A High-Dimensional Feature Clustering Approach to Support Search and Classification in Volumetric Data

A Thesis Presented

by

Mauricio Rafael Maurer

to

The Graduate School

in Partial Fullfilment of the

Requirements

for the Degree of

Master of Science

in

Computer Science

Stony Brook University

May 2008

Stony Brook University

The Graduate School

Mauricio Rafael Maurer

We, the thesis committee for the above candidate for the Master of Science degree, hereby recommend acceptance of this thesis.

Klaus Mueller - Thesis Advisor
Professor, Computer Science

Arie Kaufman - Chairperson of Defense
Professor, Computer Science

Jie Gao
Professor, Computer Science

This thesis is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

ABSTRACT OF THE THESIS

A High-Dimensional Feature Clustering Approach to Support Search and Classification in Volumetric Data

by

Mauricio Rafael Maurer

Master of Science

in

Computer Science

Stony Brook University
2008

Density and gradient values are not the only information present in volumetric data. Although most often used in current transfer function guided feature detection and enhancement, a wider variety of supplementary metrics can be extracted to reveal additional useful facts for classification of volume datasets. In this work we investigate the use of local density and gradient histograms in producing a high-dimensional feature-vector. The result can then be used for a more precise evaluation of the data and provides a foundation for feature mining in static and dynamic data. Nevertheless, feature vector characterization of local neighborhoods proved to be unmanageable under current transfer function paradigm due to its large size. We therefore propose to use cluster analysis for the task, to support users in the undertaking of defining feature models from high-dimensional data to be employed for object-level classification, detection and categorization.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The ultimate purpose of visualization is to exploit the analytical powers of the human brain to integrate low level features, made visible in the graphics rendering, into a full diagnostic evaluation. While the feature detection is a pre-attentive process, the subsequent integration is a conjunctive one that takes time and conscious attention. In fact, this bottom-up approach constitutes the fundamental underpinnings of *Integration Theory* [23] (as opposed to *Gestalt Theory*, which is top-down), and has also given the new field of Visual Analytics its great appeal. Discrete (sampled) data rendering has produced a great variety of techniques to make data features visually salient for easy detection by the human visual system. One measure of distinguishing these many techniques is by the underlying feature model, whose parameters are captured in a feature vector. Here, the simplest model is the one that is purely sample-based, that is, just the sample value (we shall assume scalar densities for this work, without loss of generality), which can be visually enhanced via suitable RGB mappings in transfer functions. This leaves it completely up to the user to infer structure in the visual integration process. The next step up is to assist the user in making the conjunctive feature integration. Shape is a strong cue, and to infer it, one can calculate gradients from the sample neighborhood. More advanced local computations can emphasize higher-level shape features, such as curvature, suggestive contours, and others. All of these features have in common that they have direct visual manifestations which can be readily brought out by their interaction with light. Thus, the underlying models have direct correspondences to familiar real-life expressive artifacts. Similar is true for vector and flow fields where streamlines or deforming textures can be used to visually aid in feature integration. But feature models can also reflect more abstract events, which can be deduced from the data. Popular here are physical as well as topological events in iso-contour and flow fields, which all can be visualized as a feature space of graph-like representations [18, 27]. Glyphs have also been devised to visually communicate the presence, structure, and location of such features to the user [26].

In all of the above cases, image analysis, guided by a suitable feature model, was used to perform some low-level feature integration for the user, so that these patterns could be perceived, visually encoded by appropriate graphical means. But there are many situations where feature models are not known (yet), or at least have not been formulated by a descriptive analytical process. In this case, one must resort back to raw patterns in the data and derive a model from scratch, using a learning approach (often called *Model Learning*). Learning requires a specification of features in the data that are characteristic for the sought model, able to distinguish positive from negative candidate instances. This specification is a key to finding an accurate model. Ideally, this process is fully automatic. We accomplish this by first deriving a set of rich feature vectors from the data and then using cluster analysis to learn the feature vectors relevant to the model. Here we will be looking for features that are less intuitive on first sight, and are thus not pre-attentive, requiring computer analysis for their derivation. Then, once these rich feature vectors have been learned, they can be used for subsequent classification, mining and selection. In that regard, unlike other works in volume graphics based on machine learning [24] our focus is not segmentation, but model-based object-level classification, detection, categorization and visual feature content communication.

This work is structured as follows. First, Chapter 2 will discuss relevant work. Then, Chapter 3 will present the low-level feature detection and description mechanisms we use, while Chapter 4 will describe the high-level learning framework to support classification and mining, and the visualization of the outcomes. Finally, Chapter 5 will present results of some case studies we conducted with our system, and Chapter 6 will end with conclusions and a brief discussion of future work.

CHAPTER 2

RELATED WORK

Voxel clustering has been a frequent mechanism for feature detection. It was popularized in the pioneering work of Kindlemann and Durkin [10] who identified region (object) boundaries by their characteristic arc-footprint in a 2D density-gradient magnitude scatter plot. However, once there are a sufficiently large number of boundaries, this scatter plot becomes excessively cluttered, making the detection of isolated arcs virtually impossible. For this reason Kniss et al. [12] devised an innovative dual-domain interaction interface, which enables users to explore local regions of the dataset in the volume domain and simultaneously observe the emergence of arcs corresponding to local isosurfaces within the clutter of the scatter plot domain. An iterative process conducted in this manner then identifies, labels, and paints all volume features. Although the emergence of GPUs enables such real-time interaction, this type of feature detection still requires a significant amount of user skill and time. Further, the approach also pointed to the limits of how many dimensions can be simultaneously managed within such an interactive interface.

Other clustering metrics have also been proposed since then, for example the LH-values [25], which encode the two volume values encountered by moving up and down the gradient direction from a given voxel. The LH metric is a very informative characterization of a boundary, as it reveals the regions which the boundary separates. The LH histogram, however, also operates within a scatter plot of two cluster variables, that is, L and H, which can result in clutter in the presence of many features. The incorporation of more than two metrics creates what is often referred to as *Voxel Signatures* [22]. Traditionally these have included densities, first and second derivative magnitudes, but also more advanced statistical measures, such as skew and kurtosis, as well as curvature [11], but one could easily add further metrics, such as the LH values discussed above. The visualization of these high dimensional points is challenging, and most typically an array of 2D projection scatter plots is used. This, however, limits a holistic assessment of the present patterns - the user must reconstruct the possibly complex high-dimensional shape mentally. Dimension reduction methods, such as SOM (Self Organizing Maps) [19], can be used to compress the high-dimensional space into 2D, however, this can be prone to errors and distortions, and it also requires a discretization into bins for display. An early but powerful direct high-dimensional exploration paradigm providing a combined scatter plot of an arbitrary number of variables is the Grand Tour [1]. The user is guided through hyper-space along a trajectory that is decided along the way by selecting the maximum of a given projection pursuit metric. The user stops when the current view point is the local maximum of this metric. In contrast, the method of parallel coordinates [9] maps a high-dimensional point into a piece-wise linear line that spans the vertical dimension axes. With Parallel Coordinates, clusters can be isolated by brushing the appropriate data axes. Another form of displaying high-dimensional data is the spectral plot, as has been used in [16], where various cues were added to help users perceive N-dimensional relationships. However, once the number of dimensions becomes even modestly large (dozens to hundreds) all of these paradigms fall victim to the curse of dimensionality [2], which makes the search for interesting patterns an unwieldy task.

To circumvent the scalability problems associated with visual clustering and pattern detection, we choose to instead use more descriptive, yet low-level, feature descriptors, which allow for more informed (pre-integrated feature) clustering in the computational domain. A subsequent visualization of the inter- and intra-cluster relationships, along with iconic representation of the underlying patterns, then allows users to get a feel about the semantics of the data.

