

Approximations by Smooth Transitions in Binary Space Partitions

Marcos Lage¹, Alex Bordignon¹, Fabiano Petronetto¹,
Álvaro Veiga², Geovan Tavares¹, Thomas Lewiner¹, Hélio Lopes¹

¹Laboratório Matmídia – Departamento de Matemática

²Departamento de Engenharia Elétrica

Pontifícia Universidade Católica do Rio de Janeiro — PUC-Rio

¹{mlage,alexlaier,gts,tomlew,lopes}@mat.puc-rio.br, ²alvf@ele.puc-rio.br

Abstract

This work proposes a simple approximation scheme for discrete data that leads to an infinitely smooth result without global optimization. It combines the flexibility of Binary Space Partitions Trees with the statistical robustness of Smooth Transition Regression Trees. The construction of the tree is straightforward and easily controllable, using error-driven metrics or external constraints. Moreover, it leads to a concise representation. Applications on synthetic and real data, both scalar and vector-valued demonstrated the effectiveness of this approach.

Keywords: *Binary Space Partitions, Smooth Transition Regression Trees, Classification and Regression Tree.*

1. Introduction

Approximation of discrete data constitutes a fundamental step in modeling, analysis and visualization. It received a lot of attentions in the last decade, and many elaborate schemes have been devised, using techniques from optimization theory to statistical modeling. However, very few adaptive and efficient methods for such tasks combine simplicity and smooth results. This work proposes a simple approximation scheme based on *Binary Space Partitioning* (BSP) and *Smooth Transition Regression Trees*.

The BSP space subdivision is naturally represented by a binary tree, named *BSP-Tree*, and was introduced by Fuchs, Kedem and Maylor in [8]. It is today recognized as a very useful tool, since it is extremely simple and flexible. Binary partitioning tree is also a very suitable statistical tool, since it could work as a scheme to represent decision rules. In this context, it has been widely used to classification and regression applications through a technique called *Classification and Regression Tree* (CART) Analysis [3]. In the last decades, the interest in CART analysis has increased a

lot, mainly for classification applications [11, 13]. Indeed, regression trees are very simple to interpret and to model: it subdivides the input space into a set of rectangles and fits a constant in each one.

However, in CART models, finding the best separation hyperplane to fit the data in terms of least squares is computationally infeasible. In order to obtain the model quickly, a greedy approach is generally adopted, fixing the hyperplane always perpendicular to some canonical vector and choosing the cutting point by some heuristic strategy. Thus, the resulting binary partition tree corresponds to a variation of the so famous KD-Tree [16].

Another CART disadvantage is that the regression function is not differentiable on the border of the rectangular regions. To solve this problem, Rosa et al. proposed in [5] other tree regression model that combines the CART model with the Smooth Transition Regression (STR), which is used in non-linear time series analysis [9]. This new tree model, called the *Smooth Transition Regression Tree* (STR-Tree), substitutes the indicator function on the CART regression model by a smooth fuzzy membership function. It has also been applied to time series analysis with multiple-regimes [5]. However, similarly to the CART model, it restricts the hyperplanes to be axis-aligned, reducing the flexibility of the approximation.

Contributions In this work we propose a modification of the STR-Tree to obtain a simple and fast approximation scheme. It can be considered as a hierarchical plane-basis function modeling (similarly to radial-basis functions [4]), inheriting both the flexibility of BSP and the smoothness of STR-Trees. Moreover, it does not require global optimization while being infinitely smooth. The tree construction is easily controllable, using error-driven metrics or external constraints, and generally leads to concise representations. We apply this new scheme to vector field and scalar field reconstruction from sparse data.

Paper outline. Section 2 discusses the BSP-Tree and the CART and STR-Trees regression schemes. Section 3 introduces a new method based on STR-Trees to smooth function approximations. Section 4 shows the results. And finally, section 5 concludes the work and proposes future directions.

2. Previous and related works

In this section we discuss three kinds of trees related to this work: the BSP-Tree, the CART and the STR-Tree.

