smoothing.

1.2 Contributions of this paper

As previously discussed, existing literature on JRA offers useful tools for estimating discontinuous functional relationships. However, most of the methods focus on cases with only one or two predictor variables and are not easily extendable to scenarios with multiple predictors. Recently, Kang et al. (2021) propose a jump regression model that accommodates multiple predictors, but their approach assumes an additive structure in the functional relationship, effectively disregarding interaction terms among the predictors. To elucidate this limitation, consider the model: $w = f(z_1, z_2) + \varepsilon$ with two predictor variables z_1 and z_2. Now, the additive model i.e., $f(z_1, z_2) = f_0 + f_1(z_1) + f_2(z_2)$ is suitable for preserving the jump location curves parallel to X-axis and Y-axis. However, this model is not flexible enough to describe and preserve the jump location curves in other directions. The proposed jump regression model overcomes such limitations of the JRA literature mentioned above. The proposed algorithm does not require pruning for estimating the underlying function; however, we use some constraints to limit the depth of the tree, which may lead to significant computational advantages. Many of the recent tree-based algorithms, e.g., Zhan et al. (2024), assume the underlying function can be expressed as a *ridge* function. In practice, it is often not a valid assumption. For

instance, the image intensity functions are discontinuous in nature and not necessarily a ridge function. In this paper, we do not make any such strong assumptions. Instead, we impose certain assumptions on the underlying discontinuous points to address this issue. The proofs of the theoretical properties of the proposed tree-based method are much simpler, and the overall framework of the proofs is substantially different from what is available in the literature on regression trees. This is due to a different set of assumptions on the underlying regression function. Theoretical properties and their proofs, thus, demonstrate the novelty of this paper, in addition to excellent numerical results.

1.3 Organization of this paper

The remainder of this paper is organized as follows. Section 2 describes the proposed methodology. We study its statistical properties in Section 3. Section 4 shows one major application of the proposed method in the form of image denoising. Numerical performance of the proposed algorithm is shown in Section 5. A few remarks in Section 6 conclude this paper.

2. Proposed Methodology

In this paper, we work with the standard regression framework, and the statistical model can be expressed as

$$w(\boldsymbol{z}) = f(\boldsymbol{z}) + \varepsilon, \tag{2.1}$$

where w is real-valued response or output variable, $\boldsymbol{z} = (z_1, z_2, \ldots z_p)$ is the predictor (input, feature, or covariate vector) variable which lie on an equally space lattice at the points $\left(\frac{i_1}{n}, \ldots, \frac{i_p}{n}\right)$, where each $i_j \in 1, 2, \ldots, n$ on the design space $\Omega = [0,1]^p$ and, ε is the error variable. We have a dataset $\mathcal{D}_n = \{(w_i, \boldsymbol{z_i}) : 1 \leq i \leq n^p\}$, consisting of independent samples drawn from the model in (2.1). Additionally, we assume that f is *piecewise continuous*, and has the following form:

$$f(\boldsymbol{z}) = \sum_{l=1}^{P} g_l(\boldsymbol{z}) \mathbb{I}_{\Gamma_l}(\boldsymbol{z}), \tag{2.2}$$

where $g_l(\boldsymbol{z}) = g(\boldsymbol{z}) + \delta_l$, and δ_l is the jump size in Γ_l. In literature, this is popularly known as jump regression analysis (JRA). Note that, $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_P\}$ is a finite partition of Ω such that: (i) Each Γ_l is a connected region in the design space. (ii) Defining $\partial\Gamma_l$ as the set of boundary points in Γ_l, $f(\boldsymbol{z})$ is continuous in $\Gamma_l \backslash \partial\Gamma_l$, for $l = 1, 2, \ldots, P$. (iii) $\bigcup_{i=1}^{P} \Gamma_l =$

Ω. (iv) There exists at most finitely many points $\{\boldsymbol{z}^*, k = 1, 2, \ldots, K^*\}$ in $[\bigcup_{i=1}^{n} \partial\Gamma] \bigcap \Omega$ such that for each $\boldsymbol{z}^*$ with $k = 1, 2, \ldots K^*$, there are $\Gamma_{k_1}^*, \Gamma_{k_2}^* \in \{\Gamma_l, l = 1, 2, \ldots P\}$ satisfying

(a) $\boldsymbol{z}^* \in \Gamma_{k_1}^* \bigcap \Gamma_{k_2}^*$, and

(b) $\lim\limits_{\substack{\boldsymbol{z} \to \boldsymbol{z}^* \\ \boldsymbol{z} \in \Gamma_{k_1}^*}} f(\boldsymbol{z}) = \lim\limits_{\substack{\boldsymbol{z} \to \boldsymbol{z}^* \\ \boldsymbol{z} \in \Gamma_{k_2}^*}} f(\boldsymbol{z})$.

Condition (a) implies that the point lies on multiple JLCs, whereas condition (b) implies that the limiting value across two partitions created by the JLCs match. In the imaging context, this means that two different partitions of the image are leaking into each other through a point on the relevant JLC. Such continuity points are sporadic, and should be treated differently from the usual continuity points. Our primary objective is to develop a smoothing technique that effectively preserves the jump location curves (JLCs) in the regression surfaces. To achieve this, we partition $\mathcal{D}_n$ in such a manner that each partition contains observations from one side of a JLC. To perform this, we construct a tree using the ORT algorithm, where each leaf node of the tree indicates one partition of the data and contains observations from one side of a JLC. To this end, each partition yields an estimator $\widehat{f}_n$ of f in model (2.2) via *leaf-only* averaging.

2.1 Recursive partitioning of the data

In this subsection, we introduce the algorithm by which we construct the recursive ORT, a pivotal step in the context of the proposed jump-preserving surface-smoothing technique. Note that we consider partitioning with respect to hyperplanes only. This restriction is imposed to ensure that the partitions remain convex, which is necessary for the subsequent proofs. However, if we could extend the covariates or pixel coordinates by including nonlinear functions of them, and correspondingly extend the data in a meaningful way, then the results developed here could potentially be applied in that setting to achieve improved performance.

