HHS Public Access

Author manuscript

Vol Graph. Author manuscript; available in PMC 2015 August 14.

Published in final edited form as:

Vol Graph. 2010 ; : 69–76. doi:10.2312/VG/VG10/069-076.

Multi-dimensional Reduction and Transfer Function Design using Parallel Coordinates

X. Zhao and **A. Kaufman**

Stony Brook University, USA

Abstract

Multi-dimensional transfer functions are widely used to provide appropriate data classification for direct volume rendering. Nevertheless, the design of a multi-dimensional transfer function is a complicated task. In this paper, we propose to use parallel coordinates, a powerful tool to visualize high-dimensional geometry and analyze multivariate data, for multi-dimensional transfer function design. This approach has two major advantages: (1) Combining the information of spatial space (voxel position) and parameter space; (2) Selecting appropriate high-dimensional parameters to obtain sophisticated data classification. Although parallel coordinates offers simple interface for the user to design the high-dimensional transfer function, some extra work such as sorting the coordinates is inevitable. Therefore, we use a local linear embedding technique for dimension reduction to reduce the burdensome calculations in the high dimensional parameter space and to represent the transfer function concisely. With the aid of parallel coordinates, we propose some novel high-dimensional transfer function widgets for better visualization results. We demonstrate the capability of our parallel coordinates based transfer function (PCbTF) design method for direct volume rendering using CT and MRI datasets.

1. Introduction

Transfer functions (TF), as an important classification method, have been proposed to produce images that display or highlight the region of interest in the dataset. Some significant experience has been accumulated on how to identify the accurate boundaries between different materials, such as 1D scalar values based TF and various 2D TFs with respect to gradient magnitude, directional first and second derivative [KD98], curvature [KWTM03] and statistical measures [TLM01]. However, the specification of a transfer function to accurately identify different objects in a complex volume dataset is still a challenging task.

In this paper, we propose a novel multi-dimensional (nD) TF design method, termed parallel coordinates based transfer function (PCbTF) design method, and a new dimension reduction technique to simplify the TF design using local linear embedding (LLE) [RS00]. LLE maps its inputs into a single global coordinate system of lower dimensionality, and thus its optimizations do not involve local minima, which leads to the ability of learning the global structure of nonlinear manifolds (details in Section 3.3). A simple but effective user interface has also been developed to assist in the nD TF design. Through our framework, a carefully designed nD TF can emphasize details which are difficult to visualize by other approaches.

The pipeline of PCbTF is shown as Figure 1. First, for each voxel of input dataset, various high dimensional parameters are calculated. Next, parameters are selected according to the patterns of corresponding polylines drawn in PC. For the high dimensional TF design, the user can choose to either interactively design special widgets on the coordinates directly or automatically project all the attribute parameters to the 2D space by the LLE technique as dimension reduction, and then assign colors and opacities to the classes calculated by a k-mean algorithm in the 2D space.

Following the pipeline, our paper is organized as follows: Section 2 presents the related work. Section 3 describes our approach in details. Section 4 discusses the implementation. Section 5 analyzes results obtained by our method. Section 6 draws the conclusions and points out future work.

2. Related Work

Multi-dimensional transfer function design

The transfer function design, especially the multi-dimensional TF design, has become a fundamental and important research thrust. Kniss et al. [KKH02] have introduced dual domain interaction to facilitate identification of 3D boundaries using a probe that facilitates manual segmentation of various materials. Roettger et al. [RMS05] have used the voxel barycenter and the region variance to assist in manual specification of colors for similar features in the process of volume rendering. Canban et al. [CR08] have used first-, second-, and high-order local statistical texture properties to effectively assign voxels to different opacities and colors using texture-based transfer function. Maciejewski et al. [MWCE09] have proposed a novel non-parametric clustering method to design TF. Although only 2D TF shown as examples, the clustering method can also be extended to design nD TF. For the high dimensional computation, He et al. [HHKP96] and Marks et al. [MAB*97] have proposed a solution to the parameter selection problem, where by the user choosing TF by browsing through many rendered images. Tzeng et al. [TLM05] have presented a new approach to the volume classification problem, relying on an intelligent system to abstract high dimensional mapping functions from the user. For all the above methods, however, the user only specifies the classification in parameter space directly, limiting the visualization effects of data properties that can be used simultaneously. As an alternative, we propose to apply parallel coordinates (PC), a powerful nD parameters visualization and data analysis tool.

Parallel coordinates based transfer function design

Parallel coordinates, first presented by Inselberg et al. [ID90], is a helpful nD visualization tool. For high dimensional data visualization and visual clustering, Lum et al. [LSM06] have used local textures, scale-based filtering, and parallel coordinates to better classify volume data interactively. The effectiveness of this approach is limited by how to sort the PC to identify the different materials. Fua et al. [FWR99] have programmed a useful software XmdvTool, which provides a multi-resolution view of the data via hierarchical clustering, and uses a variation on PC to convey the aggregation information for clusters. Zhou et al. [ZYQ*08] have proposed an optimization scheme designed to minimize the curvature of the

polyline edges and maximize the parallelism of adjacent edges through an energy function. Tory et al. [TPM05] have presented a user interface, based on PC, to facilitate the exploration of volume data. By explicitly representing the visualization parameter space, the interface offers an overview of rendering options and enables the user to easily explore different parameters.

Combining the above two ideas, Pradhan et al. [PBM05] have proposed a novel approach that allows the user to analyze signatures/parameters of data volumes using PC. Parallel coordinates provides an interactive interface for the structure and feature visualization, nD TF design and segmentation. In contrast, we add several new techniques: (1) The spatial information is added as new parameters, which can maximally embed the prior-knowledge, especially for medical datasets; (2) We can analyze polyline structural patterns in the PC to manually remove the correlated parameters as direct dimension reduction (subsection 3.2.1) or automatically apply dimension reduction using LLE; (3) For the TF design, instead of using only one signature/parameter to analysis the volumetric dataset, we design a real high dimensional TF (subsection 3.2.2), considering nD parameters together. Our method is able to produce better classification and visualization results for some difficult tasks.

