

The APDF Module

Version 2.0

<http://www.inf.unibz.it/dis/projects/3dvdm/>

Arturas Mazeika, Andrej Taliun, Michael Böhlen

October 10, 2005

Contents

1	Introduction	3
2	Installation	4
2.1	Hardware and Software Requirements	4
2.2	Download and Installation Instructions	5
2.3	Running	7
3	Getting Started	8
4	Structure of the GUI	10
4.1	The Data Selection Group	13
4.2	The Attributes, APDF Estimation Group	14
4.3	The Density Surface Group	16
4.4	The APDF Group	17
4.5	The Coordinate System Group	18
5	Illustrations of Density Surfaces	19
5.1	The Ball Dataset	19
5.2	The Spiral Dataset	20
5.3	The Plane-Hole Dataset	20
5.4	The Cone Dataset	20
6	Illustrations of APDF Trees	22

6.1	Plane-Cylinder Dataset	22
6.2	Cone Dataset	23
6.3	Linear Dataset	23
7	APDF Tree Computations	25
7.1	One-Dimensional APDF Tree	25
7.2	Two-Dimensional APDF Tree	26
7.3	A Complete Iteration of a Two-Dimensional APDF Tree	28
7.4	Three-Dimensional APDF Tree	29
8	Probability Density Functions and Density Surface Primer	31
8.1	Kernel Estimation	31
8.2	Density Surfaces	35

Chapter 1

Introduction

This manual describes the design and implementation of the 3DVDM Adaptive Probability Density Function (APDF) module, which is part of the 3DVDM system. Density information is crucial statistical information that can be used for a number of purposes. This manual illustrates how the APDF method with the help of density surface (DS) support visual data mining. The APDF module offers tools to compute and display density surfaces for 3D data.

The purpose of this manual is to describe the APDF module and related software components such that you can install the required components and run the APDF module. You get to know the functionality of the APDF module and learn how to use it.

Throughout the manual we use the term APDF module (Adaptive Probability Density Function) to denote the specific module of the 3DVDM system. We use the term DS (Density Surface) to denote a specific visualization tool of the APDF method.

The organization of the manual is the following. Chapter 2 describes the installation procedure of the VR++ and the 3DVDM systems. Chapter 3 introduces to the APDF module of the 3DVDM system. Chapter 4 discusses in detail the graphical user interface of the module. Chapters 5, 6, and 7 present screen shots of the APDF module. Finally, we give a primer on probability density functions and density surfaces in Chapter 8.

Chapter 2

Installation

2.1 Hardware and Software Requirements

There are no special hardware requirements to run the 3DVDM system. Essentially, the system should work on any hardware compatible with Linux. Note though that the software is computation and graphic resource intensive. For the exploration of large datasets (containing several million observations) we recommend 512MB memory, a fast graphical card, and a state-of-the-art CPU.

All experiments presented in this manual were produced on a 1GHz Pentium 4 PC with 512MB of RAM and a Geforce2 GTS 220 graphical card. The computation of the probability density function for good visual results typically requires 1–5 seconds. The 3DVDM System is able to visualize up to 100'000 objects (displayed as tetrahedra) and still ensure a smooth interactive navigation.

The APDF module is integrated into the 3DVDM System and was compiled and run on the following operating systems:

- RedHat Fedora Core Linux 4
- Debian Linux 3.1

Currently, RedHat Fedora Core 4 and Debian 3.1 are used as the main development platform for the APDF module.

To install the 3DVDM System one needs a Linux installation that includes support for the development libraries for OpenGL, GLUT, GTK, and GSL. Most Linux stock installations include all the necessary components.

2.2 Download and Installation Instructions

Below we give a minimal description of the installation procedure for Debian and Fedora that should also work for most other Linux installations. For more detailed information we refer to the installation instructions of the component packages [2, 3, 1].

Here we only provide the Debian and Redhat Fedora package names of the required software packages. The VR++ and 3DVDM packages can be downloaded from <http://www.inf.unibz.it/dis/projects/3dvdm>.

Essential Debian packages:

- lam4-dev
- freeglut3-dev
- libtiff4-dev
- libXmu-dev
- libgtk1.2-dev
- libgsl0-dev

Essential Redhat Fedora packages:

- lam
- freeglut-devel
- gsl-devel
- libtiff
- xorg-x11-devel
- gtk+-devel

Essential VR++ and 3DVDM packages:

- 3dvdm.tar.gz
- vr++.tar.gz
- 3dvdm-data.tgz

Unpacking:

```
$ tar xvfz 3dvdm.tgz
$ tar xvfz vr++.tar.gz
$ mv 3dvdm-data.tgz ~; cd ; tar xvfz 3dvdm-data.tgz
```

Installation:

```
$ cd vr++; make; cd ..
$ cd 3dvdm; make
```

Several notes are in order:

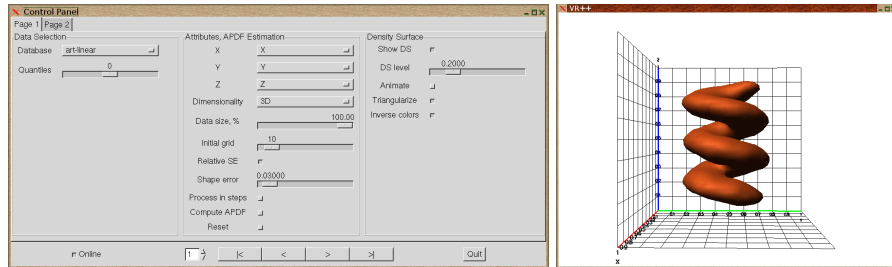
- Installation of Debian packages requires root privileges. Typically one needs to run the following command to install all required Debian packages:

```
# apt-get install lam4-dev freeglut3-dev \
libtiff4-dev libXmu-dev libgtk1.2-dev libgsl0-dev
```

- Installation of the Redhat Fedora packages requires the root privileges. Typically one needs to run the following command to install all the required Redhat Fedora packages:

```
yum install lam freeglut-devel gsl-devel \
libtiff xorg-x11-devel gtk+-devel
```

- The 3DVDM package (3dvdm.tgz) and the VR++ package (vr++.tar.gz) should be extracted into the same directory. The extraction will produce two directories: 3dvdm and vr++.



(a) Menu

(b) Visualization

Figure 2.1: The Windows of the 3DVDM-DS System

- The data files of the 3DVDM package (`3dvdm-data.tgz`) must be extracted in the home directory of the user. The extraction produces `VR++` subdirectory in the home directory of the user.
- If you have a non-standard setup you have to edit the user make files and adjust them as appropriate (cf. `user.makes/user.make.linux`)

2.3 Running

In order to run the APDF module, change the current directory to the binary directory of the 3DVDM system:

```
cd 3dvdm/bin
```

To start the APDF module simply type

```
./apdf
```

Because the 3DVDM System is implemented as a multi-process system there will always be two open windows: the menu of the APDF module (cf. Figure 2.1(a)) and the main window with the density surface and/or the respective data (cf. Figure 2.1(b)). Thus, in contrast to other window applications the menu is never closed and menu selections are immediately propagated. This last feature requires some care in order to prevent the display of intermediate (and thus unwanted) graphs.