Let $\mathcal{N}$ denotes a node of the ORT which is nothing but a subset of $[0,1]^p$, and consider its two children $\mathcal{N}_L = \{\boldsymbol{z} : \boldsymbol{\alpha}^T\boldsymbol{z} \leq c\}$ and $\mathcal{N}_R = \{\boldsymbol{z} : \boldsymbol{\alpha}^T\boldsymbol{z} > c\}$. Note that $\mathcal{N}_L$ and $\mathcal{N}_R$ cannot be empty, and satisfy $\mathcal{N}_L \cup \mathcal{N}_R = \mathcal{N}$. The central idea behind the tree construction is hierarchically partitioning the data in a greedy manner through a recursive binary splitting rule. Similar to the CART, a daughter node is treated as a parent node in the next step. Then, a parent node $\mathcal{N}$ at any depth is divided into two daughter nodes $\mathcal{N}_L$ and $\mathcal{N}_R$ by maximizing the impurity gain

$$\widehat{\Delta}(\mathcal{N}, \boldsymbol{\alpha}, c) = \frac{1}{|\mathcal{N}|}\Big[\sum_{\boldsymbol{z}\in\mathcal{N}}\big(w(\boldsymbol{z})-\bar{w}\big)^2 - \sum_{\boldsymbol{z}\in\mathcal{N}_L}\big(w(\boldsymbol{z})-\bar{w}_L\big)^2 - \sum_{\boldsymbol{z}\in\mathcal{N}_R}\big(w(\boldsymbol{z})-\bar{w}_R\big)^2\Big] \tag{2.3}$$

with respect to $(\boldsymbol{\alpha}, c)$. Here, $\bar{w}$, $\bar{w}_L$, and $\bar{w}_R$ are defined as the sample averages of the response variable w_i at the node $\mathcal{N}$, $\mathcal{N}_L$, and $\mathcal{N}_R$, respectively. The impurity gain is nothing but the decrease in the error sum of square (SSE) by increasing the depth of the tree by one step.

Suppose $(\widehat{\boldsymbol{\alpha}}, \widehat{c})$ maximizes the equation (2.3). We then split the node $\mathcal{N}$ if

$$\widehat{\Delta}(\mathcal{N}) = \widehat{\Delta}(\mathcal{N}, \widehat{\boldsymbol{\alpha}}, \widehat{c}) > r_n,$$

where r_n is the splitting threshold for the recursive tree partitioning. Since there is no pruning step in the proposed algorithm, we choose r_n in such a manner that ensures a shallow tree construction. To this end, refinement of $\mathcal{N}$ produces two daughter nodes $\mathcal{N}_L = \{\boldsymbol{z} : \widehat{\boldsymbol{\alpha}}^T\boldsymbol{z} \le \widehat{c}\}$ and $\mathcal{N}_R = \{\boldsymbol{z} : \widehat{\boldsymbol{\alpha}}^T\boldsymbol{z} > \widehat{c}\}$. These child nodes act as a parent node for the next level of the tree construction, and further recursive refinement carries forward in the similar fashion. At any level of the tree refinement, a node is considered a leaf node if $\widehat{\Delta}(\mathcal{N}, \boldsymbol{\alpha}, c) \le r_n$, and we stop the recursion when all nodes become leaf nodes. Finally, we have the leaf nodes that partition the design space, where each partition is nothing but a convex polytope in $[0,1]^p$. Note that the decision tree construction is a filtering step for the proposed

jump-preserving smoothing. The intuition behind this is to partition the data in a manner such that the data points on the same side of the JLCs always lie in the same estimated partition. Lemma 1 in Section 3 supports our intuition. A pseudo code outlining the tree construction method is presented in Algorithm 1.

Selection of r_n for numerical implementation: We choose r_n to be small enough to ensure that there were sufficient partitions to adequately represent the entire image. Note that if we choose this parameter too small, then it will only increase computation time without any improvement in the performance of the estimate. Moreover, r_n should be proportional to the variance of the noise in the image, as that will cause the loss function we used to be proportionally larger. Based on numerical simulations, we choose $r_n = 0.0001\sigma^2$ for all numerical implementations.

2.2 Proposed jump preserving surface estimator

Using the proposed algorithm, we divide the design space $[0,1]^p$ in leaf nodes, say $\{\mathcal{L}_n^i | 1 \leq i \leq K_n\}$, where K_n denotes the number of leaf nodes, and $\bigcup_{1\leq i\leq K_n} \mathcal{L}_n^i = [0,1]^p$. For a given point $x \in [0,1]^p$, let $\mathcal{L}_n^{\boldsymbol{x}}$ denote the leaf

node where it lies. Then, the proposed estimator is given by:

$$\widehat{f}_n(\boldsymbol{x}) = \frac{\sum_{z \in \mathcal{L}_n^{\boldsymbol{x}} \bigcap \mathcal{D}_n} w(\boldsymbol{z})}{|\mathcal{L}_n^{\boldsymbol{x}} \bigcap \mathcal{D}_n|}. \tag{2.4}$$

From now on, we denote $\mathcal{L}_n \bigcap \mathcal{D}_n$ by $\mathcal{L}_n$ for simplicity. Since the observations in $\mathcal{L}$ lie on one side of a JLC, the proposed estimator preserves the discontinuity well. In the application of image denoising, we slightly modify the proposed estimator. Refer to Eqn. 4.7 for details.

Algorithm 1 Recursive Tree Construction

Input: Given dataset $\mathcal{D}_n$ and r_n is the specified cutoff.
ℓ is the list of nodes and ℓ_1 is an empty list.
while ℓ is not empty, **do**
    Pick a node $\mathcal{N}$ from ℓ,
    Calculate $\tilde{\Delta}(\mathcal{N})$.
    if $\tilde{\Delta}(\mathcal{N}) \leq r_n$ **then**
        Remove $\mathcal{N}$ from ℓ
        Mark it as a leaf node by adding it to ℓ_1.
    else
        Calculate $(\widehat{\boldsymbol{\alpha}}_{opt}, \widehat{c}_{opt}) = \arg\max_{(\boldsymbol{\alpha}, c)} \widehat{\Delta}(\mathcal{N}, \boldsymbol{\alpha}, c)$
        Split $\mathcal{N}$ into $\mathcal{N}_R$ and $\mathcal{N}_L$ using the line $\{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{opt}}^{\boldsymbol{T}} \boldsymbol{z} = \widehat{c}\}$.
        Remove $\mathcal{N}$ from ℓ
        Add $\mathcal{N}_R$ and $\mathcal{N}_L$ to ℓ.
    end if
end while
Output: ℓ_1 is the list of leaf nodes.