2.1. Binary Space Partition Trees

The BSP technique is a simple and efficient method to adaptively subdivide an initial given n -dimensional domain $\mathcal{S} \subset \mathbb{R}^n$ into convex sets in order to match the geometry of a given set of input points. Fuchs, Kedem and Maylor in [8] proposed the BSP-Tree representation for this kind of space subdivision. The BSP-Tree construction is a process that takes the initial space \mathcal{S} and divides it into two new subspaces by a hyperplane that intersects its interior. These two new subspaces can both be partitioned by other hyperplanes and recursively this process continues until some stop criterion is achieved. For a hyperplane in n -dimensional space we mean a $(n - 1)$ -dimensional subspace that divides the original space into two half-spaces. For example, in three-dimensional spaces, the “hyperplane” is a plane and in two-dimensional spaces it is a line. BSP-Tree should be constructed in such a way that the convex regions on the leaves could capture better the information data for the points that are inside them. And the strategy for choosing the subdividing hyperplane and the stop criteria are application dependent. Figure 1 shows a BSP-Tree for a space partitioning.

BSP-Tree has been applied to hidden surface removal [8], image processing [15, 6], solid modeling [19, 7], point-based processing [14] and geometry compression [18, 2], just to cite a few. A *KD-tree* is a specific kind of a BSP-

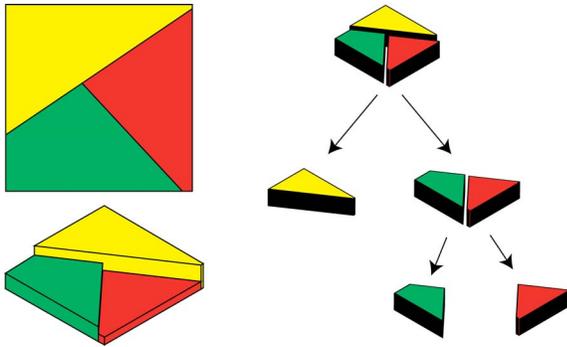


Figure 1. Two level BSP-Tree example.

Tree where the splitting hyperplanes are axis-aligned.

2.2. Classification and Regression Trees

In statistical learning, tree based methods generate simple and powerful models for classification and regression [10]. The most popular tree based method is the CART (Classification and Regression Tree) [3]. In the CART model, the regression function consists in a sum:

$$f(\mathbf{x}) = \sum_{i=1}^L c_i \mathbb{I}(\mathbf{x} \in \mathcal{R}_i), \quad (1)$$

where c_i are constants, $\mathcal{R}_i, i = 1 \dots L$ are the rectangular regions obtained from the binary partitioning tree construction with axis-aligned hyperplanes, and \mathbb{I} is an indicator function (that is equal to 1, when the argument is true, and is equal to 0 otherwise). In other words, the CART model associates to each rectangular region a constant value. In this way, the response value of the estimated function when the point \mathbf{x} is in the region \mathcal{R}_i is c_i . Once the rectangular regions \mathcal{R}_i 's are given, the best constants c_i 's are just the average of all y_i in the corresponding region, when the criterion is the minimization of the sum of the squared errors $\sum (y_i - f(\mathbf{x}_i))^2$. Figure 2 shows an example of an estimated regression function, the height for each region corresponds to the value of c_i . For more details about the CART models see [3].

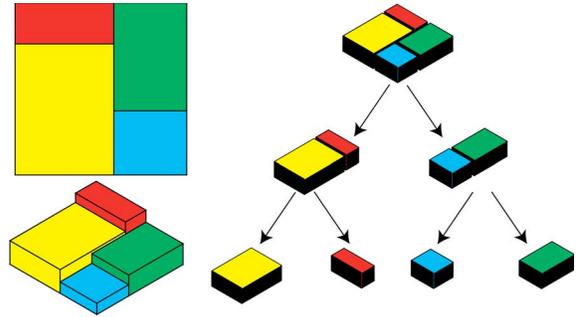


Figure 2. Two level CART regression function example.

2.3. Smooth Transition Regression Trees

The STR-Tree model, proposed by Rosa et al. in [5], takes advantage of much of the CART structure presented above, but substitutes the sharp splits of the CART model by smooth splits. In the STR-Tree regression model the indicator function \mathbb{I} in equation 1 is substituted by a logistic function, parameterized by $\lambda > 0, b \in \mathbb{R}$ and $j = \{1, \dots, n\}$, defined as:

$$G(\mathbf{x}; \lambda, b, j) = \frac{1}{1 + e^{-\lambda(e_j^t \mathbf{x} - b)}}, \quad (2)$$

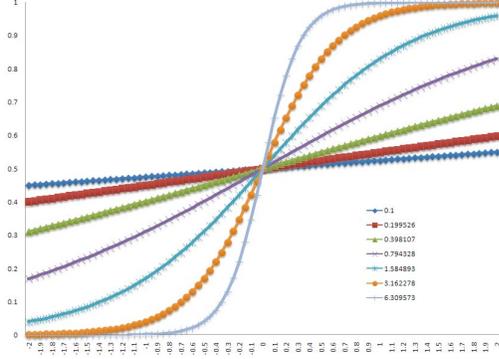


Figure 3. G smooth transition function for different values of λ .