Chapter 3

Getting Started

In order to compute and visualize a density surface one needs to complete four steps: (i) select a database, (ii) select three attributes of the selected database, (iii) compute the APDF tree for the selected attributes, and (iv) compute and visualize a density surface. All these steps are accomplished by selecting appropriate items on the GUI of the APDF module.

All the required elements are located on the Page 1 of the GUI (cf. Figure 3.1).

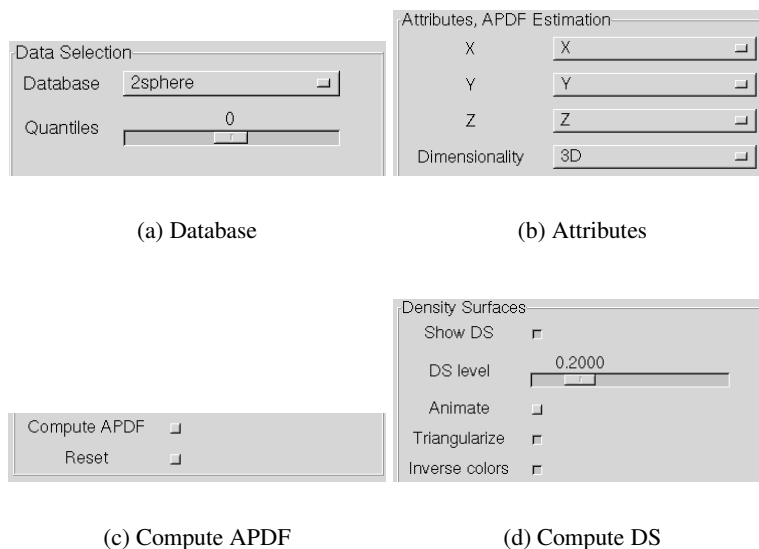


Figure 3.1: Four Steps of the Computation of a DS

In the first step we select the database for the analysis from the `Database` drop-down menu (cf. `Data Selection` group, cf. Figure 3.1(a)). This drop down menu contains all the databases available to the system for the analysis. The 3DVDM system locates all these datasets in `~/VR++/3dvdm-data` subdirectory in the users home directory. The datasets are comma separated values (CSV) files.

In the second step we select the X, Y, Z attributes of the database for the visualization (cf. `Attributes, APDF Estimation` group, Figure 3.1(b)). The drop-down menus contain the names of the attributes of the selected database.

In the third step we estimate the density function for the X, Y, Z attributes of the selected dataset. This is accomplished by pressing the `Compute APDF` check button (cf. `Attributes, APDF Estimation` group, Figure 3.1(c)).

In the fourth step we compute and visualize a density surface. A density surface is computed automatically once the APDF tree is build for the dataset. By changing the density level parameter (cf. `Density Surface` group, Figure 3.1(d)) one can compute and visualize the density surfaces of different density levels in real time.

The 3DVDM system tries to pre-select and guess as many parameters as it can. In most cases, it pre-selects the database, the attributes of the database, and the density level of the DS. The user only needs to press the `Compute APDF` button to get a visualization of the density surface.

Chapter 4

Structure of the GUI

The graphical user interface (GUI) of the APDF module (cf. Figures 4.1–4.2) consists of two pages of GUI elements, divided into six groups: Data Selection, Attributes, APDF Estimation, Density Surface (Page 1, cf. Figure 4.1), APDF, Scatter Plot, and Coordinate System Parameters (Page 2, cf. Figure 4.2).

Page 1 of the GUI contains all required parameters in order to compute and visualize a density surface. Page 2 contains supplementary parameters of the APDF module. The DataBase and Coordinate System groups are inherited from the abstract task class of VR++ and are documented in detail elsewhere.

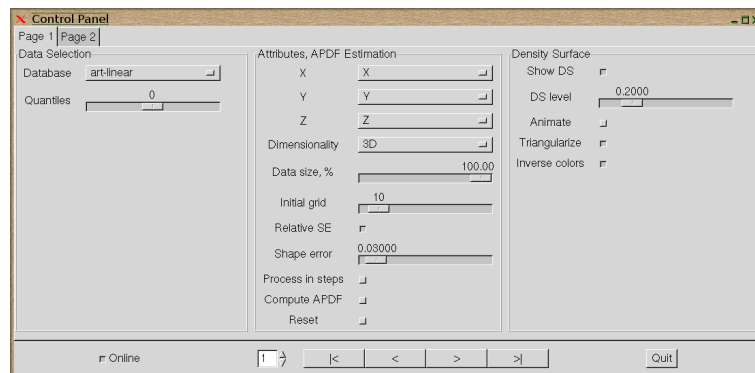


Figure 4.1: Page 1 of the GUI of the APDF Module

Page 1 of the GUI is organized according to the dependencies among the groups. For example the attributes (Attributes, APDF Estimation \rightarrow (X, Y, Z))

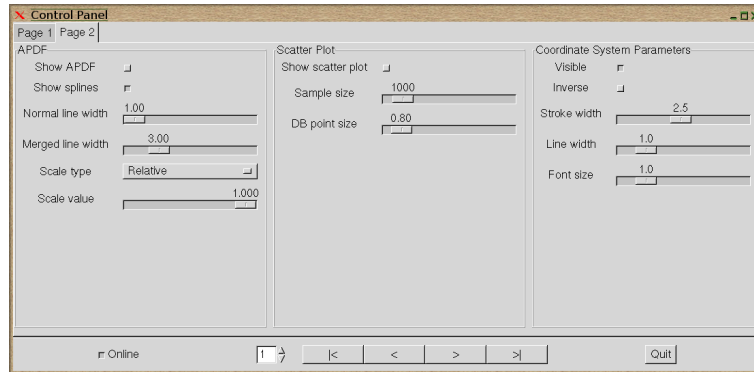


Figure 4.2: Page 2 of the GUI of the APDF Module

can only be selected if the dataset (`Data Selection` → `Database`) has been selected. Similarly, the computation and visualization of density surface must follow the estimation of the density (`Attributes`, `APDF Estimation` depends on `Data Selection`, and `Density Surface` depends on `Attributes`, `APDF Estimation`).

Page 2 of the GUI contains the visualization parameters of the APDF method. For example, by selecting appropriate parameters one can visualize the APDF tree, a sample of the dataset as a scatter plot, and switch on/off the visualization of the coordinate system. There are no dependencies between the groups of Page 2 of the GUI, and most of the elements can be selected in parallel. For example, the visualization of a scatter plot of a sample of the database (`Scatter Plot` group) can be shown together with the visualization of the APDF tree (`APDF` group).

Before we describe the different part of the GUI in detail we list and summarize all elements:

Data Selection Group

- `Database`. The name of the input data file.
- `Quantiles`. Controls the size of the coordinate system. For example, if a positive value v of quantiles is selected then $v\%$ of the data points will be visualized outside the coordinate cube.

Attributes, APDF Estimation Group

- `X, Y, Z`. The APDF module requires three spatial attributes (`X, Y, Z`). After the `Database` attribute (cf. above) has been selected the drop

down menus for X, Y, Z show all available attributes in the selected database.