One of the feature descriptors we employ are local density histograms, which have found frequent use in volume rendering before. For example, [15] utilize histograms of local neighborhoods to capture tissue characteristics for locally-sensitive transfer function specification, while [20] employ local histograms to simulate color bleeding and ambient occlusion effects. Apart from density, we also characterize local features by their gradient field statistics, in form of gradient histograms. For this, we use the SIFT (Scale Invariant Feature Transform) operator [13, 14], which has become quite popular in the computer vision literature and content-based image retrieval, but to the best of our knowledge has rarely been used in the field of visualization and volume graphics.

CHAPTER 3

LOW-LEVEL FEATURE DETECTION AND DESCRIPTION

As mentioned, we use two types of statistical low-level feature descriptors: density histograms to represent the overall texture statistics and gradient histograms to capture the perceptual effects of these. In particular the latter is targeted to support the feature integrative systems of human vision. Both will be insensitive to affine transformations, such as scale, rotation and translation, to support feature and object mining in large datasets. We describe both of these feature descriptors in the following two sections.

3.1 Density histograms

A global histogram is not descriptive enough as a classifier, and this motivated the use of more feature-oriented histograms. We did not attempt automated segmentation which is still an open research area, to confine the extent of histograms within object boundaries. Instead, we aimed for a representation that has good potential to capture the essence of an object, and yet is automatic to compute. This led to our approach of computing the density signatures in local histograms at a hierarchy of window sizes, to enable the detection of feature density statistics at multiple levels of scales. Rotation invariance is achieved by constructing the histograms within slightly overlapping radial regions. We also experimented with space-filing tilings: hexagons in 2D, rhombic dodecahedrons in 3D, and so on. For reasons of simplicity in the indexing [28], we eventually chose the radial primitives in our work.

To determine local histograms locations, an algorithm based on the sphere packing problem was performed prior to the extraction of the histograms. Overlapping regions are achieved by extending the radius of each sphere by a percentage decided upon the smallest possible radius. In our experiments, radiuses are not smaller than 4 voxels what makes this percentage be 12.5% so that the extended radius can be rounded at least one discrete unit of the sampled data or, in other words, 1 voxel. This extension is useful for two reasons: rotation invariance, as described before, and to reduce unvisited voxels due to the lattice structure that, according to the Kepler Conjecture, achieves the highest density among arrangements of spheres and reaches approximately 74%[1] of the volume. The maximum region size is determined by the smallest dimension of the volume, being at most $1/4$ of this value. Bigger patches lead to less specific descriptors and are avoided in this work. The pseudo-code algorithm presented in Figure 3.1 computes sphere centers taking into consideration the aforementioned constraints. It is worth to know that the distance between sphere centers in the x plane is one diameter, the distance between spheres in the y direction is $\sqrt{3}\cdot$diameter$/2$ whereas the distance between sphere centers in the z direction is $\sqrt{6}\cdot$ diameter$/3$. Given an initial radius $iRadius$ and the $width$, $height$ and $depth$ of the volume, the algorithm generates a list of sphere centers, that can then be used to extract local histograms centered on those point. The algorithm makes use of three auxiliary functions: `min`, `checkBounds` and `store`. Function `min` which computes the smaller of two values, `checkBounds` check a coordinate (x, y, z) againts the volume bounds and `store` saves all three coordinates, along with the patch radius, in an abstract data type T. Each successive level of the hierarchy possesses radiuses as twice as big as the previous level.

3.2 Gradient Histograms

To encode the gradient signatures we used the SIFT feature descriptor introduced in Chapter 2. The SIFT feature descriptor is based on a model of the behavior of cerebral cortex cells in primate vision, which are sensitive to intensity gradients at different levels of scale. Founded in these evolutionary principles, it has been shown to outperform other local descriptors on both textured and structured scenes. The SIFT algorithm consists of two phases: (i) the detection of critical points (the keypoints) in scale-space and (ii) the encoding of these into keypoint descriptors. The keypoints are local extrema in a difference-of-Gaussians multi-scale pyramid. They are detected by comparing a point's neighbors on the current scale and the scales above and below. A detailed fit to the nearby data is then performed for location, scale and ratio of principal curvature for each keypoint. Points having low contrast or poorly localized along an edge are then rejected limiting the set of keypoints to the most descriptive and salient ones, preserving manageability. The keypoint descriptor is an orientation histogram of local gradients, quantized into bins. It is constructed by computing the magnitude and orientation

[1] The exact number is $\frac{\pi}{3\sqrt{2}} \cong 0.74048$.

```
input : An initial radius $iRadius$ and the volume dimensions $width$, $height$ and $depth$
output: A list of packed sphere centers
$leastDimension \leftarrow$ min $(width$, min $(height, depth))$
$diameter \leftarrow (iRadius \times 2)$
repeat
    $r \leftarrow \frac{diameter}{2}$
    /* Maximum sphere diameter is one fourth of the smaller
       dimension                                            */
    for $i \leftarrow 0$ to $\frac{width}{diameter}$ do
        for $j \leftarrow 0$ to $(j \times \sqrt{3}r) \leq height$ do
            for $k \leftarrow 0$ to $(k \times \frac{2\sqrt{6}r}{3}) \leq depth$ do
                $y \leftarrow \sqrt{3} \times r \times j$
                /* If depth plane is odd, spheres are shifted in the
                   x-direction by r and in the y-direction by -r   */
                if $(k \% 2) = 0$ then
                    $x \leftarrow i \times 2r \times (j \% 2)$
                else
                    $x \leftarrow i \times 2r - r \times (j \% 2) + r$
                    $y \leftarrow y - r$
                end
                $z \leftarrow \frac{2\sqrt{6}r}{3} \times k$
                if $\neg$checkBounds($x$,$y$,$z$,$width$,$height$,$depth$) then continue
                store($T$,$x$,$y$,$z$,$r$)
            end
        end
    end
    $diameter \leftarrow 2 \times diameter$
until $diameter > \frac{leastDimension}{4}$
```

Figure 3.1: Sphere Packing algorithm - Computation of spheres centers.

at each image sample point in a region around the keypoint location, weighted by a Gaussian function with σ equal to one half the width of the descriptor window to achieve a certain level of smoothing. The samples are then aggregated into 8 bin orientation histograms describing the neighborhood over a 4×4 matrix of subregions. Figure 3.2 shows a 2D example where pre-computed gradients (background) are accumulated into 8-bin orientation histograms (foreground) for each subregion. This figure shows an 8×8 neighborhood with 2×2 subregions while SIFT uses 16×16 neighborhoods with 4×4 subregions. This histogram is stored into a 128-long feature vector, along with the scale identification and location. Further, the SIFT feature descriptor can be made orientation independent by aligning the descriptor and the gradient orientations by the keypoint orientation. Thus, the SIFT feature descriptor is highly expressive (and integrative) of a neighborhood's salient dynamics, and it is largely insensitive to affine transformations, such as scale, translation, and rotation and at least partially invariant to illumination.

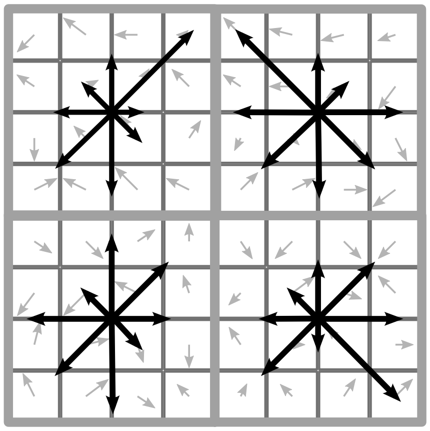
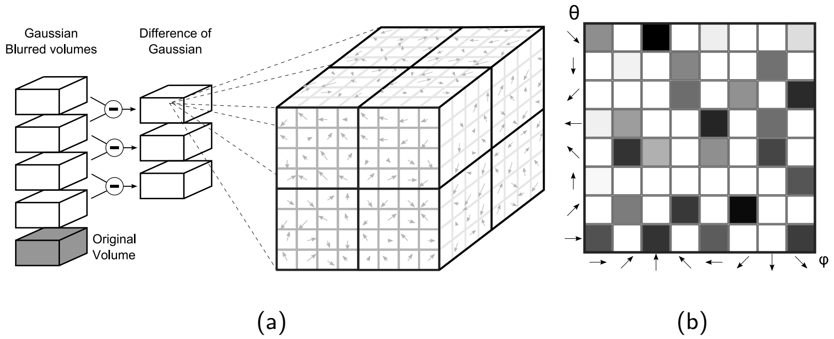


Figure 3.2: SIFT keypoint neighborhood.