Dimension reduction based transfer function design

Dimension reduction is an alternative solution for the high dimensional TF design. Takanashi et al. [TLM02] have used independent component analysis (ICA) for nD parameter reduction. Rezk-Salama et al. [RSKK06] have created models from several training datasets by principle component analysis (PCA) to reveal the desired structures. Pinto and Freitas [dMPF07] have applied self-organizing maps (SOMs) and radial basis functions (RBFs) to simplify the design of nD TF to achieve the accurate classification. For the dimension reduction, in this paper, we introduce LLE, a good nonlinear high dimension reduction method, which could effectively simplify the complex polyline analysis.

3. Design of Multi-dimensional Transfer Functions using Parallel Coordinate

In this section, we describe the following procedure to directly design PCbTF. After the data preparation and parameter extraction, we apply a simulated annealing (SA) method [KGV83] to find the best sorting order of all the parameters in the PC with respect to energy minimization. Then, the correlated parameters are removed according to mathematically defined patterns of polylines. Next, several novel widgets are proposed to help the user to design PCbTF. On the other hand, LLE, a good dimension reduction technique, is represented as an automatical method for the design of nD TF.

3.1. Data Preparation

For most volumetric datasets, two major problems, background and image noise, will significantly affect the visualization and computational speed of polylines. Simply removing the background voxels does not influence the information of the feature of interest, but will significantly decrease the computational time and operation complexity. Therefore, we implement the region growing method with several selected seeds to remove the air voxels

around the objects, such as in the CT or MRI dataset. After the background elimination, sixteen statistical attributes (angular second moment, contrast, correlation, variance, inverse difference moment, individual entropy, sum average, sum variance, sum entropy, skewness, kurtosis, correlation information measurements, intensity, gradient and second order derivative) are extracted and drawn as coordinates in the PC (following the feature equations defined in [CR08] and [HSD73]). We apply the concept of an outlier [NH06] to remove noise. Randomly distributed noises are simply removed by erasing the outlier polylines.

3.2. Transfer Function Design in PC

In this part, we focus on parameter sorting, selection and TF widget design using PC.

3.2.1. Visualization of Parameter Sorting and Selection—For the visual clustering in the PC, various methods have been published such as blending [JLJC05] and scatter plot matrix [GTD01]. We mainly implement the brush function to select, highlight and erase the polylines using the Xmdv-Tool library [xmd]. One of the most important motivations of PC is to best reveal the relationship/correlation of neighboring coordinates. However, the PC sorting problem, as NP-complete problem, has no efficient algorithm (with running time O(n!)). Peng et al. [PWR04] have provided random swapping, nearest neighbor and greedy methods to solve this problem. However, random swapping suffers from serious repetition problem, while nearest neighbor and greedy algorithms are easily stuck at the local optimization. Thus, we modified SA to find a fast optimization solution (with running time $O(n^p 2^n)$). First, new equations are defined as the clutter measurement to describe the internal energy $f(h)$ between the neighboring coordinates: $f(h)=\frac{1}{\sum_{i=1}^{n}(p_i-\frac{1}{N})^2}$, where $N = N_1 * N_2$, N_1 and N_2 are the pre-defined bin number of adjacent coordinates; p_i is the 2D joint histogram distribution probability, $p_i=\frac{l_i}{\sum_{i=1}^{N} l_i}$, while l_i is the total polyline number in the i_{th} bin.

This formula describes the basic idea: the more polyline aggregation in a single bin, the less internal energy. Algorithm 1 lists the details and conditions during the implementation. This method can quickly reach the global optimal solution using appropriate parameters, as shown in Figure 2. The best sorting order makes it easy to select important parameters: various correlated patterns (shown in Figure 3) can be identified and removed according to precise definitions by Inselberg [Ins09].

3.2.2. Widgets Design of PCbTF—Parallel coordinates preserves properties of a hypersurface by polyline patterns, which makes the TF design easy. Figure 4 shows the corresponding patterns between traditional orthogonal space (2D, 3D and nD) and PC. In addition to these basic patterns, we design various primitives as TF. Figures 5a–f show the design ideas in PC and the corresponding modifications in the 2D or 3D space. In order to adjust the opacity smoothly, we design two basic filter widgets (Gaussian and Laplacian) as shown in Figures 5g and h. All the designed patterns or primitives can be easily extended to nD by setting constraints between the neighboring coordinates: $x_2 = f_1(x_1), x_3 = f_2(x_2), \ldots, x_n = f_{n-1}(x_{n-1})$, where the functions are pre-defined by the user. Therefore, the adjustment of x_1 will automatically control the selection of other parameters. Through using the widgets, the

complicated design of high dimensional TF, which is impossible to be drawn on the screen by traditional methods, is easily implemented in the PC.

3.3. Dimension Reduction

Although the PC can directly assist the design of high dimensional TF, the major limitations are the facts that large datasets or parameter axes cause difficulty in the interpretation for the accurate classification, and relationships are only preserved between adjacent coordinates. Resorting the coordinates is an extremely time-consuming task especially for very high dimensions. Therefore, dimension reduction is motivated for the design of nD TF. We apply local linear embedding method (LLE) [RS00], an unsupervised learning algorithm that computes the low-dimensional, neighborhood-preserving embedding of high-dimensional inputs. LLE has several predominances: (1) It eliminates the need to estimate pairwise distances between the widely separated data points; (2) It maintains the global nonlinear structure from locally linear fits. By comparison with PCA [Jol89] and metric MDS [CC94], LLE is especially good at identifying the underlying complicated manifold structure. In this paper, we first implement LLE as a dimension reduction based TF design method. The algorithm is briefly described as follows: Suppose the data consists of N real-valued vectors $\overrightarrow{X_i}$, the LLE algorithm, is based on simple geometric intuitions with two cost functions:

$$\varepsilon(W)=\sum_i|\overrightarrow{X_i}-\sum_j W_{ij}\overrightarrow{X_i}|^2 \quad (1)$$

and

$$\varphi(Y)=\sum_i|\overrightarrow{Y_i}-\sum_j W_{ij}\overrightarrow{Y_i}|^2 \quad (2)$$

First, we compute the neighbor of each data point, $\overrightarrow{X_i}$. Then we compute the weights W_{ij} from the neighbor points of $\overrightarrow{X_i}$ by minimizing the cost in Eq.1. Last, we compute the best reconstruction by weights W_{ij}, through minimizing Eq.2 using its lowest nonzero eigenvectors.