where the \mathbf{e}_j is the j^{th} canonical vector. Consider the simplest tree with two leaf nodes, then the regression function in the STR-Tree model is written as:

$$f(\mathbf{x}) = c_0 G(\mathbf{x}; \lambda, b, j) + c_1 (1 - G(\mathbf{x}; \lambda, b, j)).$$

The parameter λ controls the smoothness of the transition from one node to the other. Figure 3 shows several examples of the G function for several values of λ .

Consider a full STR-Tree model with depth d , then the number of leaves is $L = 2^d$ and the number of parent nodes is $\sum_{i=0}^{d-1} 2^i$. Then, given a rectangular partition of the initial domain, the STR-Tree regression function is defined as:

$$f(\mathbf{x}) = \sum_{i=1}^L c_i B_i(\mathbf{x}; \psi), \quad (3)$$

where ψ represents all the parameters used to define the function G for each parent node p , they are: the value of

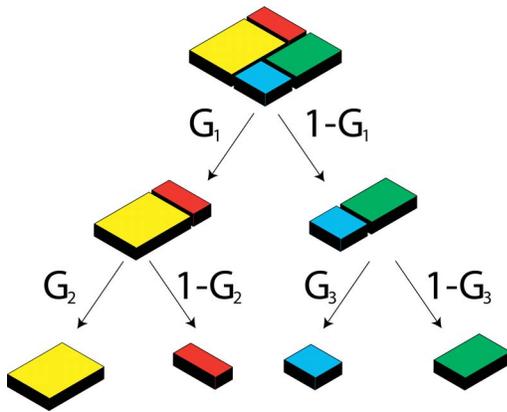


Figure 4. Blending functions construction based on the STR-Tree.

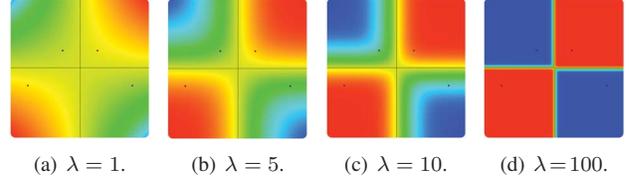


Figure 5. Approximations 4 samples from a level 2 KD-Tree, for different values of λ .

λ_p , the split canonical direction \mathbf{e}_{j_p} and the value of b_p . For the case when $d = 2$, one can write:

$$\begin{aligned} B_1(\mathbf{x}; \psi) &= G(\mathbf{x}, \lambda_1, \mathbf{e}_{j_1}, b_1)G(\mathbf{x}, \lambda_2, \mathbf{e}_{j_2}, b_2) \\ B_2(\mathbf{x}; \psi) &= G(\mathbf{x}, \lambda_1, \mathbf{e}_{j_1}, b_1)(1 - G(\mathbf{x}, \lambda_2, \mathbf{e}_{j_2}, b_2)) \\ B_3(\mathbf{x}; \psi) &= (1 - G(\mathbf{x}, \lambda_1, \mathbf{e}_{j_1}, b_1))G(\mathbf{x}, \lambda_3, \mathbf{e}_{j_3}, b_3) \\ B_4(\mathbf{x}; \psi) &= (1 - G(\mathbf{x}, \lambda_1, \mathbf{e}_{j_1}, b_1))(1 - G(\mathbf{x}, \lambda_2, \mathbf{e}_{j_2}, b_2)) \end{aligned}$$

where $\psi = (\lambda_1, \lambda_2, \lambda_3, \mathbf{e}_{j_1}, \mathbf{e}_{j_2}, \mathbf{e}_{j_3}, b_1, b_2, b_3)$. Figure 4 illustrates how these blending functions B_i are constructed, and Figure 5 illustrated this blending function when varying parameter λ of G . Notice that such construction naturally defines a partition of unity [1].

In [5] the authors propose to use a least squares approach in order to determine the better c_i that minimize the sum of the squared residuals for all input points, once the rectangular partition of the space is given.