SIFT has traditionally been used only in 2D in the context of images, but recently [3] and [21] have generalized the framework into higher dimensions, for the purpose of determining accurate feature point correspondences between medical volume datasets for object alignment and for action recognition in video data, respectively. Although both works extended SIFT to be used with 3-dimensional data, the former employs a more sophisticated keypoint localization approach while the latter relies on a random procedure for the extraction of interesting points. This methodology allows faster runtimes but fails to capture the essence of the SIFT algorithm, which is the localization of points that contain striking characteristics of an image - reproducibility is one of the main characteristics responsible for image matching. Thus, we use a similar method to extract keypoints and compute their feature descriptor to that of [3] in our application.

First a difference of Gaussian (DoG) pyramid of datasets is built. A series of Gaussian blurred volumes is computed from the first dataset with sigma $= \sigma, k\sigma, k^2\sigma, k^3\sigma, \ldots, k^n\sigma$ forming an octave. The first volume of the first octave is the original dataset and the first volume for the subsequent octaves is a half-sized version of the $(n-1)$th blurred volume of the previous level. The DoG datasets are then computed from its neighboring blurred datasets at scales k^j and k^{j+1}, generating the total of s DoG datasets. Finally, each DoG volume is searched for local maxima or minima. Similar to the SIFT algorithm, every voxel is compared against all its neighboring voxels in the current scale and in the DoG datasets in the scale above and below. In 3-dimensions this step requires checking 80 voxels whereas in SIFT only 8 pixels are compared to the current pixel. Finally, only voxels having magnitude greater than a threshold T_{DoG} are considered feature points. Because searching for local extrema needs at least 3 volumes, one at the current scale j, one at scale $j+1$ and one at scale $j-1$, the number of DoG datasets has to be at least $s+2$ and the number of blurred datasets needs to be $s+3$.



(a) (b)

Figure 3.3: Summary of 3D-SIFT algorithm.

The aforementioned 3D-SIFT feature is comprised of a 4096 cubic voxel region around the keypoint (16^3 voxels), having 64 subregions and each summarized by an 8×8 histogram of voxel orientations, as depicted in Figure 3.3. In Figure 3.3 (a) one octave of the pyramid is shown. Scale increases along the up direction. It also depicts subregions adjacent to a keypoint in a cube of 8^3 voxels centered in the keypoint. The figure shows only 2×2 subregions while the algorithm employs 4×4 regions in a neighborhood of 16^3 voxels. The 2-dimensional histogram of gradient orientations of one subregion is shown in Figure 3.3 (c). Black squares correspond to 1 and white squares to 0. The histogram is normalized to $[0, 1]$. The natural generalization of SIFT gradient direction representation to 3-dimensional datasets uses 2 spherical coordinates (θ, φ) (r, the first of the three spherical coordinates, is equal to the gradient magnitue) and gives rise to 2-dimensional orientation histograms with b bins for each coordinate, all ranging from 0 to 2π.The result is a histogram of size b^2, each coordinate being discretized into b bins of $2\pi/b$ or $(360/b)^\circ$. Since the algorithm presented by [3] is an extension from SIFT to N-SIFT (n-dimensional SIFT), the calculation of θ and π is implemented following a dynamic programming approach. Spherical coordinates can then be transformed from cartesian coordinates with the three following equations:

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \arctan\left(\frac{x}{y}\right) \\ \varphi &= \arctan\left(\frac{y}{x \ \cos\theta}\right) \end{aligned}$$

The r coordinate is the gradient magnitude and is pre-computed during the octave pyramid construction, being θ and φ the only two coordinates that describe the orientation of the voxel at a given position. In the case of 3D SIFT, each cartesian coordinate is given by the central difference method, calculated with the following equations:

$$\begin{aligned} x &= f(x+1, y, z) - f(x-1, y, z) \\ y &= f(x, y+1, z) - f(x, y-1, z) \\ z &= f(x, y, z+1) - f(x, y, z-1) \end{aligned}$$

where $f(x, y, z)$ gives the scalar value associated with coordinates (x, y, z). Every gradient magnitude, r, is weighted by a Gaussian function centered at the keypoint before being accumulated into the corresponding bin of the subregion histogram. The concatenation of such histograms form the 3D-SIFT feature descriptor.

CHAPTER 4

HIGH-LEVEL LEARNING AND CLASSIFICATION, AND SIGNATURE VISUALIZATION

Having captured the essential perceptual features of a sampled object or phenomenon by the two feature descriptors described in Chapter 3, we are now ready to assemble these into high-level constructs. Our goal is to learn a high-level representation of the sampled object, using its low-level features. Here, we distinguish between three levels. Level-1: categories (for example, smoke vs. fire), level-2: category instances (for example, turbulent smoke vs. laminar smoke), and level-3: groups of similar individual features. Note that the top two levels may overlap in terms of the third level, that is, third-level features may be part of separate categories and instances thereof. We use K-means clustering [17] at all levels to form a 3-level hierarchy.

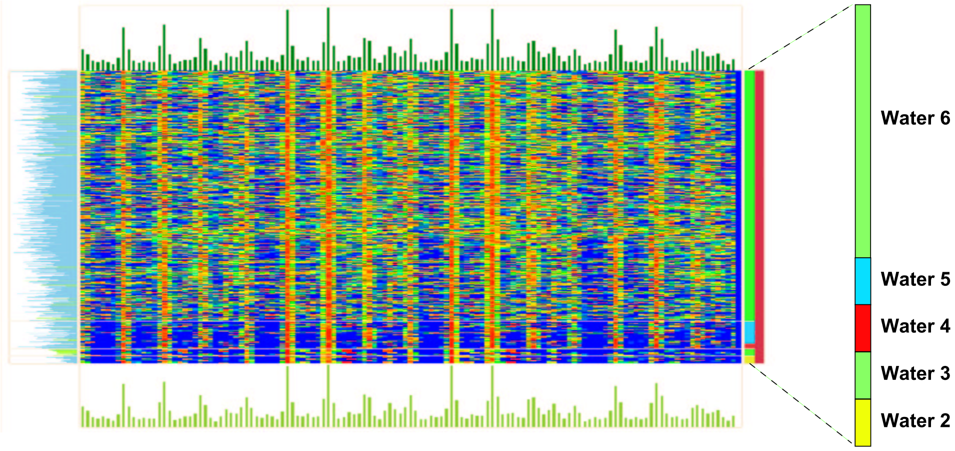
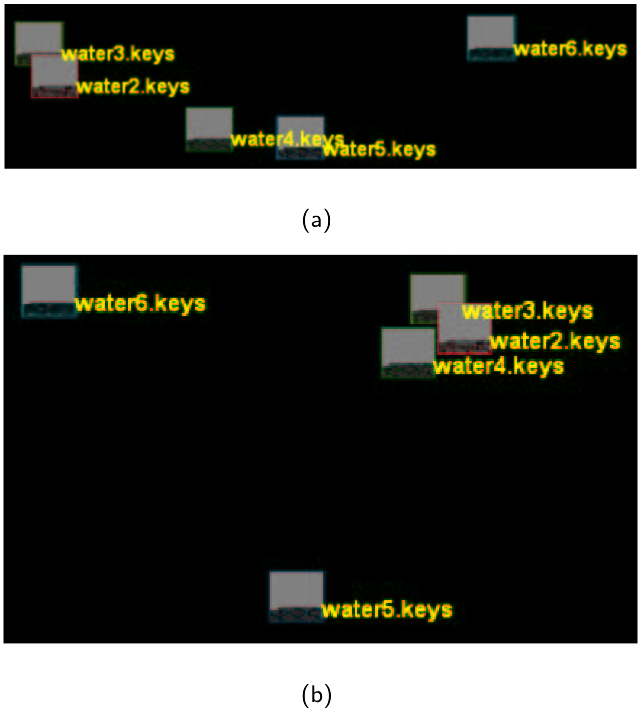


Figure 4.1: Display of SIFT feature descriptors of five water flows.