The user can control two parameters of LLE, the number of neighbors and the dimension of the embedding space (default is 2D). After applying LLE, the embedded space tends to be very abstract and its meaning unclear. Fortunately, the high dimensional relationships among voxels can be perfectly preserved and can be easily classified by k-mean algorithm in 2D space, as shown in Figure 6a. Thus, the user just needs to assign color and opacity for each class to obtain the final rendering result, as shown in Figure 6b. Due to the loss and distortion of information usually caused by dimension reduction, LLE cannot provide quantitative information as accurate as PCbTF does. However, this approach is well suitable for revealing qualitative aspects such as shape of structures and clear dissimilarity between regions.

4. Implementation

We have implemented the parameter extraction and dimension reduction as off-line processes, but to maximize system interactivity, we accelerate our volume rendering and TF specification using graphics hardware. Given a parameter representation of each voxel, the task of designing PCbTF becomes specifying a mapping from the nD vector to a color and opacity value. The classification of each voxel forms a 3D texture stored in the frame buffer. New texture will be recomputed and redrawn on the screen by the GPU whenever TF changes. We calculate another 3D RGB texture to store colors and opacities picked by the user. The interface used to display polylines and design TFs is implemented using OpenGL and FLTK libraries. Rendering uses Cg on a desktop: Intel Xeon CPU 3.60GHz, 3GB memory and Nvidia GeForce GTX 285 graphic card. This approach is fast enough for our purposes of real time interaction to modify the design of TF and update the volume rendering results.

5. Results and Discussion

In order to demonstrate the ability of our framework to deal with challenging tasks, we have tested our PCbTF on CT and MRI datasets.

5.1. Application of PCbTF Design

Standard volumetric datasets—Figure 7 shows the rendering result of the CT carp dataset generated by our technique. Through the interface, the user can design a PCbTF as shown in Figure 7c to highlight the internal region of interest. Figure 7b shows the obviously visual enhancement of the swimming bladder by comparison with the result generated by 2D TF (shown in Figure 7a).

Medical volumetric datasets—Figure 8 shows our PCbTF results by comparison with other TFs for the CT bladder dataset. Figures 8a and b show that the 1D/2D TF will fail to distinguish accurate features from bladder and other tissues. We first apply the nD TF without any spatial information and receive a clear segment of bone structure, as shown in Figure 8c. However, for the bladder, as shown in Figure 8d, many “noise” points surround the bladder because these voxels are close to the class of bladder in the parameter space but far away in the spatial space, as shown in Figures 8e and f. We further refine the classification using the prior position information. Figure 8g shows the final rendering result using PCbTF designed as depicted in Figure 8h. The bladder is clearly separated from the surrounding tissues.

Another case study is the classification of MRI prostate dataset, an extremely challenging task for the common TF, as shown in Figure 9b. From the image, a 1D TF cannot truly separate the prostate out from its surrounding tissues due to the very similar parameter characteristics. Luckily, the medical information of the prostate such as location, size and shape can be easily found. We use the anatomical and spatial information to improve the result. First, we detect the colon, an organ with obvious features (large dark region as shown in Figure 9a). According to the anatomical knowledge, the colon is the nearest organ to the prostate along the posterior end, and thus we can estimate the possible position of prostate.

The position information becomes an important parameter combined with the other selected parameters (shown in Figure 9d) for the design of PCbTF to identify the prostate. Figure 9c shows the final result, with high opacity for the prostate and zero opacity for the surrounding tissues and organs. The above cases support the ability of PCbTF to clearly and accurately distinguish objects from surrounding tissues especially for the challenging datasets that tractional TFs fail.

Meanwhile, using the novel filter widgets (details in Section 3.2.2) of the PCbTF design, we build a method to organize opacity and color settings for a simultaneous view of multiple objects or substructures in the PC. The widgets can be used for bladder cancer detection. Figure 10a highlights the cancer area located at the bottom of bladder. We apply similar widgets in the T2 weighted MRI prostate dataset to display different zones of prostate. Figure 10b shows the final result.

5.2. Application of TF Design by LLE

For some datasets, more parameters are picked as independent parameters. For example, the CT Engine dataset has five parameters: intensity, gradient, entropy, correlation information and sum average, while the CT bonsai tree dataset contains six parameters: intensity, gradient, difference moment, skewness, variance and sum entropy. In order to simplify the design process, we project these parameters to the embedded 2D space by LLE, then implement k-mean classification and design a 2D TF. Although there is information loss during the dimension reduction, Figures 11a and b still show good classification results: the user can easily recognize the piston rings and piston rods for the engine dataset, and distinguish the soil, trunk and leaves for the bonsai tree dataset by assigning them different colors.

5.3. Timing and Performance

The main time-consuming part of our pipeline is the parameter extraction, which is related to dataset size, window size and selected parameters. Table 1 shows some statistics for all the datasets used in this paper. The table shows that the volume size and window size are major timing factors - time increases as the volume size grows while the window size decreases. Meanwhile, comparing with PCbTF, the design of LLE basd TF provides an easy operation and classification method in the 2D space, although some extra mapping time is needed. From the timing point view, the bottleneck is the interactive design of PCbTF. Therefore, total design time of LLE basd TF is small, which is a good choice when no specified objects are wanted by the user.

6. Conclusion and Future Work

In this paper, we have presented a novel technique to design high dimensional PCbTF and a dimension reduction scheme to construct nD TF using LLE. A simple but effective interface is provided for the user to interactively design TF to facilitate the discovery of the best classification scheme for complex datasets in the PC. Our method effectively allows the user to visualize local properties and spatial information together. The multivariate classification and its visualization could reduce the complexity of datasets and provide a vital connection

between the dataset and the analyst. Thus, it has clear advantages for the visualization of organs and even the cancer detection. However, the technique has some limitations. There is no easy way to effectively control and organize the opacity settings in the PC, and how to properly sort the PC is still a challenge. The dimension reduction method can simplify these problems, but it will cause some information loss.