- `Dimensionality`. The dimensionality of the APDF method. Possible values: 3D, 2D, and 1D. Note that all three spatial attributes must be selected (cf. above) independent of the dimensionality of the APDF method.
- `Data Size, %`. The number of data points to be processed by the APDF method in per cent.
- `Initial Grid`. The size of the initial partition per axe. If v is selected then the APDF method will start with v^3 uniform initial partitions.
- `Relative SE`. Selects whether the shape error is absolute or relative to the maximum of the PDF.
- `Process in steps`. If on, the computation of the APDF tree is suspended at the end of each sub-iteration (cf. `Compute APDF`). `Compute APDF` triggers the next step. If off, the whole APDF tree is computed without any interaction from the user.
- `Compute APDF`. Computes the APDF tree for the selected attributes of the selected database. If `Process in steps` is not selected then the whole APDF tree is computed. If `Process in steps` is selected then unselecting/selecting the element triggers the next sub-iteration of the APDF method.
- `Reset`. Deletes the current APDF tree. After this the APDF module is ready for a new computation of the APDF tree.

Density Surface Group

- `Show DS`. Computes and visualizes a density surface for the APDF tree, once the APDF tree has been computed.
- `Level of DS`. Density level of the density surface. This controls the density of the data points enclosed by the density surface.
- `Animate`. If checked the surfaces are animated automatically by varying the density level.
- `Inverse colors`. Inverses the colors of the coordinate system.

APDF Group

- `Show APDF`. If selected, visualizes the APDF tree. If `Process in steps` parameter (cf. `Attributes, APDF Estimation group`)

is selected then the intermediate sub-iterations of the computation of the APDF tree are visualized.

- `Show Splines`. Uses splines to visualize the PDF function.
- `Normal line width`. The thickness of the partition lines of the APDF tree. Applies only if `Process in steps` is selected.
- `Merged line width`. The thickness of the node lines of the APDF tree. Applies only if `Process in steps` is selected.
- `Scale type`. If `Relative` is selected, the values of the PDF are rescaled to the coordinate cube, otherwise no rescaling is done.
- `Scale value`. Scaling parameter of the PDF values.

Scatter Plot Group

- `Show scatter plot`. If selected, visualizes a sample of the database as a scatter plot.
- `Sample Size`. The size of the visualized sample.
- `DB point size`. The size of a visualized point.

4.1 The Data Selection Group

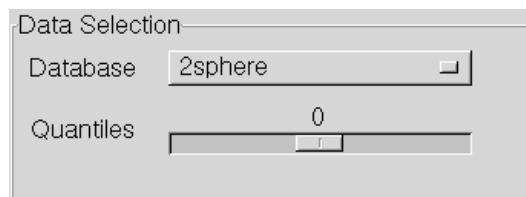


Figure 4.3: The Data Selection Group

The `Data Selection` group allows to select the data file that will be used for the analysis. The system scans for comma separated files (`.csv`) in the directory `$HOME/VR++/3dvdm-data` and lists them in the `DataBase` drop-down menu. If the dataset is being selected for the first time, the system pre-calculates basic statistical information (for example min, max of the data) and prepares the data for fast later access (for a more detailed discussion consult [2]).

`.csv` file stores information about the data. Each tuple is stored in a separate row; columns are separated by commas. The first row of the table stores the name

```
"X", "Y", "Z", "W"  
0.333245, 0.33945, 0.290604, 1  
0.679995, 0.19142, 0.521833, 1  
0.344678, 0.359624, 0.26787, 1
```

Figure 4.4: An Instance of a `.csv` File

information of the attributes (header). An example of a `.csv` file is illustrated in Figure 4.4. The database consists of four attributes (X , Y , Z , and W), and list three data records.

If the `Database` parameter is not selected the APDF module turns into a waiting state. No events are handled until `Database` and the three attributes X , Y , and Z (cf. Section 4.2) are selected.

`Changing Data Selection` triggers the recalculation of the `Attributes`, `APDF Estimation` and `Density Surface` groups.

4.2 The Attributes, APDF Estimation Group

The `Attributes`, `APDF Estimation` group selects the attributes of the selected database and computes the APDF tree for the selected data.

Once the `Database` (cf. Section 4.1) has been selected the `Attributes`, `APDF Estimation` group is updated with the new names of the attributes of the dataset. The APDF method requires X , Y , Z , and `Database` attributes to be selected before the calculation of the APDF tree. The APDF module turns into a waiting state if one of the parameters is not selected.

`Compute APDF` computes the APDF tree for the given dataset, `Shape error`, uniform initial partitioning of size `Initial grid per Axis`, and selected dimensionality `Dimension`. If `Global SE` is selected then the `Shape error` denotes the relative shape error wrt the maximum of the PDF. Otherwise, the parameter denotes the absolute shape error.

The APDF tree is computed iteratively. The APDF tree starts with a sparse uniform partition and identifies the region where the shape error is higher than `Shape error`. Addition of new partition points in these regions completes an iteration. The process is iterated until no more regions of too high shape error are found.

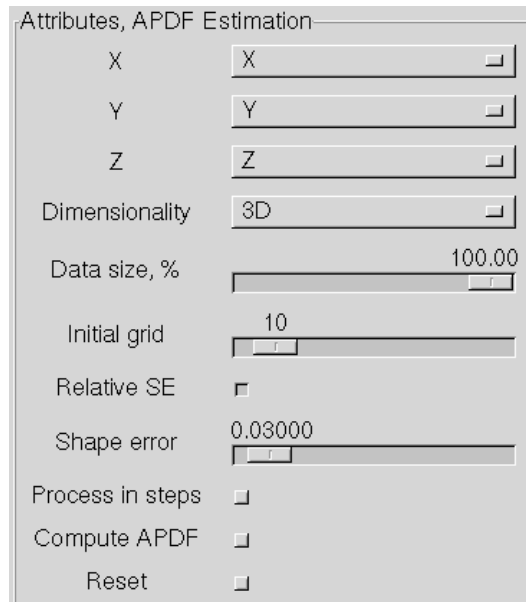


Figure 4.5: The Attributes, APDF Estimation Group

If `Process in steps` is selected then the computation of the APDF tree is suspended for input from the user at the end each sub-iteration. The user can then visualize the intermediate results of the construction of the APDF tree (cf. Section 4.4). There are five sub-iterations in one iteration of the APDF tree construction:

1. Apriori split (AS)
2. Tree optimization after AS (TO/AS)
3. Kernel additions (KA)
4. Posteriori split (PS)
5. Tree optimization after PS (TO/PS)

All sub-iterations are sequentially visualized if `Show APDF` parameter is selected (cf. Section 4.4).

The computation of the PDF tree outputs a tree computation summary to standard output. Figure 4.6 illustrates a typical output of the APDF module. The output consists of two blocks. At the beginning of the output the APDF module prints

```

It = iteration #
AS = apriori split # number of new leaves
TO/AS = tree optimization after apriori split # number of new leaves
KA = kernel addition # seconds
PS = posterior split # number of remaining leaves / removed leaves
TO/PS = tree optimization after posterior split # number of new leafs
Con = AS, TO/AS, PS, TO/PS time # seconds
OT = overall time # seconds
Err = shape error achieved after iteration

Initial step - KA: 0.02

It   AS   TO/AS  KA   PS   TO/PS  Con   OT   Err
1    64    1     0.02 64/0  1      0     0.02 0.440139
2    312   15     0.06 104/208 6     0     0.06 0.113925
3    376   27     0.19 240/136 30    0     0.19 0.121429
4    1118  64     0.46 178/940 30    0     0.46 0.0573202
5    886   57     0.55 0/886  0     0.01  0.56 0.0193337
---  ---  ---  ---  ---  ---  ---  ---  ---
Overall time: 1.31 s.
KA Time: 1.3 s.
AS, PS, TO Time: 0.01 s.
Apdf tree size: 17 kB. (17796 bytes)

```

Figure 4.6: Output of the APDF module

out the legend and the explanations of the abbreviations used in the measurements. The second block contains the actual table of measurements at the end of each sub-iteration and iteration.