3. Statistical Properties

In this section, we investigate certain statistical properties to substantiate our arguments in the proposed methodology. We derive novel theoretical results to establish the consistency of the oblique decision tree within the context of jump regression. Theorems 2 and 3 characterize necessary properties of leaf nodes that should be present in a well-chosen neighborhood for local non-parametric denoising. Lemma 1 demonstrates that the proposed algorithm correctly distinguishes the continuous regions from the JLCs. Finally, Theorems 5 and 6 ensure asymptotic consistency of $\widehat{f}_n$, and its rate of convergence, respectively. Before going into the details, we introduce relevant notations and assumptions below.

Notations & Assumptions: We define a sequence of trees by $\{\mathcal{T}_n\}$, and the sequence of leaf nodes of $\mathcal{T}_n$ by $\{\mathcal{L}_n\}$, where $\mathcal{L}_{(n+1)} \subset \mathcal{L}_n$. Each successive tree $\mathcal{T}_{n+1}$ serves as a refinement of $\mathcal{T}_n$; all internal nodes of $\mathcal{T}_{n+1}$ are already contained in $\mathcal{T}_n$, while $\mathcal{T}_{n+1}$ may include additional leaf nodes corresponding to finer resolution details. We define $|\mathcal{L}_n|$ as the number of data-points in $\mathcal{D}_n$ which are inside $\mathcal{L}_n$, and define μ as the Lebesgue measure on $[0,1]^p$. We now make the following assumptions:

A1: f is piecewise Lipschitz continuous.

A2: f is bounded. Without any loss of generality, we assume that it is bounded inside $[0,1]^p$.

A3: The data-points in $\overline{\Lambda_\ell}/int(\Lambda_\ell)$ are called jump points, and the set of jump points has Lebesgue measure zero.

A4: The set of jump location points on each JLC is a piecewise lower-dimensional hyperplane. For notational convenience, each such hyperplane is considered to be a different JLC.

A5: There exists at most countably many JLCs.

A6: The set of singular points has measure zero.

A7: The pointwise noise ε are independently distributed and follows $\mathbf{N}(0,\sigma^2)$.

A8: The splitting threshold r_n is chosen such that $\sum_n \sqrt{r_n} < \infty$.

Next, we state a series of important theoretical results regarding the proposed method. All results hold under the assumptions stated above, and proofs appear in the Supplementary Material. The first result simplifies the impurity gain measure.

Property 1.

$$\hat{\Delta}(\mathcal{N},\boldsymbol{\alpha},c) = \frac{|\mathcal{N}_L||\mathcal{N}_R|}{|\mathcal{N}|^2}\left(\bar{w}_L - \bar{w}_R\right)^2. \tag{3.5}$$

The following theorem states that the number of data points in $\mathcal{L}_n$ grows indefinitely as n increases. This result eventually ensures asymptotic consistency of $\widehat{f}_n$.

Theorem 2. *Under the assumptions A1-A8,* $|\mathcal{L}_n| \to \infty$ *almost surely.*

Remark 1. Observe that $\mu(\mathcal{L}_n)$ is non-increasing as $\mathcal{L}_{n+1}$ is a subset of $\mathcal{L}_n$. Since $\mu(\mathcal{L}_n)$ is non-increasing and non-negative, it must have a limit.

The following theorem and its corollaries state the results that we can conclude regarding f in both cases when the above limit is non-zero and zero.

Theorem 3. *Under the assumptions A1-A8, if* $\lim_{n\to\infty} \mu(\mathcal{L}_n) \neq 0$, *then* f *is a.e. constant in* $\bigcap_n \mathcal{L}_n$.

Corollary 1. *If* $\lim_{n\to\infty} \mu(\mathcal{L}_n) = 0$ *and* $\bigcap_n \mathcal{L}_n = \mathcal{L}$ *lies on a* k $(< p)$ *dimensional affine subspace of* $\mathbb{R}^p$ *with* $\mu_k(\mathcal{L}) \neq 0$, *then one of these is true.*

(i) $\mathcal{L}$ *contains an entire JLC.*

(ii) $\mathcal{L}$ *is contained entirely inside a JLC.*

(iii) f *is a.e. constant on* $\mathcal{L}$.

Using $k = 1$ in the above corollary, we get

Corollary 2. *Let* $\gamma(A)$ *denote the maximum distance between two points in the set* A. *If* $\lim_{n\to\infty} \mu(\mathcal{L}_n) = 0$ *and* $\lim_{n\to\infty} \gamma(\mathcal{L}_n) \neq 0$, *then one of the following is true:*

(i) $\bigcap_n \mathcal{L}_n$ *contains an entire JLC.*

(ii) $\bigcap_n \mathcal{L}_n$ *is contained entirely inside a JLC.*

(iii) f *is a.e. constant on* $\bigcap_n \mathcal{L}_n$.

Property 4. If $\widehat{\Gamma} = \bigcup_x \Big\{\mathcal{L}^{\boldsymbol{x}} | \mathcal{L}^{\boldsymbol{x}}$ *contains an entire JLC and* $\mu(\mathcal{L}^{\boldsymbol{x}}) = 0\Big\}$, then $\mu(\widehat{\Gamma}) = 0$.

The above property holds because at most countably many sets of these unions are distinctly non-empty, as there are at most countably many JLCs. Next, we show that the average of all continuity points of f over a leaf node converges to the true value. Observe that if $\boldsymbol{x} \in \mathcal{L}_n$ is the point of interest and there is no JLC in $\mathcal{L}_n$, then as $\mu(\mathcal{L}_n) \to 0$, $\gamma(\mathcal{L}_n) \to 0$, and f is piecewise Lipschitz, the average of the observed intensity values over the leaf node should converge to $f(\boldsymbol{x})$. However, if $\mathcal{L}_n$ contains a JLC, we need to show that all the points on the leaf node are similar, and do not have any jumps between them, which brings us to the next result.

Lemma 1. *Let* $\boldsymbol{x}$ *be a non-singular continuity point of* f. *If* $\lim_{n\to\infty} \gamma(\mathcal{L}_n^{\boldsymbol{x}}) = 0$, *then* $\exists\ N$ *such that* $\forall\ m > N$, $\mathcal{L}_m^{\boldsymbol{x}}$ *does not contain any discontinuity point.*

Finally, the following theorem establishes asymptotic consistency of $\widehat{f}_n$.

Theorem 5. *Under the assumptions A1-A8,* $\lim_{n\to\infty} \widehat{f}_n(\boldsymbol{x}) = f(\boldsymbol{x})$ *a.e., almost surely.*

The next theorem provides a convergence rate of $\widehat{f}_n$ for points with non-zero derivatives.