3. Smooth transition BSP approximations

Until now, the STR-Tree has been applied basically to non-linear regression analysis. We propose here a modification of those to adapt them to graphics applications. Such modifications intend to capture better the geometry of the input points, using application-dependent heuristics for determining the split plane. Note that, with the BSP framework, such plane can have arbitrary direction. This flexibility on the tree allows a very simple regression function construction. In addition, in order to accelerate the regression function evaluation, we propose the use another fuzzy membership functions to accelerate the evaluation.

3.1. Problem description

Given a set of N points $\mathcal{P} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, where all these points are on the box $[0, 1]^n$. Assume that each point \mathbf{x}_i is associated with an attribute vectors $\mathbf{y}_i \in \mathbb{R}^m$, $m \geq 1$. Let \mathcal{A} denote the set of all attribute vectors $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$. We would like to obtain a regression function that describe the relationship between \mathbf{x} and \mathbf{y} .

3.2. The regression function

Suppose that a BSP-Tree that represents the binary space partition of the initial domain $[0, 1]^n$ into L convex regions

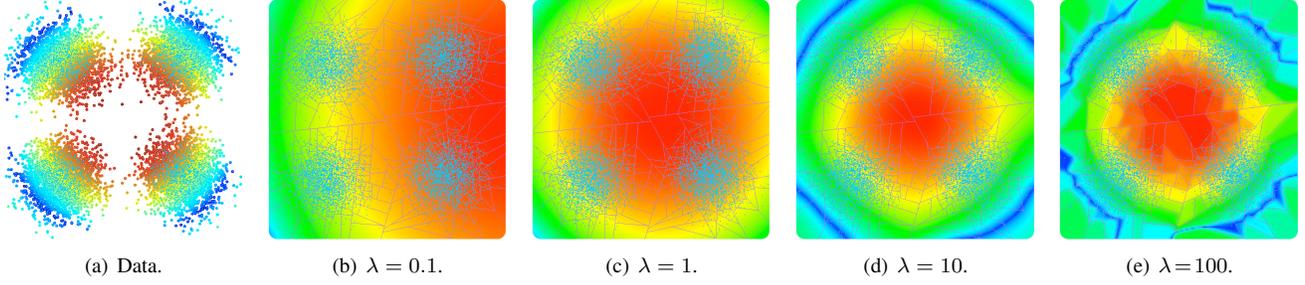


Figure 6. Approximation of a mexican-hat radial function with a level 10 BSP-Tree.

$\mathcal{R}_1, \dots, \mathcal{R}_L$ is given. We can define the regression function in the same manner of the STR-Tree model by substituting, on each internal node p of the tree, the directions \mathbf{e}_{j_p} by unitary vectors \mathbf{w}_p . In this way the fuzzy membership functions is controlled by the distance to the corresponding plane ($\mathbf{w}_p^t \mathbf{x} - b$).

G and the regression function then follow:

$$G(\mathbf{x}; \lambda_p, b_p, \mathbf{w}_p) = \frac{1}{1 + e^{-\lambda_p(\mathbf{w}_p^t \mathbf{x} - b)}},$$

$$f(\mathbf{x}) = \sum_{i=1}^L \mathbf{c}_i B_i(\mathbf{x}; \psi).$$

Note that in our formulation, we also substitute the scalar constant c_i by a constant vector \mathbf{c}_i since we would like to reconstruct not only scalar fields (as it is done in the CART and STR-Tree models) but also vector fields. This function has a nice dependence on parameter λ (see Figure 6).

In our scheme, similarly to CART, the value of \mathbf{c}_i is the average of the \mathbf{y}_j values of the corresponding points \mathfrak{S}_j inside the convex region \mathcal{R}_i .

3.3. Function evaluation

To evaluate the constructed function, one has to traverse recursively the BSP-Tree in order to build the blending functions for each leaf node. Note that all the internal nodes have to be traversed because of the shape of G , which is close to zero (but not equal to) when the distance to the plane is a large negative number and is close to one (but not equal to) when the distance to the plane is a large positive number.

In order to accelerate the evaluation process we propose the use of a polynomial of degree 5 that approximates G . Such function will be named G^* , it depends on the distance to the corresponding split plane and on the value of a parameter λ :

$$G^*(\mathbf{x}; \lambda, \mathbf{w}, b) = \begin{cases} 0 & \text{if } d \leq -\frac{1}{\lambda} \\ 1 & \text{if } d \geq \frac{1}{\lambda} \\ \frac{1}{2} + \frac{15\lambda}{16}d - \frac{5\lambda^3}{8}d^3 + \frac{3\lambda^5}{16}d^5 & \text{otherwise} \end{cases}$$

where $d = \mathbf{w}^t \mathbf{x} - b$. In this way, if the point \mathbf{x} is far from the plane by more than $1/\lambda$, the value of G^* at this point will be 1, as a consequence, the value of $(1 - G^*)$ at the same point will be 0 (and vice-versa). So, the tree traversal could stop at this node, accelerating the evaluation.