4.3 The Density Surface Group

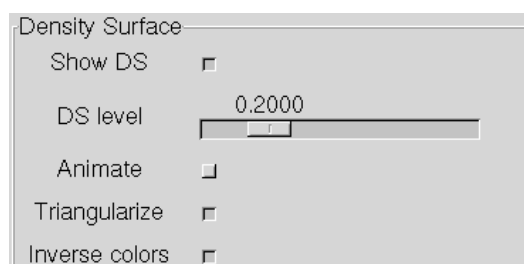


Figure 4.7: The Density Surface Group

A density surface is immediately visualized once the APDF tree is computed and the Show DS parameter is on. The DS Level determines the density level of the density surface. A high value will display a surface that encloses high-density data regions only.

If Animate is selected the density surfaces are animated. The APDF module

increases the DS Level incrementally, and computes and visualizes the density surfaces in animation. The animation of surfaces supports a comprehensive analysis of the data because it investigates and displays regions at varying density levels (more and less pronounced structures). This is particularly useful for getting a quick overview of the data.

The speed of the animation depends on the computation time required to compute a density surface.

4.4 The APDF Group

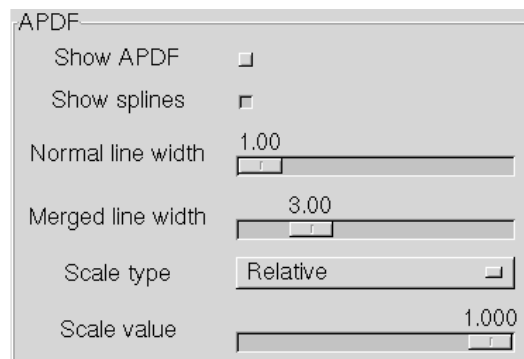


Figure 4.8: The APDF Group

The APDF group controls the visualization parameters of the APDF tree. If the `Show APDF` parameter is selected then the APDF module visualizes the APDF tree. If `Process in steps` parameter is selected (cf. Section 4.1) then the APDF module will visualize the intermediate results of the construction of the APDF tree.

The APDF module can visualize the final APDF tree (if the parameter `Process in steps` is not selected) for one-, two and three-dimensional cases. Examples of fully created one and two-dimensional APDF trees can be found in Section 6. An example of a visualization of one iteration of the APDF tree can be found in Section 7.3. Figure 7.3 illustrates all sub-iterations of one iteration.

Two parameters control the visualization of lines in the APDF tree. The thickness of partition lines is controlled by `Normal Line Width` parameter. The partition lines show where the space is split. The grouping of partitions of the same size into nodes that can be processed efficiently can be visualized with the `Merged line`

width parameter, which denotes the thickness of border lines of nodes.

4.5 The Coordinate System Group

The `Coordinate System` group allows to show/hide the coordinate system (`Visible`), invert the background (`Inverse`) change the thickness of grid lines (`Line width`), font size of the labels (`Font size`) and (`Stroke width`).

The functionality of the `Inverse` parameter is disabled. Instead, the user should use `Inverse colors` from the `Density Surface` menu to invert the background of the visualization.

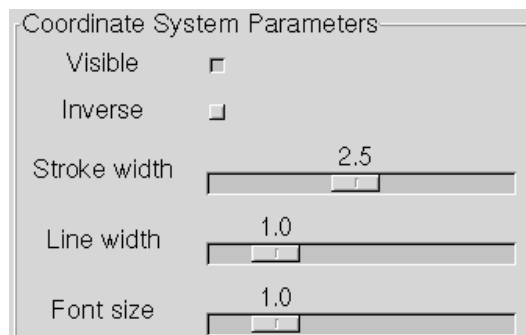


Figure 4.9: The Coordinate System Group

The group is primarily designed to create figures that can be printed on paper.

Chapter 5

Illustrations of Density Surfaces

In this chapter we give screen shots of density surfaces. Animations are displayed by a number of density surfaces of varying density side by side.

5.1 The Ball Dataset

The ball dataset `art-ball.csv` is a synthetic dataset simulated by a three-dimensional normal random variable. The highest density point is in the middle of the unit cube, and decreases as the distance from the center point increases.

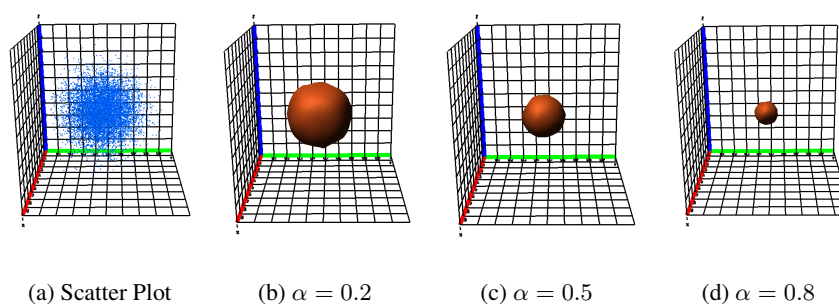


Figure 5.1: The Ball Dataset

Figure 5.1 shows a scatter plot and three density surfaces for the dataset for different density levels. As the density level increases the density surface envelops only the most dense data points.

5.2 The Spiral Dataset

The Spiral dataset (`art-spiral.csv`) is a synthetic dataset, where the data are generated around a spiral in the three-dimensional space. Figure 5.2 shows a scatter plot together with three density surfaces. Clearly, it is hard to see the structure from the scatter plot. The DS for $\alpha = 0.2$ envelops almost all data points. The structure is revealed as the level is increased to $\alpha = 0.8$.

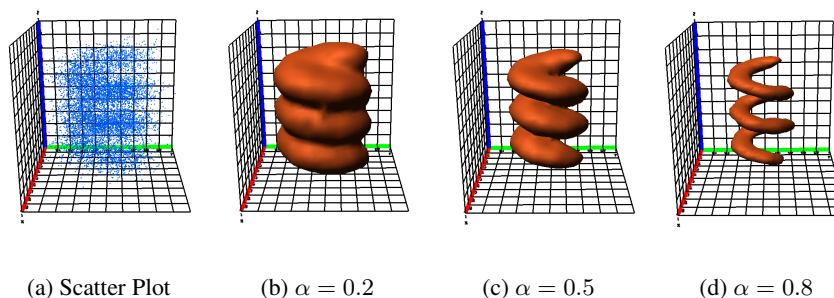


Figure 5.2: The Spiral Dataset

5.3 The Plane-Hole Dataset

The Plane-Hole (`art-plane-hole.csv`) dataset is a synthetic dataset, where the data are generated on a plane with a hole in it. Figure 5.3 shows a scatter plot together with three density surfaces. This figure illustrates that the triangularization of the density surfaces works perfectly fine with non-convex data structures.

5.4 The Cone Dataset

The Cone (`art-cone.csv`) dataset is a synthetic dataset, where the data resembles a cone in three-dimensional space. Figure 5.4 shows a scatter plot together with three density surfaces.