Theorem 6. *Let $\boldsymbol{x}_0$ be a point such that f' is continuous at $\boldsymbol{x}_0$, and $|f'(\boldsymbol{x}_0)| > 2\delta$ where $\delta > 0$. This ensures the existence of a neighborhood $N_{\boldsymbol{x}_0}$ of $\boldsymbol{x}_0$ such that, for all $\boldsymbol{x} \in N_{\boldsymbol{x}_0}$, $|f'(\boldsymbol{x}) - f'(\boldsymbol{x}_0)| < \delta$ and $|f'(\boldsymbol{x})| > \delta$. Then, under the assumptions A1-A8, the proposed estimator $\widehat{f}_n(\boldsymbol{x})$ as in (2.4) converges to $f(\boldsymbol{x}_0)$ at the rate $o\left(\left(\sqrt{r_n \left|\log\left|\log r_n\right|\right|}\, C_{\chi^2_1}\right)^{1/4}\right)$, $C_{\chi^2_1}$ being a constant depending on a χ^2_1 random variable.*

4. An Application of the Proposed Method: Image Denoising

Image denoising is a fundamental problem in the field of image processing. It serves as a crucial pre-processing step in many applications to make further analysis meaningful and reliable. For example, in sequential image surveillance, denoising plays a central role. In addition to removing noise, a key requirement for effective denoising methods is the preservation of edge structures within the image. In the JRA literature [Qiu (2005)], a 2D grayscale image is expressed as a discontinuous regression surface, where the edges of objects in the image are treated as points of discontinuity or "jump points." In this section, we introduce a new image denoising method based on an oblique-axis regression tree, designed to preserve these jumps in

the regression surface. Although the algorithm is versatile and applicable to various types of images, we focus on 2D grayscale images here for simplicity. The proposed 2D grayscale image denoising technique directly applies the algorithm from Section 2 with $p = 2$. Consequently, the 2D JRA model can be described similarly as follows:

$$w_{ij} = f(x_i, y_j) + \varepsilon_{ij} \ , \quad \text{for} \quad i, j = 1, 2, \ldots, n, \tag{4.6}$$

where $\mathcal{D}_n = \{(x_i, y_j) : i, j = 1, 2, ..., n\}$ are equally spaced design points (or pixels) in the design space $\Omega = [0, 1] \times [0, 1]$, $f(x, y)$ is the unknown image intensity function at (x, y), and ε_{ij} are independent and identically distributed (i.i.d.) random errors with mean 0 and variance $\sigma^2 > 0$. To denoise an image, we perform recursive tree splitting until all nodes become leaf nodes, for a specified cut-off r_n depending on n. Using the tree partitioning method outlined in Algorithm 1, we partition the design space $[0, 1]^2$ in K_n number of convex polygons, represented as $\{\mathcal{L}_n^i | 1 \leq i \leq K_n\}$. The resulting estimator is obtained by averaging the image intensity values within the leaf nodes. However, in practice, large leaf nodes may cause the loss of intricate details in the image surface. To address this issue, we consider *leaf-wise local weighted* averaging instead of averaging over the whole leaf node. For this adjustment, we consider a square neighborhood at each

pixel (x, y) as $B(x, y; h_n)$ where $2h_n > 0$ is the length of each side of the square. The modified estimator at $f(x, y)$ is thus defined as:

$$\widehat{f}_n^M(x,y) = \frac{\sum_{(u_i,v_j)\in\mathcal{L}_n^{(x,y)}\bigcap B(x,y;h_n)} w(u_i,v_j)SS_{(x,y)}(u_i,v_j)}{\sum_{(u_i,v_j)\in\mathcal{L}_n^{(x,y)}\bigcap B(x,y;h_n)} SS_{(x,y)}(u_i,v_j)}. \tag{4.7}$$

Here, $\mathcal{L}_n^{(x,y)}$ represents the leaf node containing the test pixel coordinate (x, y), (u_i, v_j) denotes pixel coordinates within $\mathcal{L}_n^{(x,y)}$, and $SS_{(x,y)}(u_i, v_j)$ is a similarity score between the pixels (x, y) and (u_i, v_j) expressed as follows:

$$SS_{(x,y)}(u_i,v_j) = \exp\left(-\frac{\kappa_n||B(u_i,v_j;h_n) - B(x,y;h_n)||^2}{(nh_n)^2}\right).$$

In the above expression, the distance between the neighborhoods represents the L_2-distance between the pixel intensities. Regarding the choice of h_n, selecting a value too large could blur fine details, while a very small value may result in a noisy estimator $\widehat{f}_n^M(x, y)$. Based on numerical experience, we recommend choosing $h_n \in [\frac{2}{n}, \frac{4}{n}]$. Note that the proposed method is not very sensitive to the choices of h_n. For numerical implementation, we select $h_n = \frac{3}{n}$, and the scaling parameter $\kappa_n = 5$. This is based on checking with various values of h_n and κ_n on the various images. Figure 1 demonstrates that the proposed method can partition the pixels of a given

image appropriately, and thus can remove noise while preserving the edge details.

Figure 1: Demonstration of partitioning of the pixels in the test image. Images from left to right are the noisy image, the denoised image by the proposed method, and estimated partitions, respectively. The partitions are represented by different shades. Note that the estimated partitions are subsets of the true partitions.

5. Numerical Studies

In this section, we evaluate the performance of the proposed method through numerical analysis, comparing it with several state-of-the-art methods from the literature. Since jump preserving surface estimation is common in image analysis, we conduct our numerical experiments with various simulated images and real images. Including the noise removal, another major aspect in the context of image denoising is edge preservation. We demonstrate the performance of the proposed method both in terms of noise removal and edge preservation. To evaluate the noise removal performance of the proposed method in comparison with the state-of-the-art methods, we consider

root-expected mean square error (REMSE), defined as

$$REMSE(\widehat{f}, f) = \sqrt{\mathbb{E}\Big[\sum_{(u,v)\in\Omega}\Big(\widehat{f}(u,v) - f(u,v)\Big)^2\Big]}.$$