3.4. Derivatives

In several scientific simulation or graphics applications it is necessary to compute the derivatives of the approximate function. For example, for vector field reconstruction one would like to compute the curl or the divergent of the field at the given point, and for scalar field reconstruction the gradient. To evaluate the derivative of the regression function in this scheme, we have to evaluate the derivative of an average of the blending functions, which is calculus exercise. The algorithm to do this evaluation efficiently is also based on the BSP-Tree traversal. To do so, it recurses to compute the vector values of both the field and its derivatives, since they are needed for the multiplication derivation rule.

Observe that the proposed regression function is C^∞ everywhere when the blending function is constructed by the use of the G fuzzy membership function, and is C^2 everywhere when the fuzzy membership function is G^* .

3.5. Building the BSP-Tree

The first step in our algorithm is to find a partition of the initial domain into convex regions by the use of a BSP-Tree. The subdivision process relies on two criteria: how to find the best separating hyperplane and when to stop the process of subdivision.

Three types of heuristic criteria are generally used to control the separation hyperplane at each level: fitting to the input points' distribution in the cell; adapting to the current approximation error; or using external variables such as global density targets.

From our experiments, we suggest to determine the separating hyperplane based on the input points: perpendicular to the first *principal component* that passes to the center of

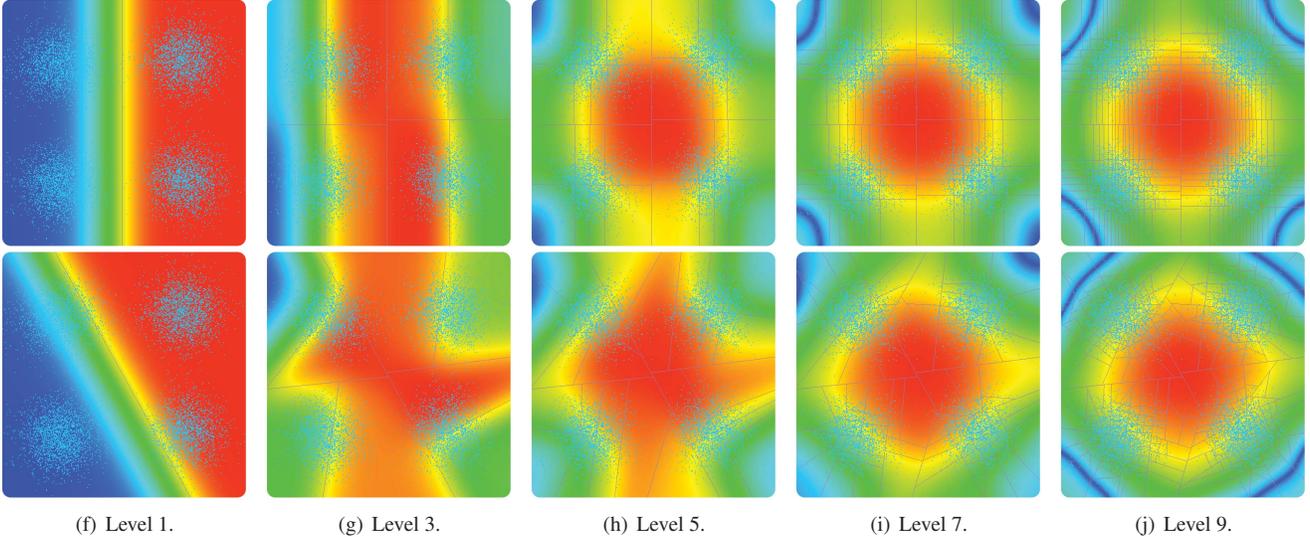


Figure 7. Approximation of the radial function of Figure 6 with KD-Trees (top) and BSP-Trees from the points distribution (bottom), $\lambda = 10$. The BSP trees better respects the radial nature of the function, in particular in diagonal parts.

mass of the input points contained on the given region. Another criteria when the dimension of the input points \mathbf{x} is equal to the target vectors \mathbf{y} is to use the hyperplane parallel to the average of the target vectors that passes to the center of mass of the input points contained on the region.