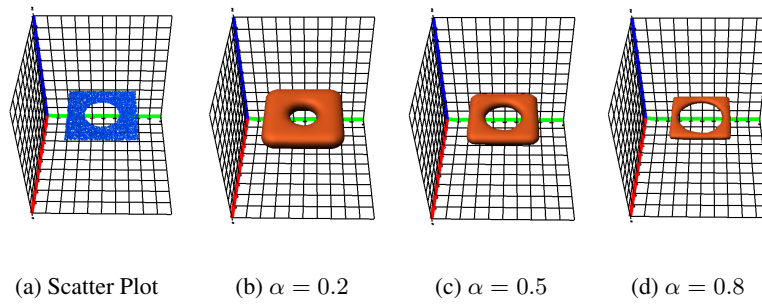


Figure 5.3: The Plane-Hole Dataset

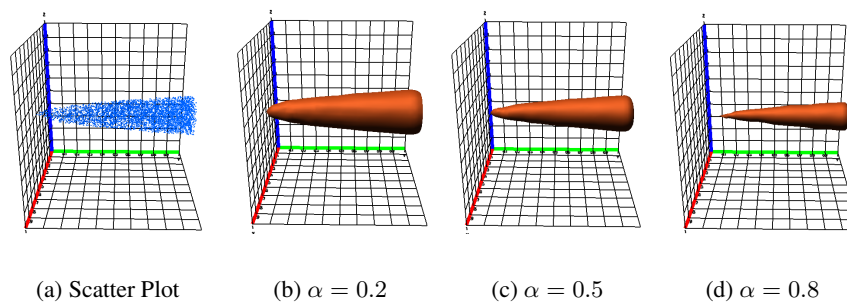


Figure 5.4: The Cone Dataset

Chapter 6

Illustrations of APDF Trees

This chapter illustrates one-, two-, and three-dimensional APDF trees at the end of all iterations (`Process in steps` is off).

6.1 Plane-Cylinder Dataset

Figure 6.1 illustrates the APDF trees for the `art-plane-cylinder` dataset. The PDF of the dataset is linear in most of the universe, therefore only a few partition points are needed to ensure the selected shape error there. More partition points are allocated for the cylinder to describe the non-linear PDF there.

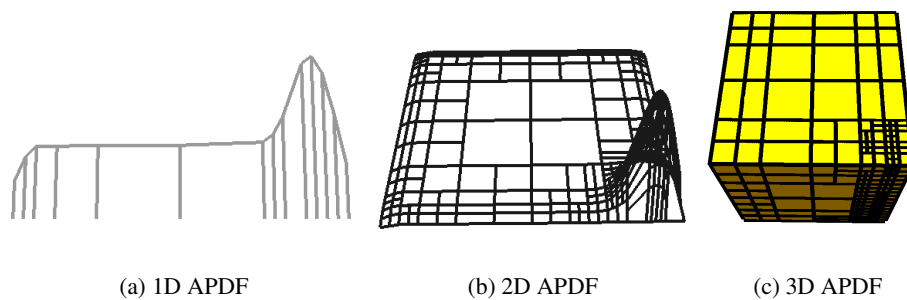


Figure 6.1: The Plane-Cylinder Dataset

In two- and three-dimensional cases the PDF is non-linear towards one dimension only in the areas towards outside of the universe. There the APDF method splits

the cubes in one direction only. This results in oblong rectangles/cubes (the height and the width of rectangles are not equal).

6.2 Cone Dataset

Figure 6.2 illustrates the APDF tree for the `art-cone` data points. The dataset consists of datasets visually similar to the cone.

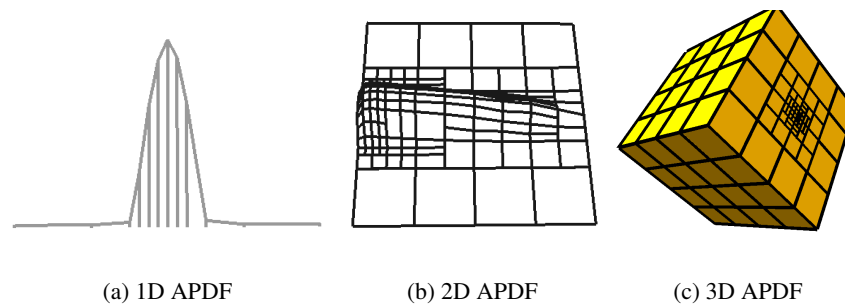


Figure 6.2: The Cone Dataset

Figure 6.2(a) shows one-dimensional APDF tree. Most of the partition points are allocated in the middle of the structure to cope with the non-linearity of the PDF. The two-dimensional APDF is illustrated in Figure 6.2(b). There most of the rectangles are oblong, since the PDF is non-linear only in the Y direction. The three-dimensional APDF is illustrated in Figure 6.2(c). Again, most of the splits are according to the X coordinate, and are visible only in the YZ plane.

6.3 Linear Dataset

Figure 6.3 illustrates the APDF trees for `art-linear` dataset. The distribution of the data is linear according to each coordinate. The density increases from corner $(0, 0, 0)$ towards the other corner of the universe, and then decreases very fast at the opposite borders of the universe.

The dataset illustrates the directional splits of the APDF tree very nicely (cf. Figures 6.3(b)–6.3(b)). The APDF does not allocate any partition points in the area of linearity of PDF, and introduces directional splits in the areas of non-linearity of the PDF.

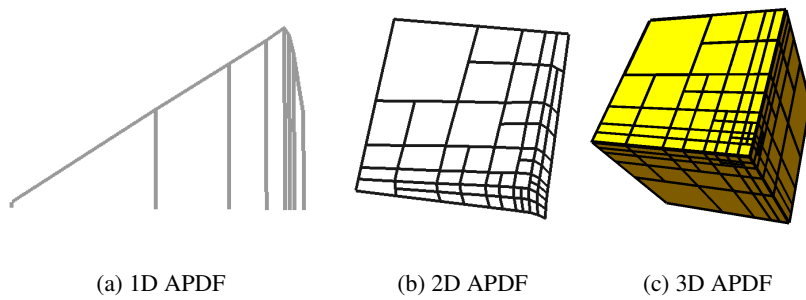


Figure 6.3: The Linear Dataset

Chapter 7

APDF Tree Computations

This chapter illustrates the construction of the APDF tree (`Process in steps` is on). Most of the illustrations show the creation of the tree as the number of iterations increases except Section 7.3, which illustrates one complete iteration of the APDF tree.

In the illustrations the colors have the following meaning. In the result of an a priori split yellow regions are the candidates of high shape error. Such regions are split and (later by the posteriori split) verified for the high shape error. In the result of a posteriori split green regions indicate that the shape error was indeed too high in the region.

7.1 One-Dimensional APDF Tree

Figure 7.1 shows a visualization of the APDF tree at the end of each iteration for normally distributed data. The approximation starts with a uniform initial partitioning of 4 points (cf. Figure 7.1(a)). It takes 5 iterations to construct a partition that ensures a uniform precision of the selected shape error (cf. Figure 7.1(b)–7.1(f)). In the first iterations most of the newly added partition points indeed decreased the shape error (cf. green areas, cf. Figure 7.1(b)–7.1(d)). In the last two iterations the areas where the shape error is below the threshold (cf. white and yellow areas, Figure 7.1(e)–7.1(f)) dominate.