On the other hand, the performance of edge preserving can be quantified by a similarity measure between the sets of detected edge pixels of the denoised image and the true image. One such similarity measurement between the denoised image and the true image can be expressed as

$$d_{KQ}(\widehat{\Gamma}_{\widehat{f}}, \widehat{\Gamma}_f) = \frac{0.5}{|\widehat{\Gamma}_{\widehat{f}}|}\sum_{(u,v)\in\widehat{\Gamma}_{\widehat{f}}} d_E((u,v), \widehat{\Gamma}_f) + \frac{0.5}{|\widehat{\Gamma}_f|}\sum_{(u,v)\in\widehat{\Gamma}_f} d_E((u,v), \Gamma_{\widehat{f}}),$$

where $\widehat{\Gamma}_{\widehat{f}}$ and $\widehat{\Gamma}_f$ are the detected edge pixels for the denoised image and the true image, respectively. It is to be noted that d_{KQ} does not hold the properties of a metric. One popular metric to compute the distance between two point sets is *Hausdorff* distance. However, *Hausdorff* metric is very sensitive to individual points. Therefore, we select d_{KQ} to compute the similarity measurement between the point-sets of the detected edge pixels of the two images. For a good image denoising procedure, it is expected that both the values of REMSE and d_{KQ} would be small. Edge preservation can be assessed with any edge-detection method. In our numerical studies, we adopt the local linear kernel (LLK) approach available in the DRIP package

Kang and Qiu (2024) on CRAN-R, particularly due to its robustness in the presence of noise.

5.1 A brief description of the competing methods

To evaluate the numerical performance of the proposed method in terms of REMSE and jump preservation, we select the following three popular state-of-the-art methods: (i) image denoising using usual random forest technique Breiman (2001), (ii) edge preserving image denoising method proposed by Qiu (2009), and (iii) image denoising method by non-local means proposed by Buades et al. (2011).

(i) Image denoising using random forest (RF): Random forest is the most popular tree-based ensemble learning algorithm in the context of nonparametric regression. Since it is an ensemble method, it is capable of accommodating the complex shapes in the data. One of the major advantages of the random forest over a single tree-based method (CART) is that it can avoid the problem of overfitting, unlike CART. Therefore, random forest could be used as a potential image denoising method. However, in the case of image smoothing, to accommodate the complex structure of the image object, random forest splits more, leading to deeper trees, which aggravates the problem of overfitting. The proposed method also involves

a decision tree, and therefore it is comparable with random forest-based methods.

(ii) Jump preserving local linear kernel smoothing (JPLLK): The method first divides the local neighborhood into two parts using gradient information. Then, three different estimates are calculated using local linear kernel regression using the left, right, and whole neighborhood, respectively. The best of these three locally fitted surfaces is chosen, depending on their mean squared error. Then, we follow the same procedure on the fitted surface again. But this time, one of those three estimates is chosen as the final estimate based on their estimated variance. This method is very good at preserving jumps, although its image denoising performance falls short when compared to our proposed method.

(iii) Image denoising based on non-local means (NLM): In this method, the true image intensity value at a given pixel is estimated using a weighted average on a large portion of the entire image. To calculate the weights, each pixel is scored according to its similarity with the given pixel. Then, these scores are used as weights for the estimation. For convenience, we choose an implementation proposed by Froment (2014), as it automatically calculates the parameters for the denoising method. Among these competing methods, the non-local means method usually performs at

best at image denoising in the sense of visual appearances, at the cost of blurring the image.

5.2 Simulations

In the simulation study, we consider a triangle image as the test image. See the extreme left of the upper panel in Figure 2 for the true triangle image. The image intensity function of the test image can be expressed as

$$f(u, v) = \mathbb{I}[(u, v) \in \Omega_1], \quad (u, v) \in [0, 1] \times [0, 1], \tag{5.8}$$

where $\mathbb{I}$ is the indicator function, and Ω_1 denotes the region enclosed by the triangle. Throughout this section, we use Gaussian additive noise to generate the noisy image surface. See the third image from the left of the upper panel in Figure 2 for the noisy test image. We carry out the simulation study under the following settings: (i) to demonstrate the noise removal property, we use three different noise levels with $\sigma = 0.1, 0.2,$ and 0.3. (ii) to demonstrate the consistency of the proposed estimation method, we perform further simulations with two different image resolutions: 200×200, i.e., $n = 200$, and 400×400, i.e., $n = 400$. Table 1 shows the REMSE values for various cases and methods. Moreover, the summary of the performances

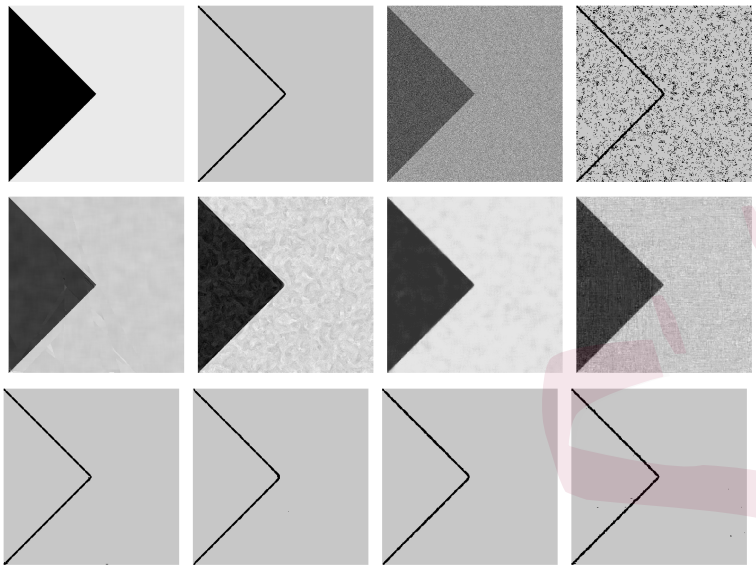

Figure 2: Performance comparison on the simulated test image with pointwise Gaussian noise with $\sigma = 0.3$. The first row shows the original image, original edges, noisy image, and detected edges on the noisy image, respectively. The second row shows denoised images produced by the proposed method, JPLLK, NLM, and RF. The third row shows the detected edges of the corresponding denoised images.

of the concerned methods with respect to edge preservation is provided in Table 2. Based on 100 replications, the results are shown in Table 1.

From Tables 1 and 2, we can see that the proposed ORT-based method performs better when $n = 400$ rather than 200, in terms of both REMSE and edge preservation. As the proposed method is asymptotically consistent, these results are intuitively reasonable. The proposed method is uniformly better than RF and JPLLK both in noise removal and edge preservation. Except for the case when $n = 200$ and $\sigma = 0.1$, the pro-

Table 1 Comparisons of various methods on the simulated test image using (REMSE$\times 10^3$) values based on 100 independent replications with standard error within the parentheses.