We generated animated sequences of water flows to experiment with our framework. Figure 4.3 presents all five flows used in the tests: (a) flow 2; (b) flow 3; (c) flow 4; (d) flow 5 and (e) flow 6. We computed SIFT feature descriptors of five different water flows, and clustered them with K-means. First, SIFT keypoints were extracted from each frame of the flow. Then, all extracted keypoints were concatenated in one big set that contained the SIFT keypoints for an entire flow sequence. Figure 4.1 shows a spectral plot of the separated clusters. Each row is one feature vector which value is represented by colors (from blue to red, the value increases). The complexity of this spectral plot prohibits one to discern cluster relationships in an intuitive fashion. We therefore added further feature integrative support by visualizing the clusters (at all three levels) via Multidimensional Scaling (MDS). For this, we computed the distances between all cluster pairs using various distance metrics and use MDS to flatten the N-D space into 2D space. The closer the distance between elements in the MDS plot, the more similar the clusters are. We have tried two similarity metrics to calculate distances for an MDS map: (i) simple Euclidean distance between cluster centers and (ii) Clustering Using Representative (CURE) similarity [7].

In CURE, a constant number c of *well scattered* points is randomly chosen, which captures the shape and extent of the cluster. These points are then multiplied by a fraction α, shrinking the set around the centroid of the cluster. This step prevents CURE from the undesirable effects caused by outliers. Picking several points, instead of a small number of them, permits the CURE algorithm to learn non-spherical clusters. Adjusting the parameter α in the range $[0, 1]$, it is possible to radically change the behavior of the algorithm and modify the shape of the learnt clusters. After comparing the distance between every two pairs of two clusters, the smaller distance is then chosen as the representative distance among two cluster. In our experiments, we use $c = 12$ and $\alpha = 0.4$.

Figure 4.2 shows an MDS plot generated by the (a) Euclidian of centers distance and (b) CURE similarity metrics (of the flows of Figure 4.3). Here, flows 2 and 3 and flows 4 and 5 are very similar, while flow 6 is extremely different from all others. However, as mentioned, if the cluster shape is not uniformly distributed from the cluster center, this metric may not capture the similarity correctly. In Item (b) of Figure 4.2, using the CURE similarity, flow 4 is gathered with flows 2 and 3 while flow 5 is separated from the group. This shows that the flow 5 may have a similar cluster center than flows 2, 3, and 4 but it does not have a similar shape within the group. With the CURE metric, flow 6 is still preserved as a separated pattern. Furthermore, to give the user more insight on the cluster's composition we represent each MDS node as an icon that pictures a matrix of the dimension values. From the bottom left to the top right corner of each

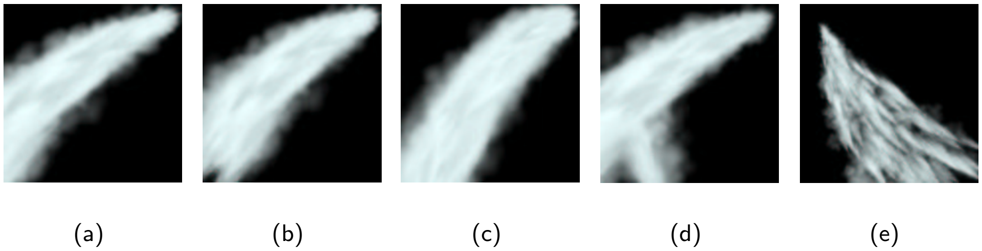


(a)

(b)

Figure 4.2: MDS maps of the Water Family of flows.

node, the matrix is divided into the number of dimensions. Every cell shows the average value of each dimension ny means of brightness. This small icon provides users with more information on the activities of all dimensions.



(a) (b) (c) (d) (e)

Figure 4.3: Single frames of different water jet flows.

The user is now free to drill into the MDS plot to view the cluster decomposition and explore the intra- and inter-cluster relationships. This exposes the level-3 features in relation to the level-2 instances (still using only one level-1 category: water flow). We run K-means clustering on each flow cluster using $K = 5$ and 10 iterations. Figure 4.4 shows the entire MDS map (for example, water3.keys-2 denotes the 2^{nd} cluster of the flow 3). Important groups are circled and numbered in white letters. We observe that small clusters from different flows overlap at several locations. This demonstrates in an intuitive way that there are some common features in these water flows.

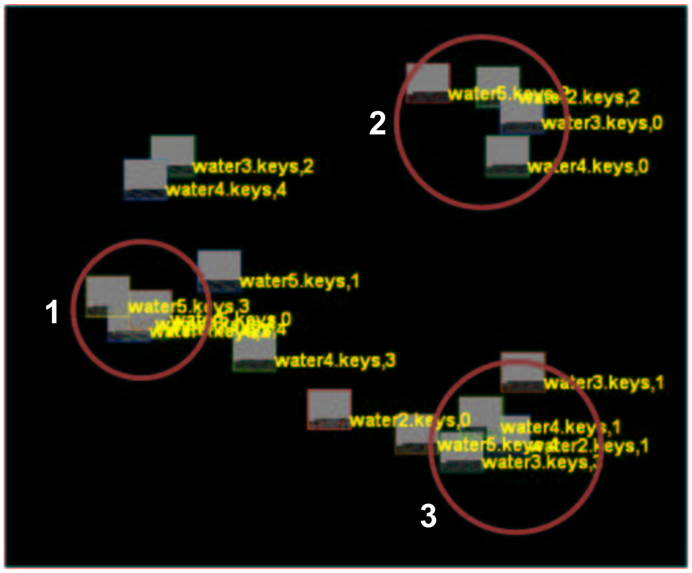


Figure 4.4: Separating out individual (flow) features (level-3 clusters) in the level-2 clusters using K-means.

The spectral plot in Figure 4.5 expands the inner words contained in circle 1 of Figure 4.4. It is possible to recognize that water2-4, water3-4, water4-2, and water5-0 are very similar to each other. Histograms on top of each word represent the overall sum of each column in the word drawn below. This visual tool improves the ease of atesting the similarity among words. Using these learnt relationships, we can now easily categorize new water flow datasets at all 3 levels.

In Figure 4.6, the third frame of each of the flows presented in Figure 4.5 are showed with red dots marking the keypoints clustered into circle number one of Figure 4.4. Clusters water2-4, water3-4, water4-2 and water5-0 correspond to items (a), (b), (c) and (d) of Figure 4.6. Although several keypoints exist in each cluster, only those that belong to frame number 3 are shown, since other keypoints were found in different frames. It is observable that these visual words fall into similar spatial locations, in between the flow

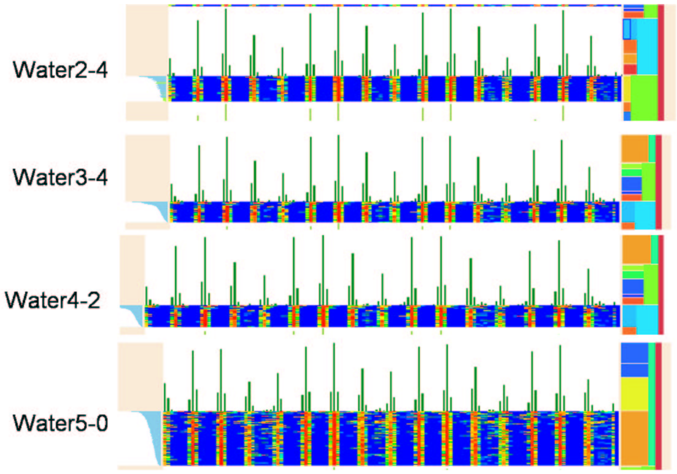


Figure 4.5: Some features displayed in the spectral plot.

and the background. This evidence paves the way for the more concrete use of the low- and high-level descriptors discussed so far. The application of those descriptor will be explored in more detail in the next chapter, in which we also briefly discuss the results.