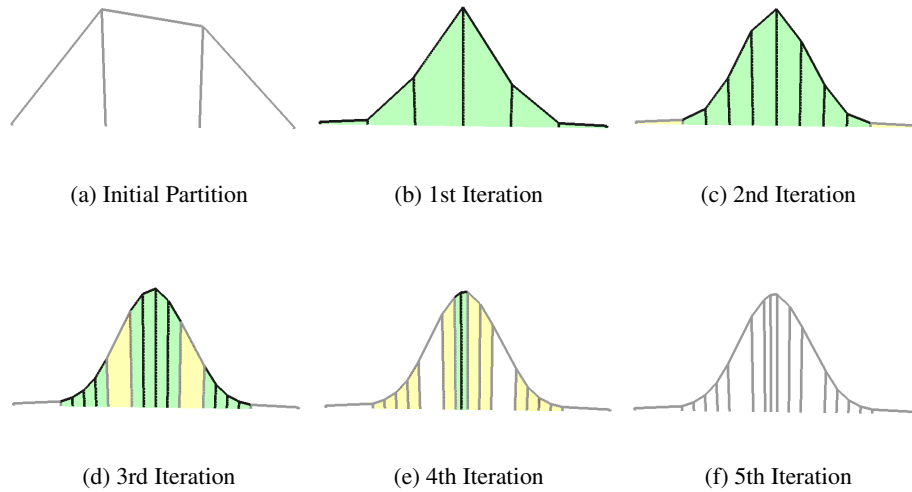
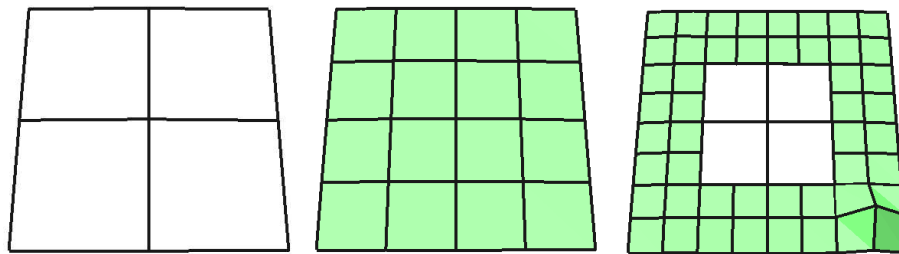


Figure 7.1: Visualization of 1D APDF Tree, Normal Distribution

7.2 Two-Dimensional APDF Tree

Figure 7.2 illustrates a two-dimensional APDF tree. The PDF of the dataset is constant in the middle part of the unit square, slightly decreases towards the outside of the unit square, and has a peak in the bottom-right corner (cf. Figure 7.2(e)).

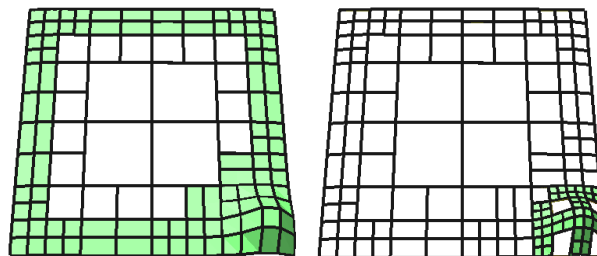
The visualization starts with a uniform partition and computes kernel additions on the intersection of partition lines (cf. Figure 7.2(a)). Then the APDF method estimates the areas of non linearity of PDF and splits the corresponding rectangles. All rectangles are split after the first iteration (cf. Figure 7.2(b)). This process is continued until the approximated shape error is uniformly low in all rectangles of the APDF tree. In iteration 2 (cf. Figure 7.2(c)) only the rectangles at the borders are split. These are the areas where the PDF is non-linear. Since the PDF is linear in the center of the space, no additional points are introduced there. The 3rd iteration adds more partition point at the borders of the dataset. The rectangles are split in both X and Y directions in the corners of the 2D universe. The PDF is non-linear in both directions there. The rest of the rectangles are split in one direction adapting to the non-linearity of the PDF. The shorter edge of the rectangle-leaf denotes direction of non-linearity of the PDF. The 4th iteration completes the creation of the APDF tree. There only the rectangles in the cylinder are of the universe are split. The PDF is non-linear at the extrema points of the PDF there: at the maximum



(a) Initial Step: Kernel Additions on UP

(b) 1st Iteration

(c) 2nd Iteration



(d) 3rd Iteration

(e) 4th Iteration and Final Partition

Figure 7.2: Visualization of a Two-Dimensional APDF Tree

density point of the structure and at the decrease areas of the peak.

7.3 A Complete Iteration of a Two-Dimensional APDF Tree

Figure 7.3 shows one complete iteration of the APDF tree. Figure 7.3(a) is the same as in Figure 7.2(c). Figure 7.3(e) is the same as Figure 7.2(d).

Six components completes the iteration of the APDF tree: *apriori split* (AS), *optimization of the tree* (TO/AS), *kernel additions* (KA), *posteriori split* (PS), and the second reorganization of the tree (TO/PS).

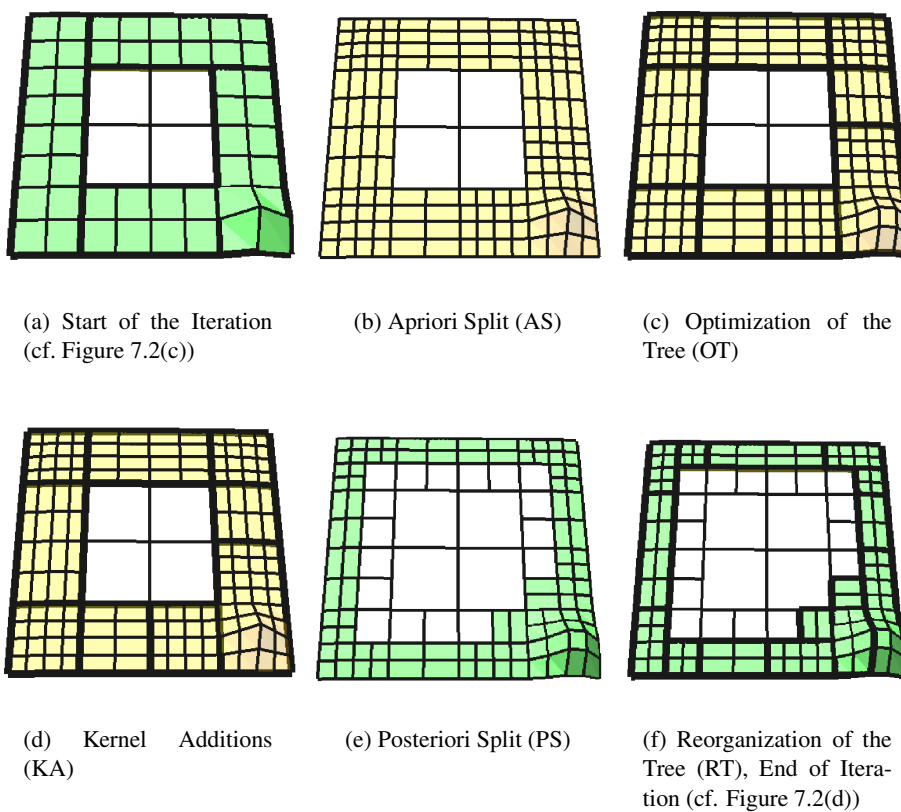


Figure 7.3: A Complete Iteration of the APDF Method

The iteration starts with an APDF tree and the set of nodes C that have been split in the previous iteration. The shape error can be too high in nodes C . In contrast,

if a node was not split in the previous iteration the shape error is already good and no more partition points needs to be added to the area.