Further work includes the effective scheme of assigning the opacity in nD TFs, the design of PCbTF for dynamic volume rendering, and the multi-resolution visualization via hierarchical fuzzy clustering. The automatic PCbTF design based on visual clustering algorithms [ZCQ*09] [ZYQ*08] [ADoL04] is also a promising direction.

Acknowledgments

This work has been partially supported by NSF grants CCF-0702699 and IIS-0916235 and a grant from USRF.

References

ADoL04. Artero AO, De oliveira MCF, Levkowitz H. Uncovering clusters in crowded parallel coordinates visualizations. IEEE Symp on Information Visualization. 2004:81–88.

CC94. Cox, T.; Cox, M. Multidimensional Scaling. Chapman Hall; London: 1994.

CR08. Caban JJ, Rheingans P. Texture-based transfer functions for direct volume rendering. IEEE Trans on Visualization and Computer Graphics. 2008; 14(6):1364–1371. [PubMed: 18988985]

dMPF07. de Moura Pinto F, Freitas CMDS. Design of multi-dimensional transfer functions using dimensional reduction. Eurographics IEEE-VGTC Symposium on Visualization. 2007:131–138.

FWR99. Fua Y, Ward M, Rundensteiner E. Hierarchical parallel coordinates for exploration of large datasets. IEEE Visualization. 1999:43–50.

GTD01. Gahegan M, Takatsuka M, Dai X. An exploration into the definition, operationalization and evaluation of geographical categories. International Conference on GeoCompu-tation. 2001:66–75.

HHKP96. He T, Hong L, Kaufman A, Pfister H. Generation of transfer functions with stochastic search techniques. IEEE Visualization. 1996:227–234.

HSD73. Haralick R, Shanmugam K, Dinstein I. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics. 1973; 3:610–621.

ID90. Inselberg A, Dimsdale B. Parallel coordinates: a tool for visualizing multidimensional geometry. IEEE Visualization. 1990:361–378.

Ins09. Inselberg, A. Parallel coordinates: visual multidimensional geometry and its applications. Springer; New York, NY, USA: 2009.

JLJC05. Johansson, J.; Ljung, P.; Jern, M.; Cooper, M. Revealing structure within clustered parallel coordinates displays. IEEE Symposium on Information Visualization; 2005. p. 125-132.

Jol89. Jolliffe, I. Principal Component Analysis. Springer Verlag; New York: 1989.

KD98. Kindlmann G, Durkin J. Semi-automatic generation of transfer functions for direct volume rendering. Volume Visualization. 1998:79–86.

KGV83. Kirkpatrick S, Gelatt C, Vecchi M. Optimization by simulated annealing. Science. 1983; 220:671–680. [PubMed: 17813860]

KKH02. Kniss J, Kindlmann G, Hansen C. Multidimensional transfer functions for interactive volume rendering. IEEE Trans on Visualization and Computer Graphics. 2002; 8(3):270–285.

KWTM03. Kindlmann G, Whitaker R, Tasdizen T, Moler T. Curvature-based transfer functions for direct volume rendering: Methods and applications. IEEE Visualization. 2003:513–520.

LSM06. Lum EB, Shearer J, Ma K-L. Interactive multi-scale exploration for volume classification. Vis Comput. 2006; 22(9):622–630.

MAB*97. Marks, J.; Andalman, B.; Beardsley, P.; Freeman, W.; Gibson, S.; Hodgins, J.; Kang, T.; Mirtich, B.; Pfister, H.; Ruml, W.; Ryall, K.; Seims, J.; Shieber, S. Design galleries" A general approach to setting parameters for computer graphics and animation. SIGGRAPH Conference; 1997. p. 389-400.

MWCE09. Maciejewski R, Woo I, Chen W, Ebert D. Structuring feature space: A non-parametric method for volumetric transfer function generation. IEEE Transactions on Visualization and Computer Graphics. 2009; 15(6):1473–1480. [PubMed: 19834223]

NH06. Novotny M, Hauser H. Outlier-preserving **F**ocus+ **C**ontext visualization in parallel coordinates. IEEE Transactions on Visualization and Computer Graphics. 2006; 12(5):893–900. [PubMed: 17080814]

PBM05. Pradhan, K.; Bartz, D.; Mueller, K. Tech Rep. Vol. 11. Dept. of Computer Science (WSI), University of Tubingen; 2005. Signature-Space: A Multidimensional, Exploratory Approach for the Analysis of Volume Data.

PWR04. Peng, W.; Ward, MO.; Rundensteiner, EA. Clutter reduction in multi-dimensional data visualization using dimension reordering. IEEE Symposium on Information Visualization; 2004. p. 89-96.

RMS05. Roettger, S.; MB; Stamminger, M. Spatialized transfer functions. Eurographics IEEE VGTC Symposium on Visualization; 2005. p. 271-278.

RS00. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000; 290(5500):2323–2326. [PubMed: 11125150]

RSKK06. Rezk-Salama C, Keller M, Kohlmann P. High-level user interfaces for transfer function design with semantics. IEEE Transactions on Visualization and Computer Graphics. 2006; 12(5): 1021–1028. [PubMed: 17080830]

TLM01. Tenginakai S, Lee J, Machiraju R. Salient iso-surface detection with model-independent statistical signatures. IEEE Visualization. 2001:231–238.

TLM02. Takanashi, I.; Lum, EB.; Muraki, S. Ispace: Interactive volume data classification techniques using independent component analysis. The Pacific Conference on Computer Graphics and Applications; 2002. p. 366-374.

TLM05. Tzeng F-Y, Lum EB, Ma K-L. An intelligent system approach to higher-dimensional classification of volume data. IEEE Trans on Visualization and Computer Graphics. 2005; 11(3): 273–284. [PubMed: 15868827]

TPM05. Tory M, Potts S, Moller T. A parallel coordinates style interface for exploratory volume visualization. IEEE Transactions on Visualization and Computer Graphics. 2005; 11(1):71–80. [PubMed: 15631130]

xmd. The homepage of xmdvtool–multivariate data visualization tool. www.davis.wpi.edu/xmdv/index.html

ZCQ*09. Zhou H, Cui W, Qu H, Wu Y, Yuan X, Zhuo W. Splatting the lines in parallel coordinates. Computer Graphics Forum/IEEE-VGTC Symposium on Visualization. 2009; 28(3):759–766.