Different criteria can be used to establish when to stop the subdivision, for example: the minimum number of points in each leaf region, the maximum level of the tree, a maximum error tolerance on the input points, or a combination of these three. In this work, we use the minimum number of points per leaf region.

4. Results

We tested the proposed approximation in different contexts. First, we experimented a synthetic reconstruction from uneven sampling with different values of λ (see Figure 6). We observe that the value $\lambda = 10$ is a reasonable choice. We confirmed on other tests that it is a good default value for such scalar approximations, when the domain is mapped onto $[-1, 1]^2$. Then, we compare our approach on scalar fields with KD-tree based techniques such as STR-Trees on the same data set (see Figure 7 and Table 1). We can observe the superiority of BSP that better captures the diagonal features of the function. We further compare our approach for vector field with quadtree-based techniques such as multiple partition of unity implicit for vector fields [12] (see Figure 11). On this simulated vector field, the proposed technique outperforms quadtree-local polynomial approaches. Moreover, with the compact sup-

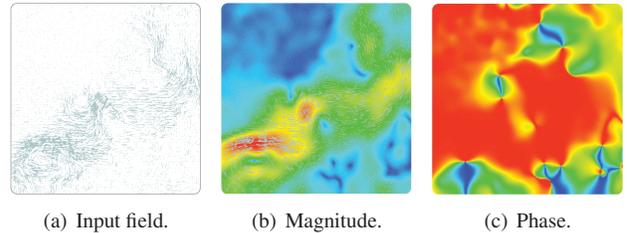


Figure 8. Approximation of a real vector field from particle-image velocimetry with a level 11 BSP: the color map represents the magnitude of the field ($\lambda = 11$).

port proposed in Section 3.3, the evaluation is faster than polynomial approaches: with G at level 11, the construction took 7.310^{-4} seconds, against 1.210^{-5} for G^* . We finally tested our technique on real vector field measured from particle image velocimetry (see Figure 8), showing the stability of the method even in the presence of noise.

To further test the possibilities of our smooth transition approximations, we compute the derivatives of reconstructed vector fields. In particular, we experiment on a synthetic vector field with six vortices and show the approximate divergence and curl computed from a random sampling (see Figure 9). We check that those derivatives stress the features of the field. Furthermore, we checked the divergence of the velocities obtained from a numerical fluid simulation [17], which should be close to zero (see Figure 10).

Finally, we apply our technique to curve reconstruction, similarly to RBF techniques [4]. From a curve sampled

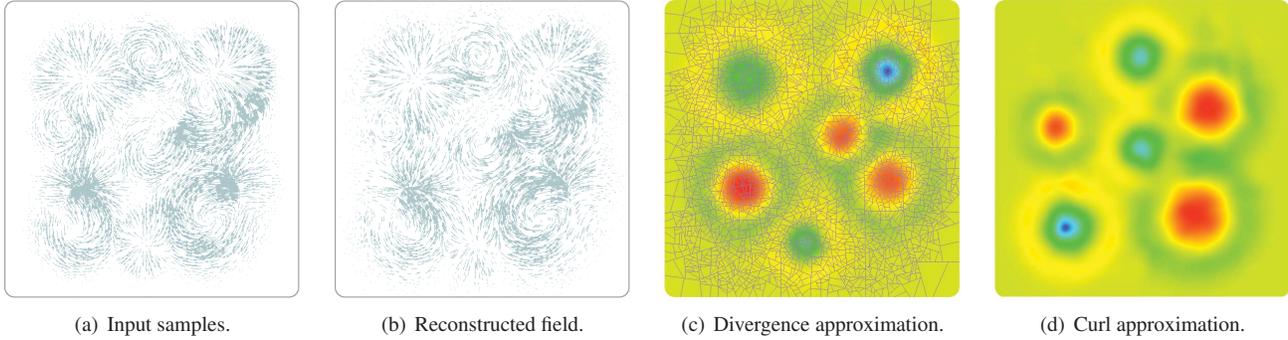


Figure 9. Divergence and curl of the approximation on a synthetic vector field with six vortices, using level 11 BSP: the vortices are clearly identified.