		Proposed Method	Competing Methods		
Resolution	Noise	ORT	JPLLK	NLM	RF
200×200	0.3	33.8 (4.697)	54.1 (1.196)	51.1 (1.559)	108.3 (1.095)
	0.2	25.5 (4.237)	40.0 (0.659)	29.6 (1.136)	86.5 (0.728)
	0.1	18.1 (4.752)	27.4 (0.382)	14.4 (0.511)	70.6 (0.717)
400×400	0.3	22.3 (4.024)	49.5 (0.539)	40.0 (0.821)	91.6 (0.852)
	0.2	15.4 (3.822)	33.9 (0.318)	24.9 (0.584)	69.6 (0.522)
	0.1	7.9 (3.349)	19.8 (0.203)	12.5 (0.255)	53.0 (0.470)

Table 2 Comparisons of various methods regarding jump preservation on the simulated test image using ($d_{KQ} \times 10^3$).

		Proposed Method	Competing Methods		
Resolution	Noise	ORT	JPLLK	NLM	RF
200×200	0.3	0.427	0.649	0.722	2059
	0.2	0.115	0.265	0.151	0.470
	0.1	0.110	0.266	0.000	0.391
400×400	0.3	0.086	0.570	0.386	1.67
	0.2	0.086	0.087	0.059	0.275
	0.1	0.005	0.068	0.003	0.177

posed method outperforms NLM in all other cases. The denoised images are shown in the middle panel of Figure 2. From the visual impression, it appears that the denoising performances of JPLLK and RF are relatively poor compared to the proposed method. NLM performs relatively better; however, several noisy patchy regions are present in the denoised image. We demonstrate the edge detection performances in the lower panel of Figure 2. From this simulation study, we see that the proposed method outperforms the competing methods in almost all scenarios considered above.

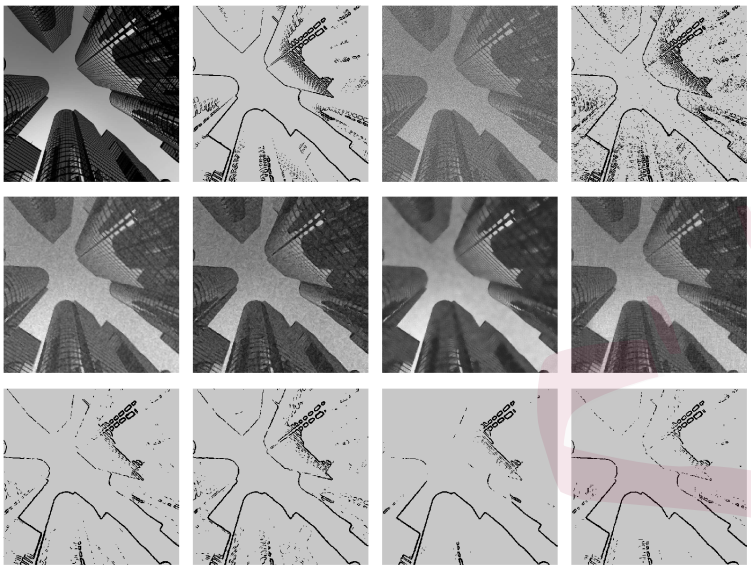

Figure 3: Performance comparison on the building image with point-wise Gaussian noise with $\sigma = 0.2$. The first row shows the original image, original edges, noisy image, and detected edges on the noisy image, respectively. The second row shows denoised images produced by the proposed method, JPLLK, NLM, and RF. The third row shows the detected edges of the corresponding denoised images.

5.3 Applications on real images

We apply the proposed ORT-based method on complicated real images and perform a comparative analysis with the state-of-the-art competing methods. In this regard, we consider two different real images. The image in the extreme left of the upper panel of Figure 3 shows high-rise buildings with resolution 523×523. The image in the extreme left of the upper panel of Figure 4 shows a moving aero-plane with resolution 628×628. The numerical performances of the concerned methods regarding noise removal

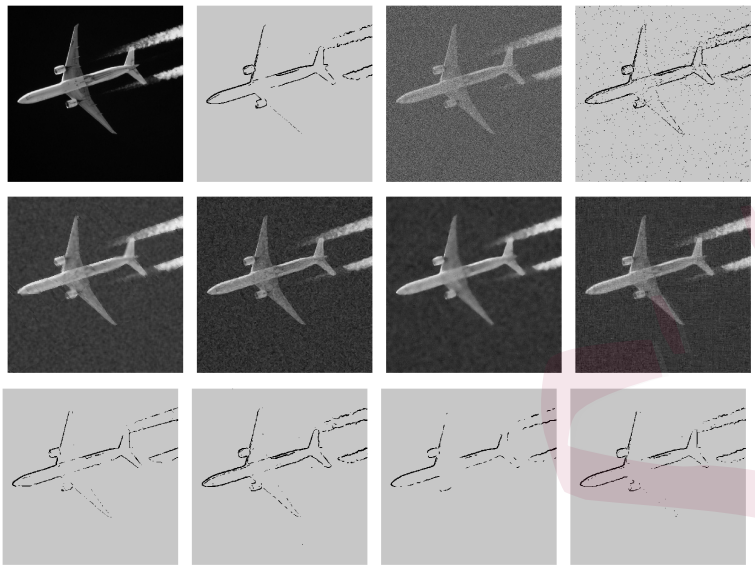

Figure 4: Performance comparison on the aero-plane image with point-wise Gaussian noise with $\sigma = 0.2$. The first row shows the original image, original edges, noisy image, and detected edges on the noisy image, respectively. The second row shows denoised images produced by the proposed method, JPLLK, NLM, and RF. The third row shows the detected edges of the corresponding denoised images.

and edge preservation are provided in Tables 3 and 4. From Table 3, we see that for the building image, the proposed method uniformly outperforms JPLLK and RF in terms of REMSE. In this regard, the NLM method is comparatively better than the proposed method except for larger noise levels, when its performance is similar to the proposed method. In terms of edge preservation, the proposed method is much better than NLM in most cases. Although edge-preservation measures are good for RF and JPLLK, their performance on noise removal is rather poor, especially when the noise

level is high. In summary, we can conclude that the performance of the proposed method shows a reasonable trade-off between noise removal and edge preservation abilities. The proposed method has good image denoising capability along with a good edge preservation property.