Apriori split scans the nodes of nodes C and estimates the shape error (cf. Figure 7.3(b)). If the estimated shape error is too high in X or Y direction then the rectangle is split according to X , Y , or both directions. The apriori split can introduce too many splits, which are later removed by the posteriori split.

After new partition points have been added, an optimization of the tree component is applied to the tree to speed up the computation of kernel additions. The TO/AS algorithm scans the nodes C and groups similarly split nodes into new nodes of *local uniform partitions*. The algorithm produces rectangular shaped nodes. Figure 7.3(c) shows the reorganization of the nodes after the apriori split. This iteration reduces the memory usage up to 16% independent of the dimensionality.

Figure 7.3(d) illustrates the KA step. The KA algorithm scans the database, and identifies the set nodes I which are influenced by the data point. The partition points in I nodes are then updated with the kernel additions.

Posteriori split is illustrated in Figure 7.3(e). This step scans the nodes and removes the partition points, which did not increase the precision of the estimator. These points indicate that the shape error was already low in the area and the AS algorithm over-split the rectangle.

The second tree optimization step completes the iteration (cf. Figure 7.3(f)). Rectangles of the shape are grouped into the same node.

7.4 Three-Dimensional APDF Tree

In this Section we illustrate a three-dimensional APDF tree at the end of each iteration. We used the `art-plane-cylinder` dataset to illustrate the visualization of the tree.

The plane-cylinder dataset (cf. Figure 7.4(a)) consists of two structures in the three-dimensional space: a three-dimensional hyperspace (uniform distribution of points in the unit cube), and a cylinder (points distributed in a cylinder, in bottom-right of the XY plane). The density in the cylinder peaks in the center of the cylinder according to the XY directions. The density of the cylinder according to the Z dimension is constant.

Figures 7.4(b)– 7.4(f) illustrates the APDF tree at the end of each iteration. First iteration splits all cubes (cf. green cubes in Figure 7.4(b)). Then all cubes are organized into one node (cf. the cube with black border, Figure 7.4(b)). In the

second iteration all the outside cubes are split in all directions, though the inner cubes are split in one direction only (towards the outside). Although this is not visible directly, one can derive this information from the organization of the cubes into nodes (cf. cubes with black borders in Figure 7.4(c)). Figure 7.4(d) illustrates the APDF tree at the end of the 3rd iteration. The yellow cubes denote the AS splits that are removed by the PS split. The 4th iteration (cf. Figure 7.4(e) adds the last essential partition points into the APDF tree. The last iteration (cf. Figure 7.4(f) ensures that the shape error is low in all regions. The step adds a few partition points. Since all of them are removed by the PS split.

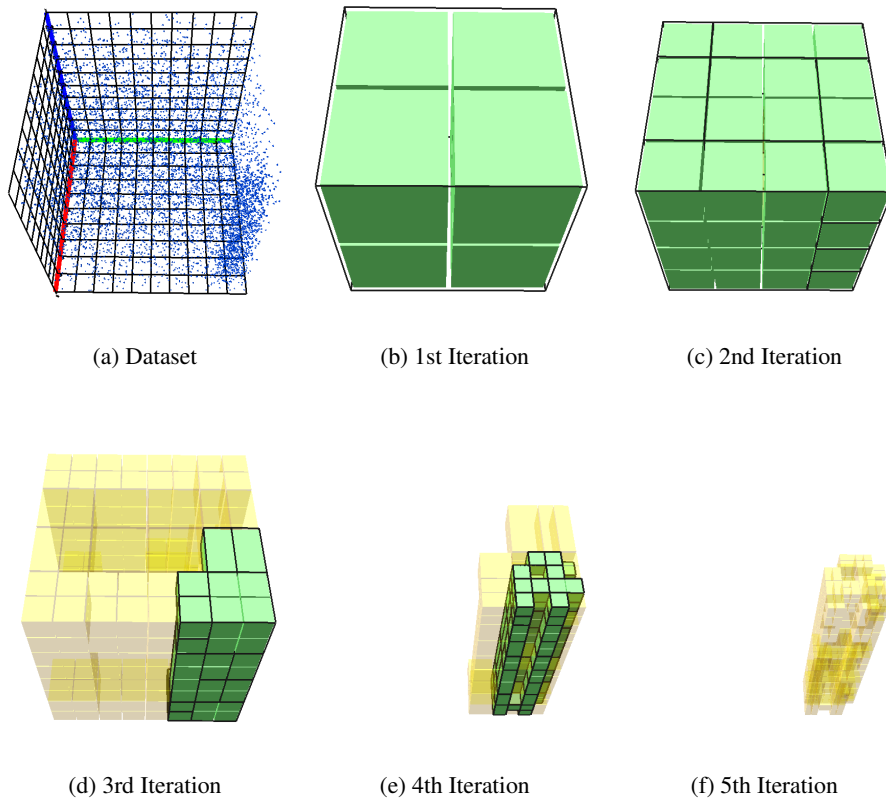


Figure 7.4: 3D APDF Tree

Chapter 8

Probability Density Functions and Density Surface Primer

In this chapter we first summarize probability density functions (PDFs) and the kernel estimation of PDFs (Section 8.1). A more detailed discussion of the kernel estimation method can be found elsewhere [4]. The spiral-sphere dataset is used to illustrate the kernel estimation. Next, we define density surfaces (DSes) and illustrate DSes for the sphere (Section 8.2). The definitions are given for the three-dimensional case. Figures are given for the one- and three-dimensional cases to better illustrate the concepts.

8.1 Kernel Estimation

The probability density function for an absolutely continuous random variable is defined as follows:

Definition 8.1.1 Let X be an absolutely continuous random variable with probability measure P . Function f with

$$\begin{aligned} F((x_1, x_2, x_3) \in A) \\ = \iiint_{(t_1, t_2, t_3) \in A} f(t_1, t_2, t_3) dt_1 dt_2 dt_3 \end{aligned}$$

is a *probability density function* (PDF) of random variable X .

Intuitively, by accumulating the density of dataset over region A we get the number of points that fall into A .

In general, we have to estimate the PDF because we are dealing with databases for which the PDF is unknown. We use the kernel method [5, 4] to estimate the PDF of the data.

Definition 8.1.2 The kernel estimate for a database containing observations $X^i, i = 1, \dots, n$, at point x is defined as follows:

$$\hat{f}_K(x_1, x_2, x_3) = \frac{1}{nh^3} \sum_{k=1}^n K\left(\frac{x_1 - X_1^i}{h}, \frac{x_2 - X_2^i}{h}, \frac{x_3 - X_3^i}{h}\right),$$

where K is a function (kernel) with $K \geq 0$, $\int K = 1$, and $K(x) = K(-x)$, $h > 0$ (smoothing parameter).