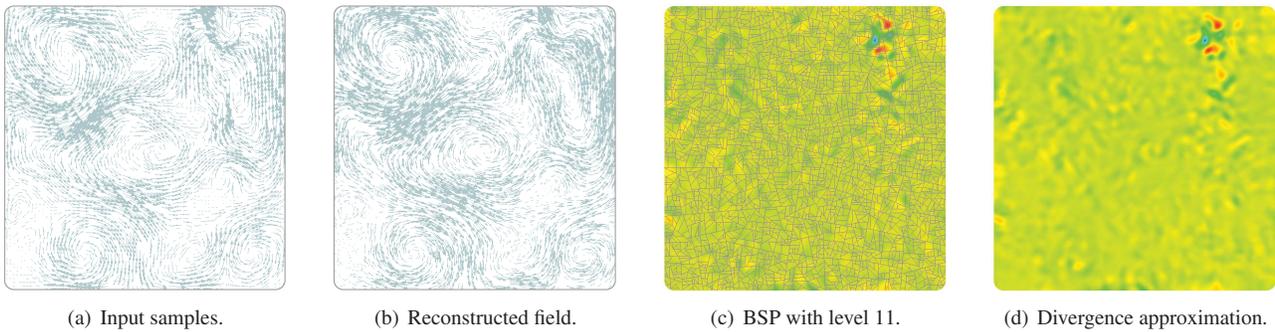


Figure 10. Divergence of a vector field from a fluid simulation, using [17] ($\lambda = 125$): the null divergence of the field is well respected.

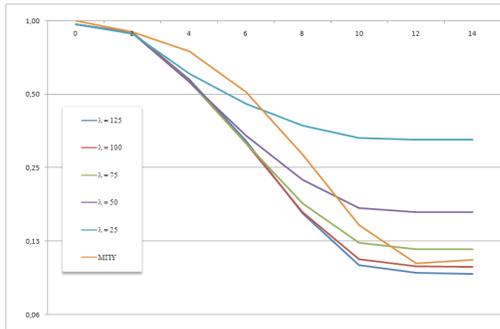


Figure 11. Relative error of the magnitude the field of Figure 10, for different values of λ and comparing with [12].

with points \mathbf{p}_i and normals \mathbf{n}_i , we create two point sets $\{\mathbf{p}_i - \epsilon \mathbf{n}_i\}$ and $\{\mathbf{p}_i + \epsilon \mathbf{n}_i\}$, and associate them to values -1 and $+1$ respectively. The reconstructed distance fields are shown in Figure 12. We can observe that the field is robust far from the points and does recover topological features of the curve, although the singularities almost induce spurious components in the reconstruction.

	KD-tree	BSP
# leaves	11505	10809
Maximal tree level	17	16
Magnitude mean relative error	0.3860	0.3449
Phase mean relative error	0.3342	0.3207

Table 1. Comparison with KD-tree approximations on the example of Figure 7, with $\lambda = 1.5$, error tolerance 10^{-3} and 11652 input samples.

5. Conclusion and future works

In this work, we introduce a smooth approximation mechanism that does not require global optimization. It relies on a combination of binary-space partition and smooth transition regression trees, leading to a simple plane-basis function modeling. We show results on scalar and vectorial approximations, outperforming previous approaches. We further demonstrate applicability in specific contexts such as vector fields' feature detection or curve reconstruction.

We intend to extend this work in higher dimensions, which is straightforward in the implementation. Moreover, we plan to further study BSP refinement criteria based on