Table 3 Comparisons of various methods on the real images using (REMSE$\times 10^3$) values based on 100 independent replications with standard error within the parentheses.

		Proposed Method	Competing Methods		
Image	Noise	ORT	JPLLK	NLM	RF
Building	0.3	95.8 (0.873)	102.2 (0.324)	96.2 (0.343)	106.1 (0.497)
	0.2	82.1 (0.402)	100.2 (0.221)	77.0 (0.248)	84.8 (0.415)
	0.1	58.8 (3.22)	80.3 (0.153)	48.6 (0.142)	64.1 (0.269)
Plane	0.3	45.4 (0.741)	89.0 (0.505)	44.6 (0.301)	82.1 (0.829)
	0.2	37.4 (0.621)	60.9 (0.183)	33.2 (0.191)	56.1 (0.507)
	0.1	27.7 (0.704)	33.9 (0.099)	16.7 (0.082)	34.8 (0.299)

Table 4 Comparisons of various methods regarding jump preservation on real images using ($d_{KQ} \times 10^3$).

		Proposed Method	Competing Methods		
Image	Noise	ORT	JPLLK	NLM	RF
Building	0.3	13.2	4.14	65.16	8.53
	0.2	6.86	4.14	25.26	6.67
	0.1	1.59	1.37	2.38	5.63
Plane	0.3	8.34	15.25	31.45	3.46
	0.2	4.56	2.33	6.51	1.67
	0.1	1.63	1.10	0.75	2.29

5.4 Comparison with a state-of-the-art deep learning model

We now compare the proposed algorithm with one of the best-performing state-of-the-art deep learning based models. Note that this is not an apple-to-apple comparison, as the deep learning based models are trained on

multiple images of various types, while the proposed method is for a single image only. The model named *Restormer*, proposed by Zamir et al. (2022), is a transformer-based model tailored for image restoration tasks such as denoising. Unlike conventional convolutional approaches, it employs multi-head attention and feeds the forward network to effectively model both complex local textures and long-range spatial dependencies. This design allows *Restormer* to suppress noise while preserving structural details and natural image fidelity. As a result, it achieves state-of-the-art performance in real image denoising and other restoration problems. Table 5 shows the performance of the proposed algorithm when compared to the pre-trained model *Restormer* under the same setup as before. This is reasonable because the images on which *Restormer* was originally trained are possibly different from the test images we apply to. From Table 5, we can see that

Table 5 Comparisons with *Restormer* on the real images using (REMSE$\times 10^3$) values based on 100 independent replications with standard error within the parenthesis.

		Proposed Method	Competing Method
Image	Noise	ORT	Restormer
Building	0.3	95.8 (0.873)	165.7(0.5)
	0.2	82.1 (0.402)	100.2(0.3)
	0.1	58.8 (3.22)	52.7(0.1)
Plane	0.3	45.4 (0.741)	154.1(0.4)
	0.2	37.4 (0.621)	75.2(0.4)
	0.1	27.7 (0.704)	35.4(0.1)

the proposed algorithm works much better in high noise scenarios when

compared to *Restormer*. Note that the performance of *Restormer* may be affected by the high noise used in the demonstrated cases, and it may not have been trained using noise of this magnitude. For implementation of *Restormer*, we utilize the official codes and pre-trained weights provided by Zamir et al. (2022).

5.5 Performance comparison on a large-scale dataset

We further evaluate the proposed method using the SIDD+NTIRE challenge dataset (Abdelhamed et al., 2020). For this analysis, the validation dataset is employed, as it provides publicly available ground-truth images. The dataset comprises 32 base images, each with 32 cropped variants, resulting in a total of 1024 images with a spatial resolution of 256×256. In the NTIRE challenge, 22 participating teams submitted 24 distinct algorithms, all trained on the validation data and subsequently assessed on the test set. Our proposed algorithm attains a peak signal-to-noise ratio, i.e., PSNR of 33.291 on the sRGB benchmark. This result is noteworthy, as it situates the proposed algorithm within the competitive range of the leading methods, including some of the best-performing deep learning approaches reported in the original challenge. Considering that the proposed method does not rely on training with large datasets, it also demonstrates broader

applicability and potential for generalization.

5.6 Applications on 3-D image denoising

In this section, we demonstrate the application of the proposed algorithm to 3-D image denoising. In a simulation study, we consider a tetrahedron with its base aligned parallel to the XY-plane. We then add Gaussian noise with various standard deviations to the image for testing the denoising performance of the proposed method. The simulated image has a resolution of $128 \times 128 \times 128$. In the real-data setting, we use a lung CT scan from the dataset provided by Li et al. (2020). The image is rescaled to $128 \times 128 \times 128$, intensities are normalized to $[0, 1]$, and then point-wise Gaussian noise is applied. We evaluate the performance of ORT under noise levels with standard deviations of 0.1, 0.2, and 0.3 across 10 independent trials. The results of ORT on both simulated and real images are presented in Figure 5 and Table 6.

Table 6 Performance evaluation of the proposed method for 3-D image denoising using RMSE.

Noise Level	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$
Simulated Image	0.0140 (0.0053)	0.0267(0.0046)	0.03831 (0.0047)
Real Image	0.0394 (0.0003)	0.0536 (0.0002)	0.0702 (0.0001)

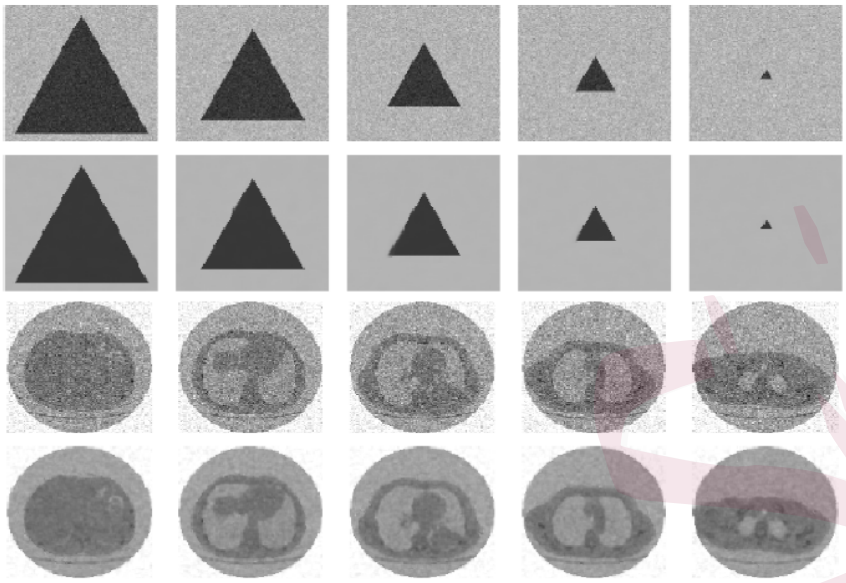

Figure 5: The top two rows show various slices of the noisy and denoised tetrahedron image, while the bottom two rows show the corresponding results for a real lung CT image. The columns correspond to the slices along the Z-axis at indices 13, 38, 64, 90, and 115.