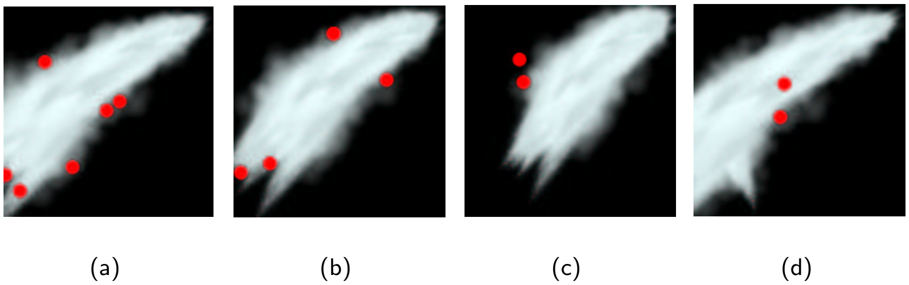

(a)
(b)
(c)
(d)

Figure 4.6: Third frame of flows 2, 3, 4 and 5 with dots marking locations of keypoints.

CHAPTER 5

RESULTS

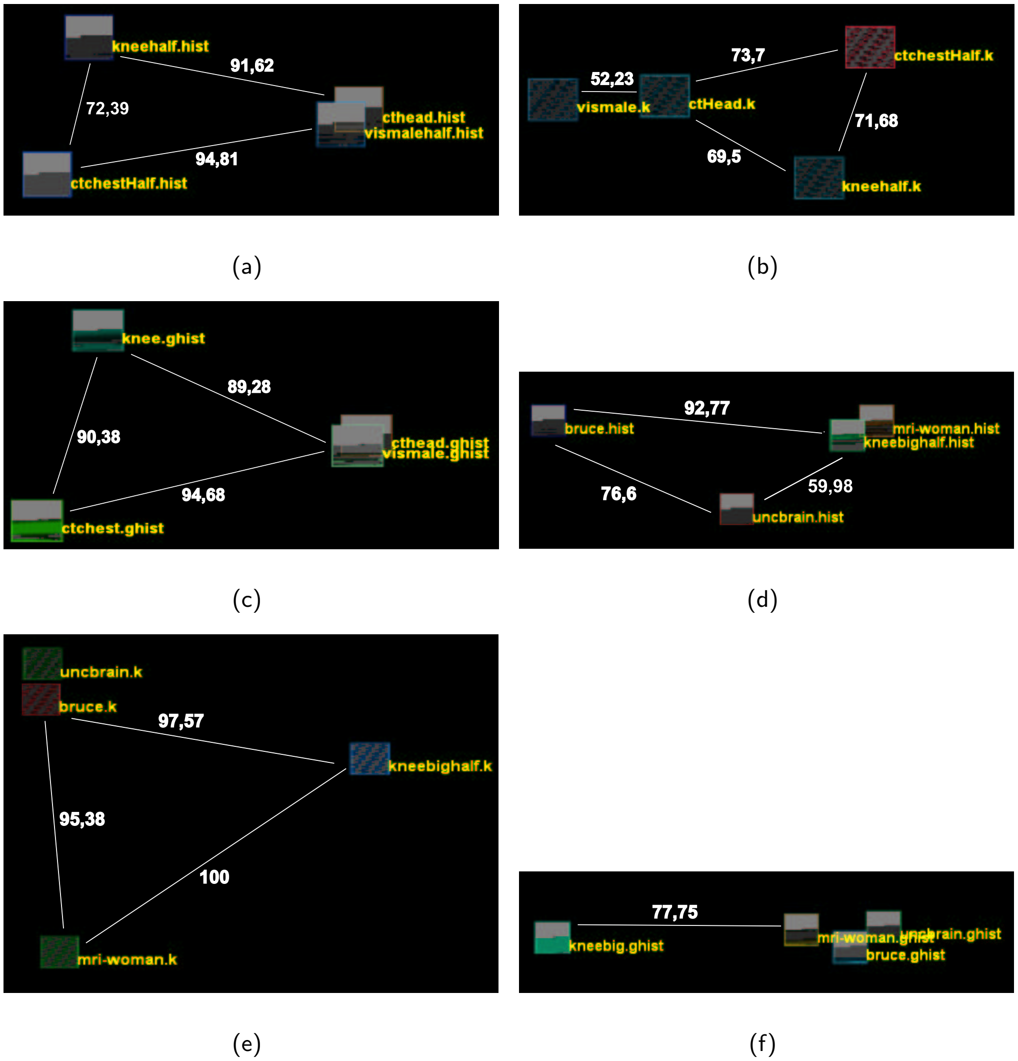
We applied the previously described low- and high-level techniques in conjunction with different goals. Next, we first explain the process used to cluster different body parts in MRI and CT datasets and 3-dimensional flows and then we describe the difference in the results obtained by applying SIFT and 3D SIFT to 2-dimensional flow data in a series of frames or assembled as a volume, respectively.

5.1 Categorization of MRI and CT volumes

Four different medical datasets for each modality (CT and MRI) had their global and local density histograms computed, along with the extraction of 3D SIFT feature descriptors. Table 5.1 summarizes all datasets and their characteristics. For the calculation of the local density histograms, the rhombic dodecahedrons were inscribed in a sphere with radius equal to 8 voxels extended 12.5% to allow for the overlapping of adjacent spheres. At each successive level, the radius of the sphere was doubled, eventually reaching one fourth of the smallest dimension of the volume. The extraction of 3D SIFT feature descriptors was executed with $T_{DoG} = 0.0075$ (considering the voxels were normalized in the range [0,1]), $\sigma = 4.0$, 3 DoG datasets per octave and 8 bins for the gradient orientation histogram (for each spherical coordinate).

Dataset name	Body part	Modality	Dimensions
kneehalf	Knee	CT	$190 \times 115 \times 153$
ctchestHalf	Chest	CT	$192 \times 192 \times 120$
ctHead	Head	CT	$128 \times 128 \times 112$
vismale	Head	CT	$128 \times 256 \times 256$
bruce	Head	MRI	$256 \times 256 \times 156$
uncbrain	Head	MRI	$128 \times 128 \times 72$
mri-woman	Head	MRI	$256 \times 256 \times 109$
kneebighalf	Knee	MRI	$256 \times 256 \times 87$

Table 5.1: Main characteristics of the CT and MRI datasets used in our experiments.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.1: MDS plot of different datasets of different body parts, divided by modality.

Figure 5.1 shows three different MDS plots for each modality and presents the clustering of all different body parts within their respective modality. White numbers are actually the original Euclidean distances between clusters. White lines connect clusters whom distances are shown. Figure 5.1 is divided as follows: (a) CT, local histograms, Euclidean distance of centers; (b) CT, 3D SIFT, Euclidean distance of centers; (c) CT, Global histogram, Euclidean distance of centers and CURE give the same plot; (d)

MRI, local histograms, CURE; (e) MRI, 3D SIFT, CURE; (f) MRI, global histogram, Euclidean distance of centers and CURE give the same plot. We observe that the body parts originated from CT scans are well categorized by the MDS plot using global and local histograms. The result obtained by applying 3D SIFT was also good, but might create an ambiguity in an automated categorization of heads if compared to other body parts. The result obtained with MRI datasets was not as accurate, especially when local histograms were used as low-level descriptors. All head datasets were kept distant from each other and the knee was clustered with the mri-woman dataset. 3D SIFT provided a slightly better descriptor, even though the mri-woman dataset was clustered far from the other head volumes. The lack of consistent medical data prevents us from drawing definitive conclusions, but from this small set of volumes it is possible to see that some correlation (positive) between similar body parts exists and more sophisticated clustering techniques may achieve better results.