ZYQ*08. Zhou, H.; Yuan, X.; Qu, H.; Cui, W.; Chen, B. Visual clustering in parallel coordinates; Computer Graphics Forum/IEEE-VGTC Symposium on Visualization; 2008. p. 1047-1054.

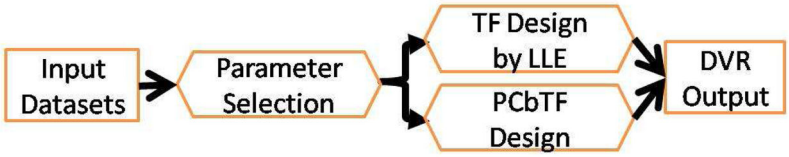

Figure 1.
An overview of our pipeline.

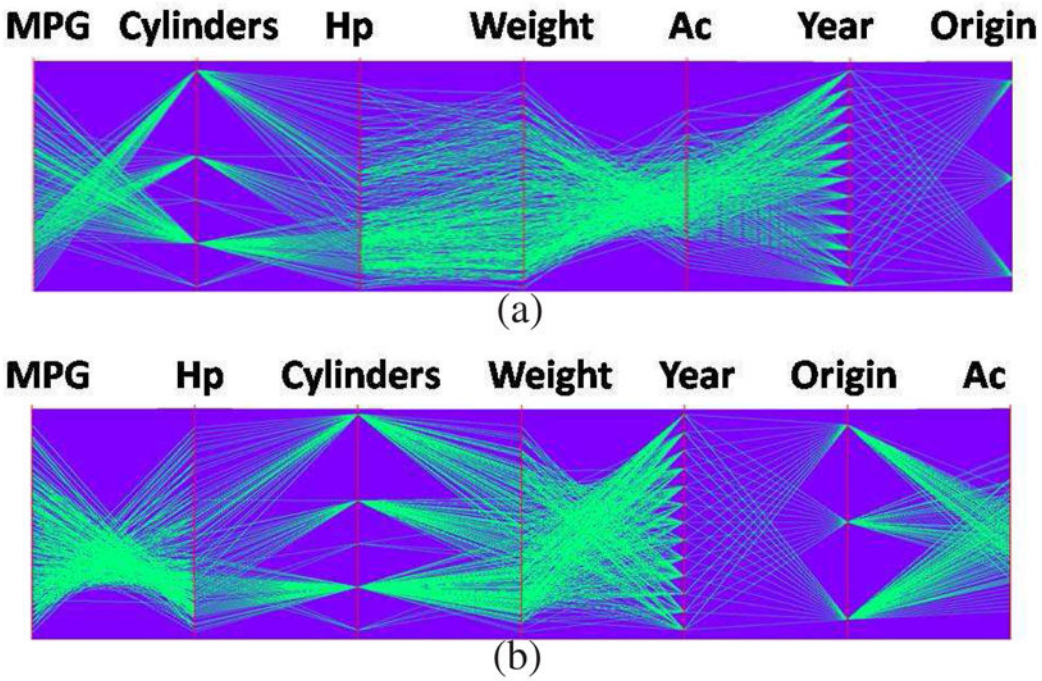

Figure 2.
Sorting results of the car dataset (7 dimensions, 392 items). (a) Random sorting sequence. (b) The same optimal sorting result generated by brutal search (T=20s) and our method (T=1.6s) with different computational time.

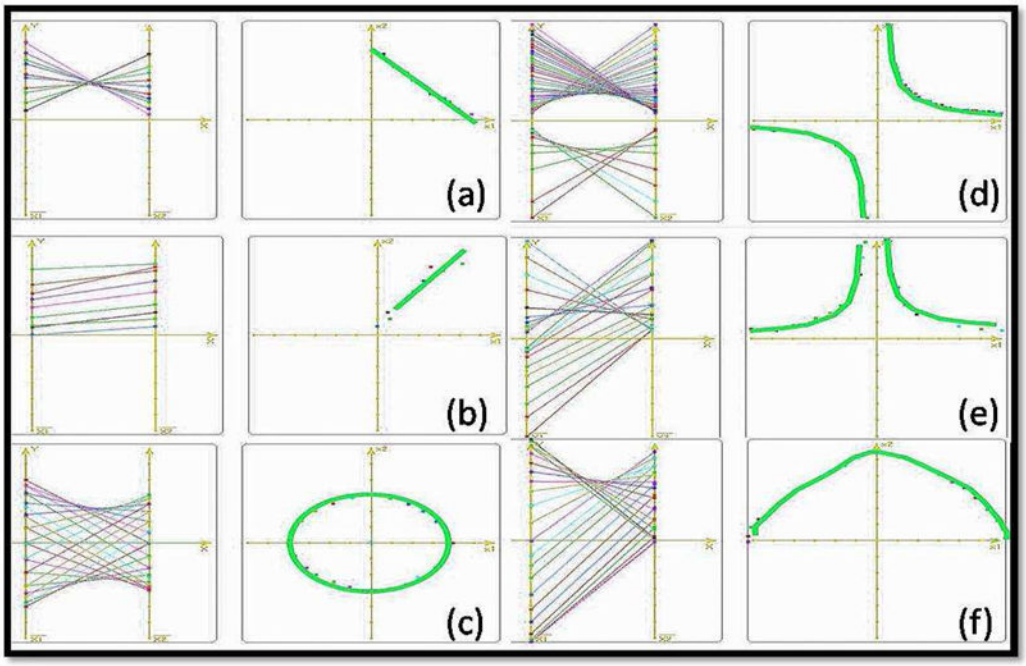

Figure 3.

Typical correlation patterns in the PC. (a) Inverse linear, (b) direct linear, (c) elliptical, (d) hyperbolic, (e) quadratic, and (f) parabola correlation.