6. Concluding Remarks

In this paper, we propose a framework using Oblique-axis Regression Tree to estimate the underlying discontinuous function on a lattice and show its point-wise convergence with an explicit convergence rate under mild assumptions. We also show that similar points are often grouped in the same leaf node of the decision tree we obtain from the proposed algorithm, which is very desirable since it facilitates further analysis. Then, we proceed to apply this algorithm to noisy images for removing noise, and show

its superior performance compared to many other state-of-the-art methods. However, there are scopes for improvement. In this paper, we only show that the algorithm converges when the number of lattice points increases to infinity. Instead, one can consider a different approach and keep the number of lattice points fixed while letting the number of samples go to infinity. In this way, one can extend the use of the proposed algorithm to various real-life problems, such as image monitoring. The partitioning is restricted to linear types only to ensure that the partitions remain convex, which is necessary for subsequent proofs. However, if we could extend the covariates or pixel coordinates by including nonlinear functions of them, and correspondingly extend the image in a meaningful way, then the results developed here could potentially be applied in that setting to achieve improved performance. Instead of working with equally spaced design points, one can expand its capability to work with general tabular data, where the design points are not necessarily evenly distributed. Another future direction of research is to generate multiple sample images using bootstrapping techniques and build a random forest using the proposed algorithm to facilitate better estimates. Apart from these, the proposed algorithm can also be applied on other problems, e.g., image segmentation, denoising of an image sequence, and so on.

Acknowledgments: The authors thank the editor, the associate editor, and two anonymous reviewers for their comments and suggestions, which greatly improved the quality of this paper.

References

Abdelhamed, A., M. Afifi, R. Timofte, M. Brown, Y. Cao, Z. Zhang, W. Zuo, X. Zhang, J. Liu, W. Chen, C. Wen, M. Liu, S. Lv, Y. Zhang, Z. Pan, B. Li, T. Xi, Y. Fan, X. Yu, and V. Kumar (2020, 06). Ntire 2020 challenge on real image denoising: Dataset, methods and results. pp. 2077–2088.

Breiman, L. (1984). *Classification and Regression Trees*. (The Wadsworth statistics / probability series). Wadsworth International Group.

Breiman, L. (2001). Random forests. *Machine learning 45*, 5–32.

Buades, A., B. Coll, and J.-M. Morel (2011). Non-local means denoising. *Image Processing On Line 1*, 208–212.

Cattaneo, M. D., R. Chandak, and J. M. Klusowski (2024). Convergence rates of oblique regression trees for flexible function libraries. *The Annals of Statistics 52*(2), 466–490.

Chaudhuri, A. and S. Chatterjee (2023). A cross-validation framework for signal denoising with applications to trend filtering, dyadic cart and beyond. *The Annals of Statistics 51*(4), 1534–1560.

Chu, C., I. Glad, F. Godtliebsen, and J. Marron (1998). Edge-preserving smoothers for image

processing. *Journal of the American Statistical Association 93*(442), 526–541.

Donoho, D. L. (1997). Cart and best-ortho-basis: a connection. *The Annals of statistics 25*(5), 1870–1911.

Froment, J. (2014). Parameter-free fast pixelwise non-local means denoising. *Image Processing On Line 4*, 300–326.

Gonzalez, R. and R. Woods (2018). *Digital Image Processing* (4th ed.). USA: Pearson.

Gramacy, R. B. and H. K. H. Lee (2008). Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association 103*(483), 1119–1130.

Kang, Y. and P. Qiu (2024). *DRIP: Discontinuous Regression and Image Processing*. R package version 2.3.

Kang, Y., Y. Shi, Y. Jiao, W. Li, and D. Xiang (2021). Fitting jump additive models. *Computational Statistics & Data Analysis 162*, 107266.

Konomi, B. A., H. Sang, and B. K. Mallick (2014). Adaptive bayesian nonstationary modeling for large spatial datasets using covariance approximations. *Journal of Computational and Graphical Statistics 23*(3), 802–829.

Li, P., S. Wang, T. Li, J. Lu, Y. HuangFu, and D. Wang (2020). A large-scale ct and pet/ct dataset for lung cancer diagnosis (lung-pet-ct-dx). `https://www.cancerimagingarchive.net/collection/lung-pet-ct-dx/`.

REFERENCES

Mukherjee, P. S. and P. Qiu (2011). 3-d image denoising by local smoothing and nonparametric regression. *Technometrics 53*(2), 196–208.

Qiu, P. (2005). *Image Processing and Jump Regression Analysis*. Wiley Series in Probability and Statistics. New York: Wiley.

Qiu, P. (2009). Jump-preserving surface reconstruction from noisy data. *Annals of the Institute of Statistical Mathematics 61*, 715–751.

Roy, A. and P. S. Mukherjee (2024). Image comparison based on local pixel clustering. *Technometrics 66*(4), 495–506.

Roy, A. and P. S. Mukherjee (2025). Upper quantile-based cusum-type control chart for detecting small changes in image data. *Journal of Applied Statistics 52*(11), 2156–2171.

Zamir, S. W., A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang (2022). Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5728–5739.

Zhan, H., Y. Liu, and Y. Xia (2024). Consistency of oblique decision tree and its boosting and random forest.

Author Affiliations and Contact Details:

Subhasish Basak, Indian Statistical Institute, Kolkata, India,
E-mail: subhasish.sunny@gmail.com

Anik Roy, Emory University, Atlanta, USA,
E-mail: anikroy7713@gmail.com

Partha Sarathi Mukherjee, Indian Statistical Institute, Kolkata, India,
E-mail: psmukherjee.statistics@gmail.com