5.2 Categorization of flow data

The analysis of 2-dimensional flow data was executed with four different families (synonymous to categories) of flows: Water Jet, Smoke, Fire and Flame. Each family has several members, each one different from the others with respect to scale, orientation and other characteristics such as speed, randomness, weight, strength, etc. All series were generated with the program Wondertouch Particle Illusion 3.0 [29]. We used this program since it allowed us to generate a highly diverse set of amorphous phenomena, which we recorded as a sequence of images, one for each frame. SIFT was then applied to each frame separately and all keypoint descriptors were concatenated forming one single set of feature descriptors for the entire sequence, as described before. Even though SIFT is essentially a 2-dimensional algorithm per se, the third coordinate of each point, z, is extracted from the frame where the points is located[1]. In addition, 3D SIFT was also performed on the volume assembled by stacking all frames of the flow in order. In such cases, the depth of the volume reflects the number of frames, while the width and height correspond to the width and height of the frames of the sequence (since frame dimensions don't change throughout the sequence, it is correct to assume that the width and height is equal to any frame of the flow). All parameters of the 3D SIFT algorithm were the same as those used with the CT and MRI datasets of the previous section.

[1] In other words, if a keypoint is found in frame 7 with coordinates $(23, 56)$, its 3-dimensional coordinate will be $(23, 56, 7)$.

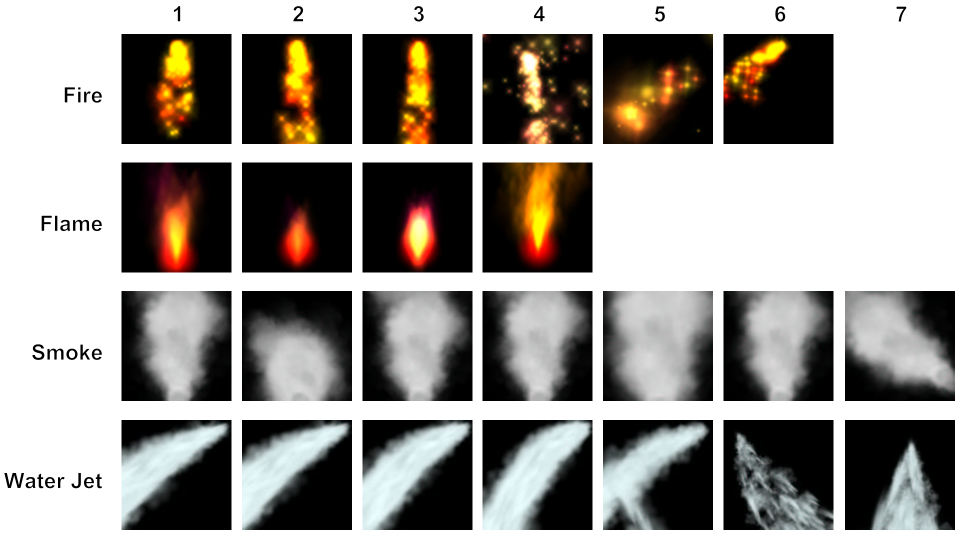
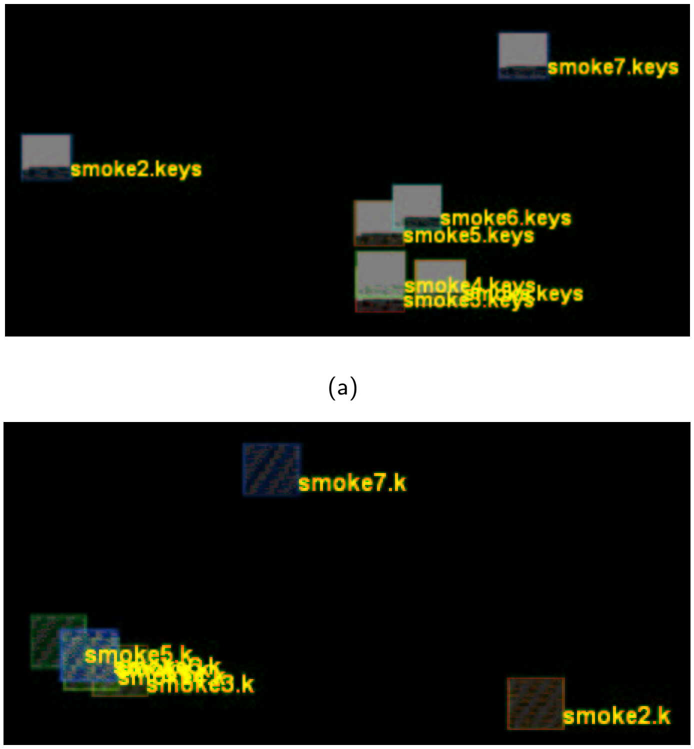


Figure 5.2: Flows used in our experiments.

Dataset name	Family	Dimensions		Dataset name	Family	Dimensions
Water 1	Water Jet	$160 \times 160 \times 120$		Smoke	Smoke	$160 \times 160 \times 120$
Water 2	Water Jet	$160 \times 160 \times 120$		Smoke 2	Smoke	$160 \times 160 \times 120$
Water 3	Water Jet	$160 \times 160 \times 120$		Smoke 3	Smoke	$160 \times 160 \times 120$
Water 4	Water Jet	$160 \times 160 \times 120$		Smoke 4	Smoke	$160 \times 160 \times 120$
Water 5	Water Jet	$160 \times 160 \times 120$		Smoke 5	Smoke	$160 \times 160 \times 120$
Water 6	Water Jet	$180 \times 180 \times 120$		Smoke 6	Smoke	$120 \times 120 \times 120$
Water 7	Water Jet	$180 \times 180 \times 180$		Smoke 7	Smoke	$120 \times 120 \times 120$
Fire 1	Fire	$160 \times 160 \times 120$		Flame 1	Flame	$160 \times 160 \times 120$
Fire 2	Fire	$160 \times 160 \times 120$		Flame 2	Flame	$160 \times 160 \times 120$
Fire 3	Fire	$160 \times 160 \times 120$		Flame 3	Flame	$160 \times 160 \times 120$
Fire 4	Fire	$160 \times 160 \times 120$		Flame 4	Flame	$160 \times 160 \times 120$
Fire 5	Fire	$160 \times 160 \times 120$				
Fire 6	Fire	$160 \times 160 \times 120$				

Table 5.2: Main characteristics of the flow datasets used in our experiments.

Figure 5.2 depicts all flows used in our experiments. One frame of each flow is presented. Although almost imperceptible, the differences between flows 1, 2 and 3 of the Fire family rely on the speed and angle of emission of the particles source. The same applies to flows 1, 3, 4, 5 and 6 (flow 5 is a closer view of flow 1 and flow 6 is a downsized version of the same flow) of the Smoke family and flows 1, 2, 3 and 4 of the Water Jet family. Table 5.2 is a summary of characteristics of flows. As described before, the $depth$ component of the dimension is the number of frames of each flow.

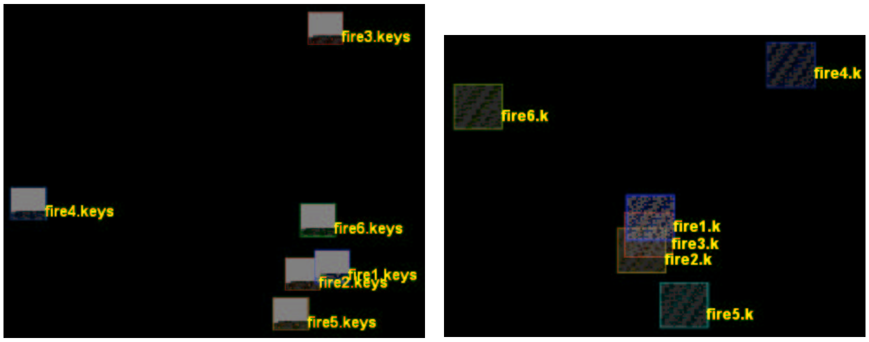


(a)

(b)

Figure 5.3: MDS plot of the Smoke Family using the Euclidean of Centers similarity metric.