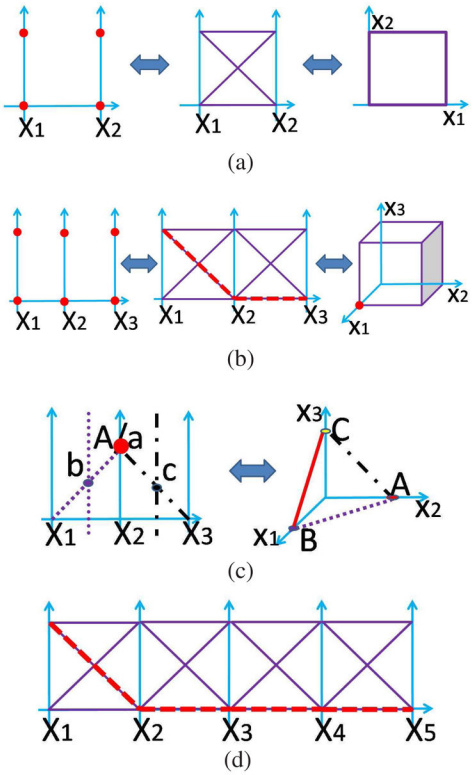

Figure 4.

Matching patterns between PC and orthogonal space. (a) 2D rectangle or square. (b) 3D cube, the dashed polyline is corresponding to the corner point of cube, with position (1,0,0). (c) 3D plane, the points of intersection of one dashed-line type in the PC is matching with the same dashed-line type in the 3D space. (d) 5D manifold, which is impossible to be displayed on the screen by the orthogonal coordinates. The dashed polyline is corresponding to (1,0,0,0,0).

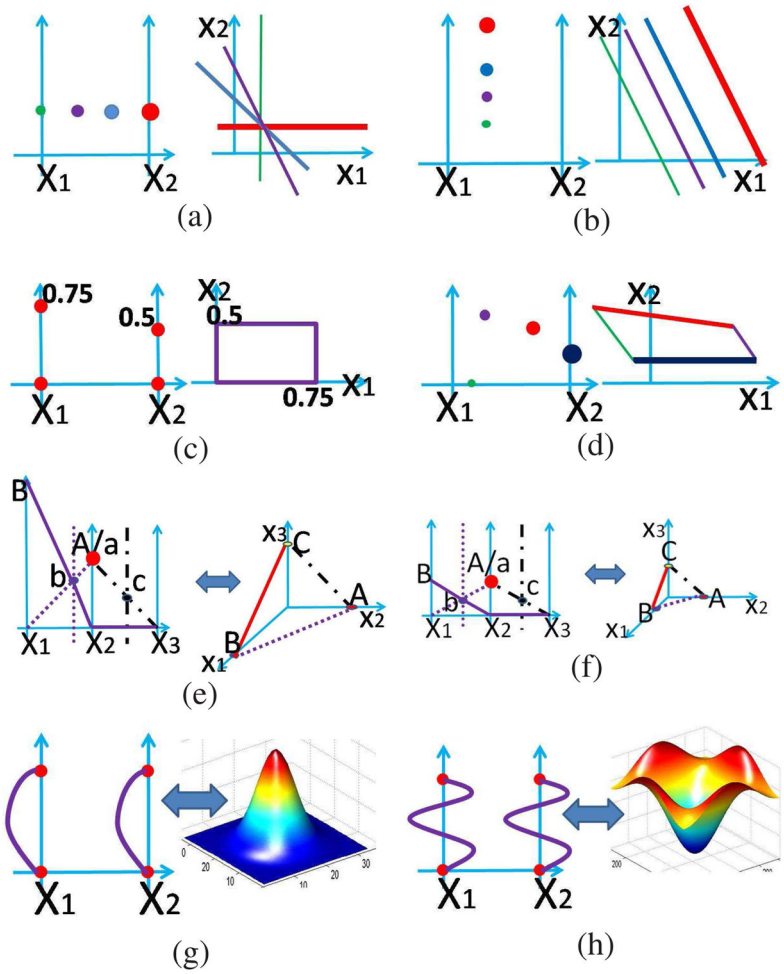

Figure 5.

PCbTF design widgets. The point sizes in the PC are matching with lines or planes of the same size/thickness in the 2D or 3D space. For a 2D line: y=a*x+b, (a) the adjustment of slope b; (b) the adjustment of rate a. For a 2D plane defined in Figure 4a, (c) 2D rectangle; (d) 2D any type polygon designed from the changes of control points. For a 3D plane defined in Figure 4c, (e) only change the vertical dashed line (point b as point of intersection) in the PC, matching with the adjustment of point B in the 3D space; (f) proportionately scale of the 3D plane (scale=0.5). Two 1D Gaussian (g) or Laplacian (h) curves are drawn in the neighboring coordinates, which are equivalent to the 2D Gaussian or Laplacian filter.

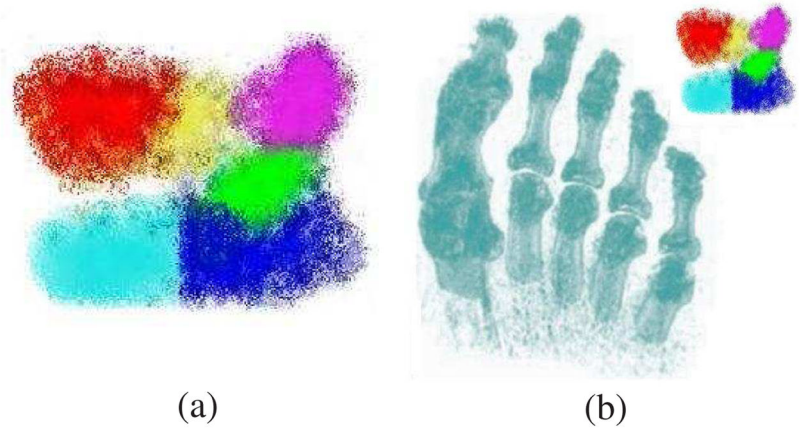

Figure 6.

TF design scheme using LLE. (a) The k-mean classes in the embedded 2D space for the CT foot dataset with five parameters: intensity, gradient, variance, entropy and angular second moment. (b) The rendering result by assigning high opacity to the classes of bone structure shown as dark and light blue regions (see (b) in the color section).