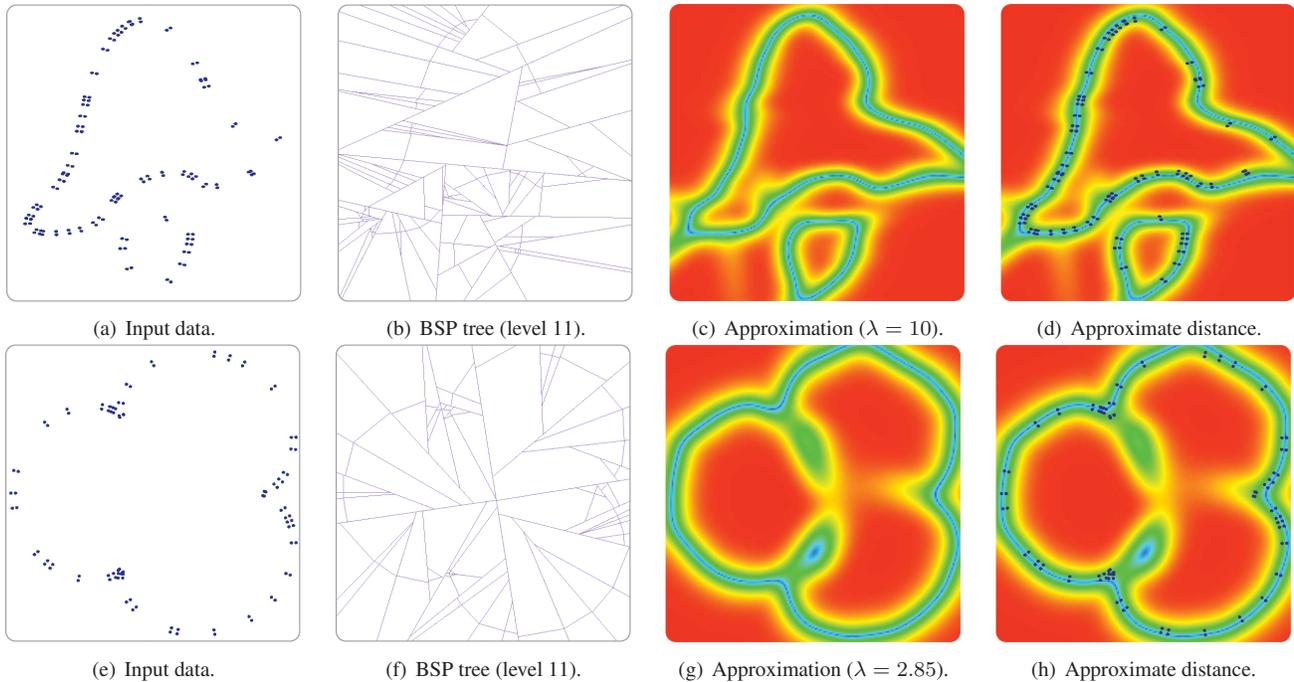


Figure 12. Curve reconstruction from a scalar plane-based function modeling.

local, non-linear optimization. Finally, the transition parameter λ may be adjusted to each BSP interior node, which may lead to even greater adaptation.

Acknowledgements. The authors would like to thank the FAPERJ and CNPq for financial support during the preparation of this paper, through projects CNPq 05/2005 and MCT/CNPq 02/2006.

References

- [1] I. Babuska and J. M. Melenk. The partition of unity method. *Numerical Methods in Engineering*, 40(4):727–758, 1997.
- [2] A. Bordignon, T. Lewiner, H. Lopes, G. Tavares, and R. Pereira. Point set compression through BSP quantization. In *Sibgrapi*, pages 229–236. IEEE, 2006.
- [3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1983.
- [4] J. Carr, R. Beatson, J. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Siggraph*, pages 67–76. ACM, 2001.
- [5] J. C. da Rosa, Á. Veiga, and M. Medeiros. Tree-structured smooth transition regression models. *Computational Statistics & Data Analysis*, 52(5):2469–2488, 2008.
- [6] S. Dekel and D. Leviatan. Adaptive multivariate approximation using binary space partitions and geometric wavelets. *Journal of Numerical Analysis*, 43(2):707–732, 2005.
- [7] M. Farias, C. Scheidegger, J. Comba, and L. Velho. Boolean operations on surfel-bounded objects using constrained bsp-trees. In *Sibgrapi*, pages 325–333. IEEE, 2005.
- [8] H. Fuchs, Z. M. Kedem, and B. F. Naylor. On visible surface generation by a priori tree structures. *Siggraph*, 14(3):124–133, 1980.
- [9] C. W. J. Granger and T. Teräsvirta. *Modeling Non-Linear Economic Relationship*. Oxford University Press, 1993.
- [10] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [11] H. Kim and W. Loh. Classification Trees with Unbiased Multiway Splits. *Journal of the American Statistical Association*, 96(454), 2001.
- [12] M. Lage, F. Petronetto, A. Paiva, H. Lopes, T. Lewiner, and G. Tavares. Vector field reconstruction from sparse samples with applications. *Sibgrapi*, pages 297–306, 2006.
- [13] S. Murthy. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, 2(4):345–389, 1998.
- [14] M. Pauly and M. Gross. Spectral processing of point-sampled geometry. *Siggraph*, pages 379–386, 2001.
- [15] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *Transactions on Image Processing*, 9(4):561–576, 2000.
- [16] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley, 1990.
- [17] J. Stam. Stable fluids. *Siggraph*, pages 121–128, 1999.
- [18] C. Stoll, H.-P. Seidel, and M. Alexa. Bsp shapes. In *Shape Modeling and Applications*, page 42. IEEE, 2006.
- [19] W. Thibault and B. Naylor. Set operations on polyhedra using binary space partitioning trees. *SIGGRAPH*, 21(4):153–162, 1987.