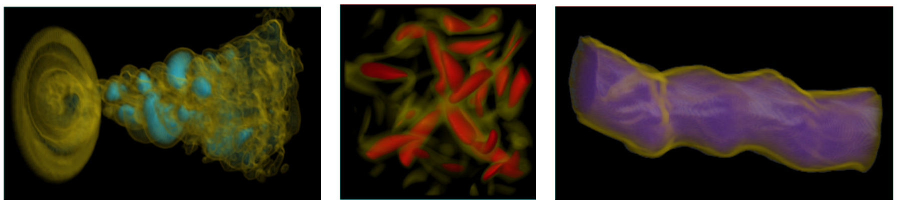
Applying SIFT to each flow and visualizing the MDS map of each family, it is possible to notice that similar flows fall very close to each other, while more different elements (even if subtly) are placed farther from the rest. As the difference increases from one dataset to the other, the distance in the MDS plot also increases. This effect can be clearly seen in Figure 5.3 in an MDS plot of the Smoke family of flows. Item (a) was generated using SIFT as keypoint descriptor while item (b) was produced employing 3D SIFT as low-level descriptors. It is interesting to note that 3D SIFT reduces the distance among flows 3, 4, 5 and 6. Flows 2 and 7 are correctly placed far apart from the other flows, meaning they are fairly different from the group. 3D SIFT succeeds in detecting the subtle similarity in more detail among flows 3, 4, 5 and 6 and places the flows closer to each other if compared to the result produced by using SIFT. Figure 5.4 presents two



(a) (b)

Figure 5.4: MDS plot of the Fire Family using the CURE similarity metric.

MDS plots from the Fire family generated with the CURE similarity metric based on SIFT and 3D SIFT descriptors in items (a) and (b), respectively. It is noticeable that SIFT fails to categorize flow 3 correctly. Flow 5 is similarly positioned on both diagrams and flow number 6 is depicted with more fidelity if SIFT descriptors are used. This result can easily be explained by the fact that flow number 6 is a rotated version of flow 1 and, because SIFT is invariant to rotation, it captures very well the similarity among flows 1, 2 and 6. 3D SIFT, although somewhat invariant to rotation, has difficulties in matching volumes rotated by more than 10°, as was also observed by [3].



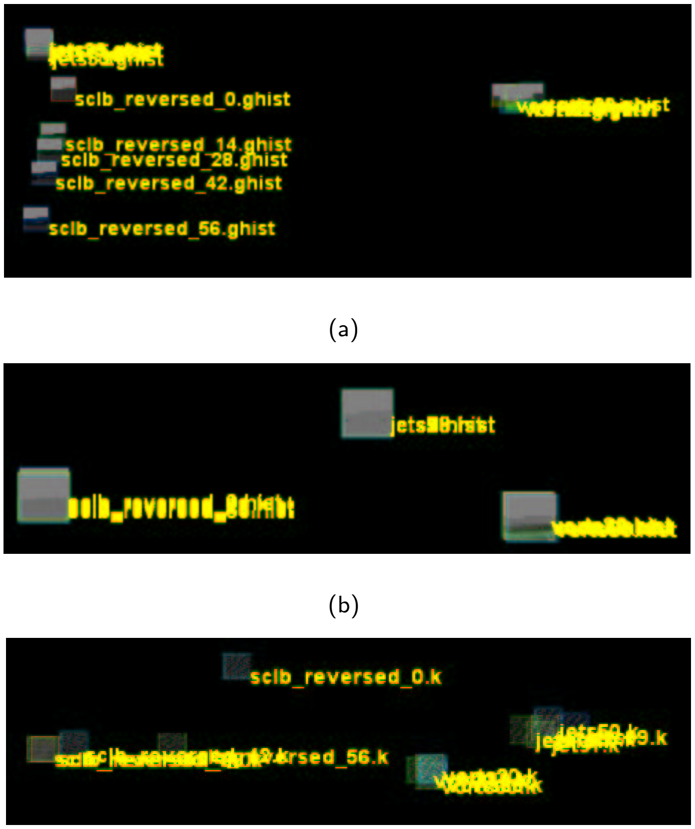
(a) (b) (c)

Figure 5.5: 3D Flows used in our experiments.

Global and local histograms and 3D SIFT were also employed with three different volumetric flows: Jet, Vortex and Shockwave (`sclb_reversed` in the pictures), all from scientific simulations. Figure 5.5 pictures all datasets in the order given above and Table 5.3 presents the main characteristics of each dataset. Five frames were extracted from each flow and processed according to the global histogram, local histograms and

Dataset name	File names	Dimensions
Turbulent Jet	`jets[1-99]`	$129 \times 129 \times 106 \times 99$
Vortex	`vorts[1-80]`	$128 \times 128 \times 128 \times 80$
Jet Shockwave 3	`sclb_reversed_[0-56]`	$256 \times 256 \times 128 \times 57$

Table 5.3: Main characteristics of the 3D flow datasets used in our experiments.



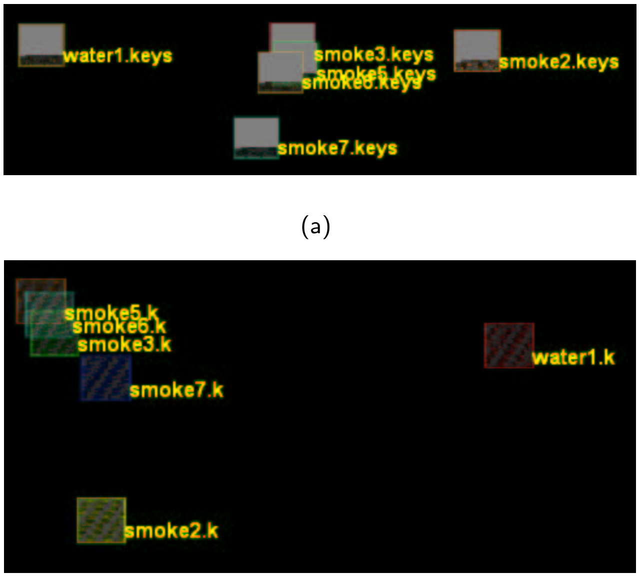
(a)

(b)

(c)

Figure 5.6: MDS plot of three volumetric flow datasets.

3D SIFT algorithms. Those frames correspond to the first and last frames and frames located at 25%, 50% and 75% of the overall length of the sequence. Figure 5.6 illustrates the difference in using global histogram, local histograms and 3D SIFT as low level descriptors in items (a), (b) and (c), respectively. Using global histograms, MDS segregates properly both the Vortex and Jet datasets but is not able to correctly cluster frames from the Shockwave dataset. If, on the other hand, local histograms are used as the low level descriptor, the diagram shows that all frames are correctly clustered. 3D SIFT descriptors provide another good clustering, except for the first frame of the Shockwave dataset, that is far from the other frames.



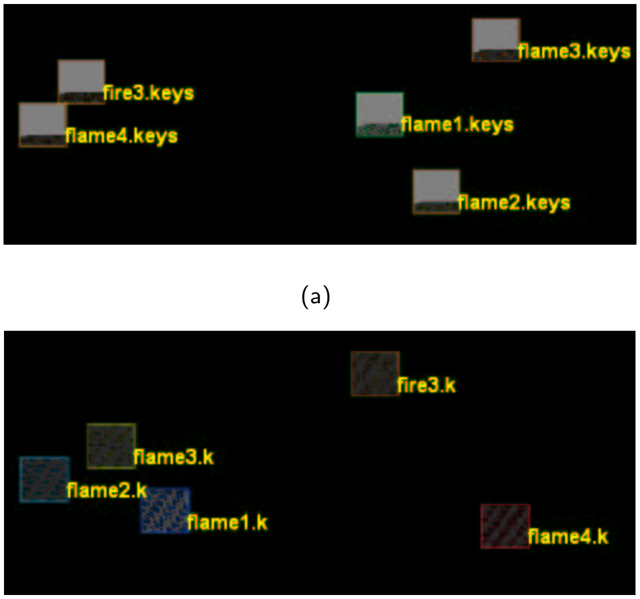
(a)

(b)

Figure 5.7: MDS plots of the Smoke Family against an unknown flown (Water Family, Flow 1).