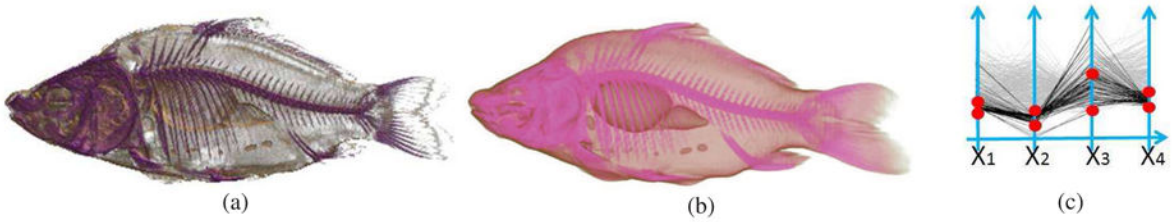

Figure 7.

Volume rendering results of the CT carp dataset generated by (a) 2D TF and (b) PCbTF. (c) The design of PCbTF with four parameters (X_1,…,X_4): intensity, gradient, sum variance and sum average (see also in the color section).

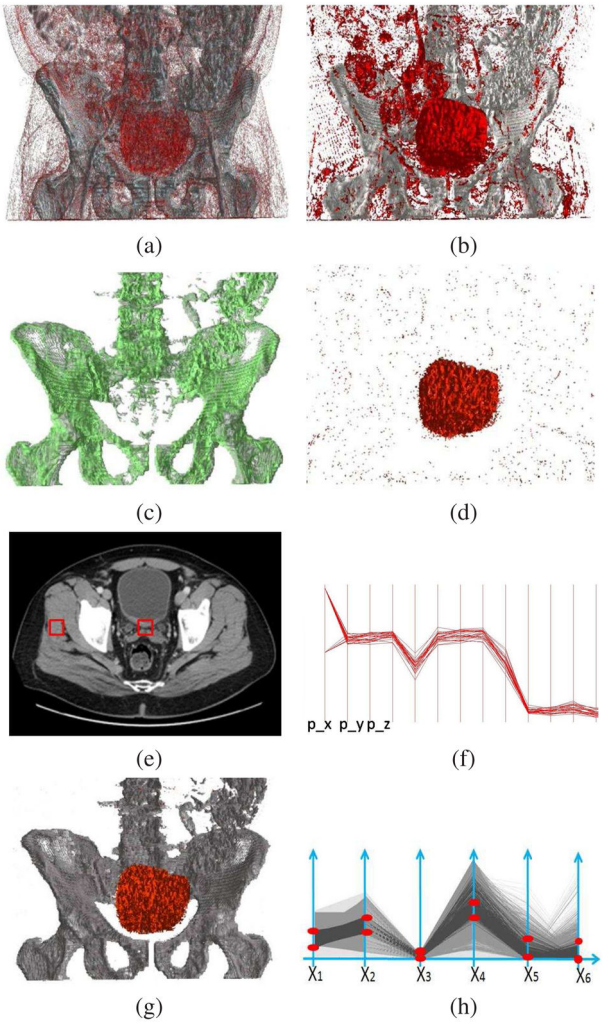

Figure 8.

Volume rendering results of the CT bladder dataset generated by (a) 1D TF and (b) 2D TF. Volume segment of (c) bone structure and (d) bladder generated by PCbTF in parameter space. Two similar parameter values (in boxes) in (e) a CT bladder slice but with different spatial values (pos_x) in the PC (f). (g) the result generated by PCbTF combined with spatial information. (h) The design of PCbTF with six parameters (X_1,…,X_6): position (x,y,z), intensity, gradient and entropy (see (a),(b),(g) in the color section).