The essence of the kernel method is that each observation increases the chances of having another observation nearby. Therefore, we draw a symmetric kernel with an area equal to 1 (cf. dashed line, Figure 8.1(a)) around each observation. Adding all kernels yields an estimate of the PDF (cf. solid line, Figure 8.1(a)).

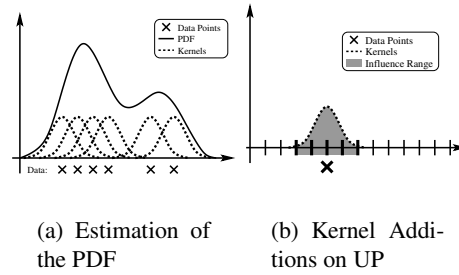


Figure 8.1: Kernel Additions, 1D Case

It has been shown [6] that the accuracy of the kernel estimation depends mostly on the smoothing parameter h and less on the choice of the kernel K . A too large smoothing parameter results in over-smoothing of the PDF (cf. Figure 8.2), whereas a too small smoothing parameter results in an under-smoothing of the PDF (cf. Figure 8.3).

Parzen [5] showed that the smoothing parameter

$$h = h_{opt} = c(K, \sigma_X, \sigma_Y, \sigma_Z) / n^{-1/7} \quad (8.1)$$

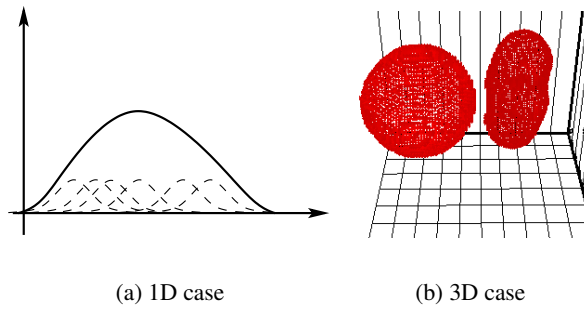


Figure 8.2: Over-smoothing: h is too large

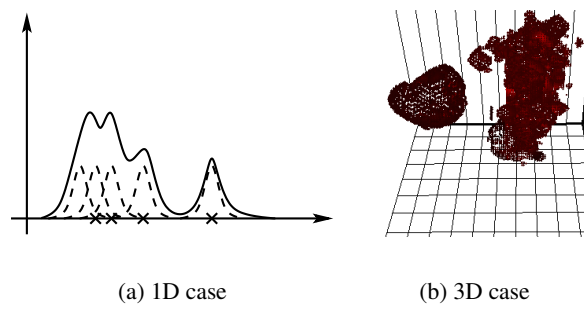


Figure 8.3: Under-smoothing: h is too small

minimizes the mean integrated square error (MISE):

$$\text{MISE} = \mathbf{E} \iiint (\hat{f}(x, y, z) - f(x, y, z))^2 dx dy dz. \quad (8.2)$$

In Equation (8.1) c is constant for a given dataset and depends on the variance σ_X^2 of the random variable X and the kernel function K .

In our VDM settings we chose the kernel to be triangular (the cheapest kernel that guarantees continuity with respect to the computational time) We chose the smoothing parameter that minimizes MISE. However, our approach is general enough, and other kernel function and smoothing parameter can be used.

Computation of the kernel additions for the uniform partition data structure (UP) is illustration in Figure 8.1(b). Note that the kernel function is not zero only in a small *contiguous* area of space. We scan the database and for each database point x we compute identify the (relatively small) interval of partition points I that is influenced by the kernel addition. We then update only the partition points in I . The time complexity of this computation is $O(n \cdot 2 \cdot h_{opt})$, where n is the size of the database, and $2 \cdot h_{opt}$ is the size of the interval I .

We can also computer kernel additions by first, scanning the partition points in the UP, and then for each *individual* partition point adding a Kernel function value. The time complexity of this step is $O(C \cdot n \cdot |P|)$. This is substantially larger than $O(n \cdot 2 \cdot h_{opt})$. Typically, one data point influences only about 0.1% of partition points for typical three-dimensional datasets we investigated in our explorations.

Kernel additions are efficient for uniform partitions of fixed, pre-determined size. This allows to effectively identify the interval of influence and the corresponding partition points. The APDF method deals with a dynamic data structure: the partition is neither fixed nor pre-determined in size. The method iteratively adds new partition points in the area of non-linearity of the PDF. The key challenge for the APDF tree is to organize the newly introduced partition points for the fast Kernel additions. If no organization is done the all newly introduced points will be processed *individually* resulting in $O(C \cdot n \cdot |P|)$ complexity. If an organization of the points into regions is done, one can decrease the time complexity and even be close to $O(n \cdot 2 \cdot h_{opt})$. The APDF method introduces such an organization of the newly introduced points. We group the points so the number of groups is minimal substantially increasing the computation of kernel additions.

8.2 Density Surfaces

Definition 8.2.1 The α density surface for the PDF $f(x)$ is the set of points:

$$DS(\alpha) = \partial \left\{ (x_1, x_2, x_3) : f(x_1, x_2, x_3) \geq \alpha \cdot \max_{x_1, x_2, x_3} f(x_1, x_2, x_3) \right\}, \quad (8.3)$$

where ∂A is the set of border points of A .

Intuitively, the α density surface envelopes all the data points, that are located at positions with density level above α .

Figure 8.4 illustrates a three-dimensional dataset and corresponding density surfaces. The data (cf. Figure 8.4(a)) is distributed according to a three-dimensional

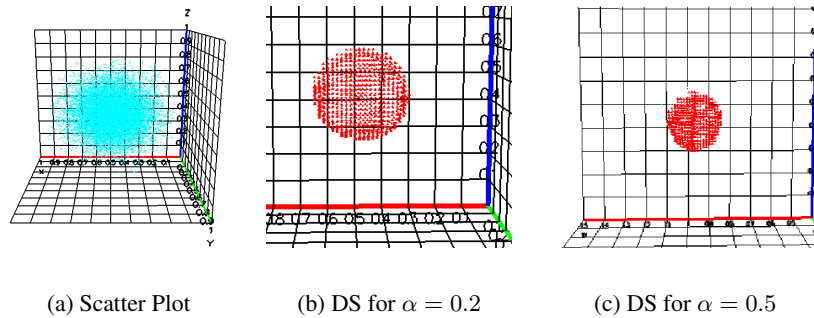


Figure 8.4: Density Surfaces for a 3D Normal Distribution

normal vector with the density peak in the center of the cluster. Figure 8.4(b) shows the density surface for density level $\alpha = 0.2$. The surface forms a sphere, which encloses all the data points at positions with density 0.2 and higher. By varying the density level α from 0.0 to 1.0 the density surface changes from the largest sphere (encloses all data points) to the smallest sphere (that enclose only the data points near the center (cf. Figure 8.4(c)).

Bibliography

- [1] *LAM / MPI Parallel Computing, MPI Tutorials: Getting started with LAM/MPI*, October 2001. <http://www.lam-mpi.org>.
- [2] H. R. Nagel. *Introduction to VR++*, *Getting Started Manual for version 0.5.1*, December 2002.
- [3] H. R. Nagel. *VR++ – Reference Manual*, *Reference Manual for version 0.5.1*, December 2002.
- [4] D. W. Scott. *Multivariate Density Estimation*. Wiley & Sons, New York, 1992.
- [5] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.
- [6] M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall, London, 1985.