Both descriptors, SIFT and 3D SIFT, were also used in the verification of clustering by similarity. After extracting both descriptors from several elements, tests were made to compare a member from outside the family against a set of known elements, to simulate search by example. Figure 5.7 presents an MDS plot of such comparison. Item (a) of the figure shows an MDS plot generated with SIFT keypoints descriptor and item (b) a plot generated with 3D SIFT desctiptors. Even though the distances between different flows appear shorter in (a), the diagram has been normalized and the absolute distance among the flow 2 of smoke and the main group of (b) is actually bigger than the distance between the same elements in (a), up to 30% larger.

Flows 2 and 7 are apart from the main cluster in both diagrams because they differ greatly from flows 3, 5 and 6. It is evident in both diagrams, especially in (b), that 3D SIFT provides a better depiction of flow 7 demonstrating that even though it has problems with volumes rotated by more than 10°, it matches flow 7 very well with similar, but rotated, members of the same family. Figure 5.8 presents an MDS map with the entire family of Fire plus an element from the Flame set. While the diagram in (a), generated from SIFT keypoints descriptors, portrays fire 3 and flame 4 close together and all other flame flows distant from each other, 3D SIFT, with its MDS map in (b), provides a better descriptor, allowing the MDS plot to represent the similarity of flows more accurately. Even though flame 4 is far from the main group of the family to which it belongs, this flow, if verified in Figure 5.2, can be seen to be fairly dissimilar from the other members of the family.



(a)

(b)

Figure 5.8: MDS plots of the Flame Family against an unknown flown (Fire Family, Flow 4).

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

We have described a framework that expresses sampled datasets common in the fields of volume graphics and visualization as a set of rich, yet general, salient low-level feature vectors, and then uses these to create high-level models of the sampled data objects. Here, we note that it was not our goal to achieve a semantic understanding of data, which is more along the lines of the topology-motivated analysis referenced in the introduction and pioneered by [8]. Instead, we aimed for a more general framework to support classification tasks needed in mining, browsing, searching, and retrieval of sampled data in large files or collection of files. We demonstrated these capabilities for a variety of scenes of sampled objects and phenomena: medical data and flow data.

At this stage we have built the models only by ways of K-means clustering to determine descriptive feature groupings based on similarity. This worked already quite well. However, a richer set of object dichotomies could be obtained by introducing probabilistic techniques, such as EM, into the framework, to discover more specific objects in the data [5]. Another useful extension of our framework would be to associate the learned object and feature models with information on appropriate rendering parameters, learned from user studies [6] or automated observers, such as Daly's Visible Differences Predictor (VDP) [4], or a combination of these. Here, it will be probably useful to also include other low-dimensional feature descriptors, such as density, gradients, LH, and the like. Finally, we also plan to determine better 3D SIFT detectors, more invariant to rotations.

BIBLIOGRAPHY

[1] Asimov, D. The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal on Scientific and Statistical Computing 6*, 1 (1985), 128–143.

[2] Bellman, A. *Adaptive Control Process*. Princeton University Press, Princeton, NJ, 1961.

[3] Cheung, W. and Hamarneh, G. N-SIFT: N-Dimensional Scale Invariant Feature Transform for Matching Medical Images. In *IEEE International Symposium on Biomedical Image Processing (ISBI)* (2007), pp. 720–723.

[4] Daly, S. J. The visible differences predictor: An algorithm for the assessment of image fidelity. In *Digital Image and Human Vision* (1993), The MIT Press, pp. 179–206.

[5] Fei-Fei, L., Fergus, R. and Perona, P. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision* (Washington, DC, USA, 2003), IEEE Computer Society, p. 1134.

[6] Giesen, J., Mueller, K., Schuberth, E. and Wang, L. Conjoint analysis to measure the perceived quality in volume rendering. *IEEE Transaction on Visualization and Computer Graphics 13*, 6 (2007), 1664–1671.

[7] Guha, S., Rastogi, R. and Shim, K. CURE: An efficient clustering algorithm for large databases. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 1998), ACM, pp. 73–84.

[8] Helman, J. and Hesselink, L. Representation and display of vector field topology in fluid flow data sets. *Computer 22*, 8 (1989), 27–36.

[9] Inselberg, A. and Dimsdale, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS '90: Proceedings of the 1st conference on Visualization '90* (Los Alamitos, CA, USA, 1990), IEEE Computer Society Press, pp. 361–378.

[10] Kindlmann, G. and Durkin, J. W. Semi-automatic generation of transfer functions for direct volume rendering. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization* (New York, NY, USA, 1998), ACM, pp. 79–86.

[11] Kindlmann, G., Whitaker, R., Tasdizen, T. and Moller, T. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 67.

[12] Kniss, J., Kindlmann, G. and Hansen, C. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics 8*, 3 (July 2002), 270–285.

[13] Lowe, D. G. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2* (Washington, DC, USA, 1999), IEEE Computer Society, p. 1150.

[14] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2 (2004), 91–110.

[15] Lundström, C., Ljung, P. and Ynnerman, A. Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1570–1579.

[16] Nam, E., Han, Y., Mueller, K., Zelenyuk, A. and Imre, D. Cluster-Sculptor: A visual analytics tool for high-dimensional data. In *IEEE Symposium on Visual Analytics Science and Technology* (Washington, DC, USA, 2007), pp. 75–82.

[17] Nistér, D. and Stewénius, H. Scalable recognition with a vocabulary tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2006), IEEE Computer Society, pp. 2161–2168.

[18] Pascucci, V. and Cole-McLaughlin, K. Efficient computation of the topology of level sets. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 187–194.

[19] Pinto, F. and Freitas, C. Design of multi-dimensional transfer functions using dimensional reduction. *Eurographics/IEEE VGTC Symposium in Visualization (EuroVis))* (2007), 130–137.

[20] Ropinski, T., Meyer-Spradow, J., Diepenbrock, S., Mensmann, J. and Hinrichs, K. H. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Eurographics 2008) 27*, 2 (2008), 567–576. accepted for publication.

[21] Scovanner, P., Ali, S. and Shah, M. A 3-dimensional SIFT descriptor and its application to action recognition. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia* (New York, NY, USA, 2007), ACM, pp. 357–360.

[22] Tenginakai, S., Lee, J. and Machiraju, R. Salient iso-surface detection with model-independent statistical signatures. In *VIS '01: Proceedings of the conference on Visualization '01* (Washington, DC, USA, 2001), IEEE Computer Society, pp. 231–238.

[23] Treisman, A. M. and Gelade, G. A feature-integration theory of attention. *Cognit Psychol 12*, 1 (January 1980), 97–136.

[24] Tzeng, F.-Y., Lum, E. B. and Ma, K.-L. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 273–284.

[25] Šereda, P., Bartroli, A. V., Serlie, I. W. O. and Gerritsen, F. A. Visualization of Boundaries in Volumetric Data Sets Using LH Histograms. *IEEE Transactions on Visualization and Computer Graphics 12*, 2 (2006), 208–218.

[26] van Walsum, T., Post, F. H., Silver, D., Post, F. J. and Post, F. J. Feature extraction and iconic visualization. *IEEE Transactions on Visualization and Computer Graphics 2*, 2 (1996), 111–119.

[27] Weinkauf, T., Theisel, H., Shi, K., Hege, H.-C. and Seidel, H.-P. Extracting higher order critical points and topological simplification of 3D vector fields. In *IEEE Visualization* (2005), pp. 71–78.

[28] White, D. A. and Jain, R. Similarity indexing with the SS-tree. In *ICDE '96: Proceedings of the Twelfth International Conference on Data Engineering* (Washington, DC, USA, 1996), IEEE Computer Society, pp. 516–523.

[29] Wondertouch. Particle illusion 3.0. `http://www.wondertouch.com`.