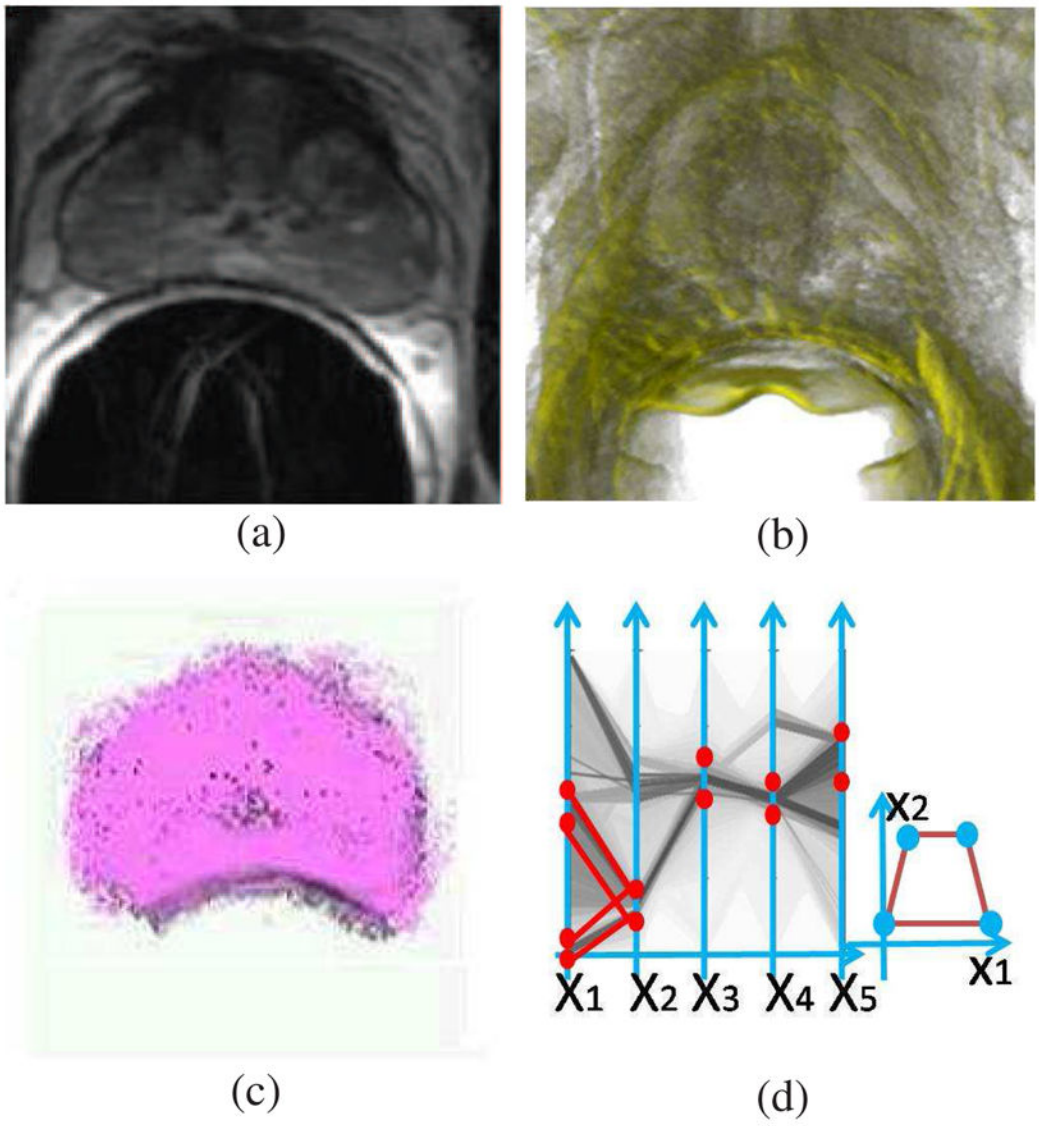

Figure 9.

Volume rendering results of the MRI prostate dataset generated by various TFs. (a) MRI axial slice of the prostate. (b) The result generated by 1D TF. (c) The result generated by PCbTF combined with anatomical knowledge and spatial information. (d) The design of PCbTF with five parameters (X_1,…,X_5): intensity, second order derivative, kurtosis, contrast and variance. Points are the boundary of the TF for each coordinate and small figure (right) shows the matching 2D pattern between X_1 and X_2 (see (b),(c) in the color section).

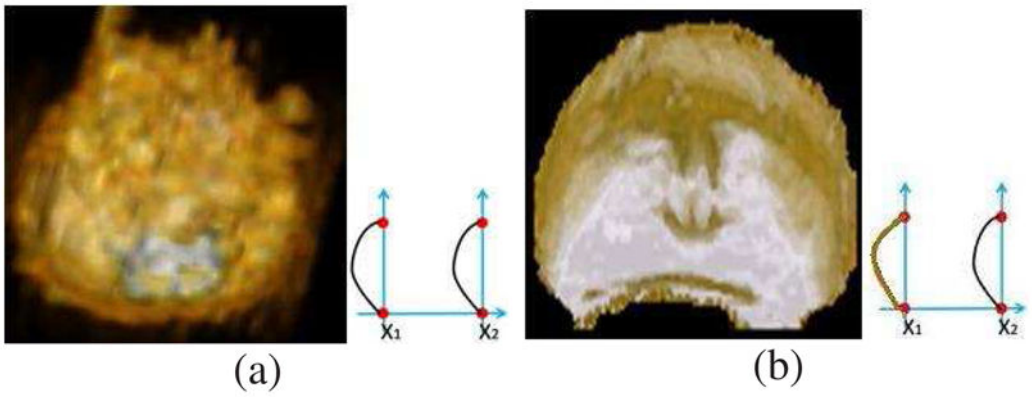

Figure 10.

Widget design in the PC for the visualization of sub-objects of interest. (a) Bladder cancer detection. Gaussian function is used on the intensity (X_1) and gradient (X_2) coordinates. Two black curves are designed to highlight features with high intensity and gradient using white color. (b) MRI prostate zones detection. Gaussian function is used on the intensity and gradient coordinates. The yellow curve enhances the opacity in the high intensity area and assigns yellow color while the black curve highlights the high gradient area and assigns white color (see also in the color section).

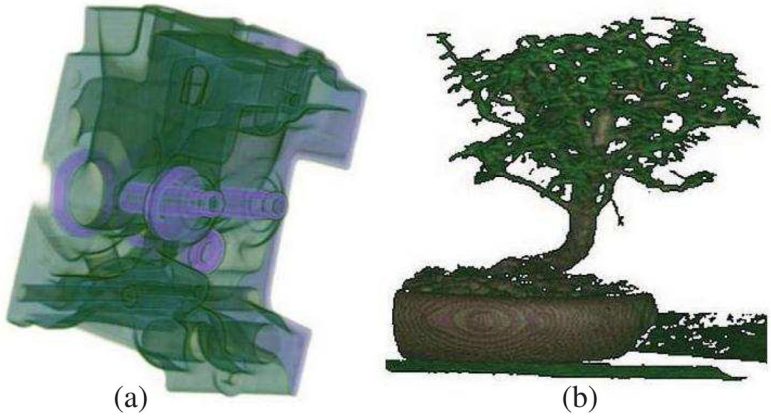

Figure 11.
Volume rendering results using LLE as dimension deduction for (a) the CT engine and (b) the CT bonsai tree datasets (see also in the color section).

Table 1

Statistics of various datasets presented in this paper. WS: window size, T.PE (sec): parameter extraction time and T.LLE (sec): local linear embedding time.

Model	Datasize	WS	T.PE	T.LLE
CT Carp	$256^2 \times 512$	9^3	43.54	–
CT Bladder	$256^2 \times 24$	$7^2 \times 3$	14.37	–
MRI Prostate	$128^2 \times 16$	$5^2 \times 3$	8.76	–
CT Foot	256^3	7^3	75.11	60.95
CT Engine	256^3	9^3	22.35	17.30
CT Bonsai	256^3	7^3	76.90	64.72

Algorithm 1

Simulated Annealing Algorithm

$T \rightarrow T_0$; $X \rightarrow X_0$ {Initial temperature, sequence}

$T_i \rightarrow T_0$; $X_i \rightarrow X_0$ {Initial the best solution X_i}

while $T_i > T_{min}$ **do**

i: $(w_1,\ldots,w_k,w_{k+1},\ldots,w_m,w_{m+1},\ldots) \rightarrow j$: $(w_1,\ldots,w_m,w_{k+1},\ldots,w_k,w_{m+1},\ldots)$ {For the random values k and m, setting $k < m$}

$df \rightarrow (f(Xj) - f(Xi))$ {Calculate energy difference}

if $df < 0$ **then**

$X_i \rightarrow X_j$

else if $df > 0$ and $\exp^{\frac{-df}{T_i}} > random()$ **then**

$X_i \rightarrow X_j$

end if

$T_i \rightarrow T_i - T_d$ {Decrease temperature}

end while

Return X_i