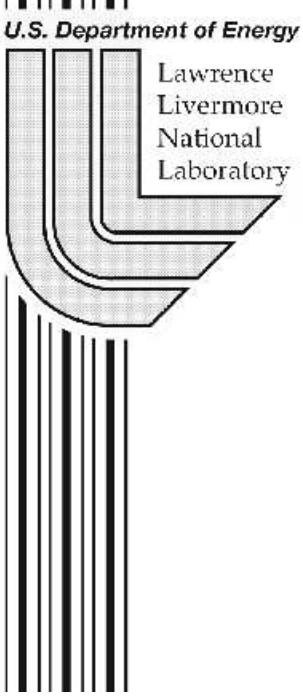
Preprint
UCRL-JC-123456

Initial experiences with retrieving similar objects in simulation data

Sen-ching S. Cheung
Chandrika Kamath

This article was submitted to the Sixth Workshop on Mining Scientific and Engineering Datasets, in conjunction with SIAM International Conference on Data Mining

May 3, 2003



This research was supported under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

Initial experiences with retrieving similar objects in simulation data

Sen-ching S. Cheung
Chandrika Kamath
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
P.O. Box 808, L-551
Livermore, CA 94551
cheung11@llnl.gov, kamath2@llnl.gov

November 24, 2003

Abstract

Comparing the output of a physics simulation with an experiment, referred to as "code validation," is often done by visually comparing the two outputs. In order to determine which simulation is a closer match to the experiment, more quantitative measures are needed. In this paper, we describe our early experiences with this problem by considering the slightly simpler problem of finding objects in a image that are similar to a given query object. Focusing on a dataset from a fluid mixing problem, we report on our experiments with different features that are used to represent the objects of interest in the data. These early results indicate that the features must be chosen carefully to correctly represent the query object and the goal of the similarity search.

1 Introduction

Computer simulations are increasingly being seen as the third mode of science, complementing theory and experiments. In order to validate the physics models in these simulations, their results must be compared with an experiment using quantitative measures - this is referred to as "code validation". In this paper, we describe how data mining and information retrieval techniques can be used to aid the validation of a simulation with an experiment. We consider the problem of shock-driven mixing of two fluids of different densities. When the interface between them is accelerated by a shock wave striking the interface perpendicularly, it results in an instability referred to as the Richtmyer-Meshkov instability [1, 2]. This instability occurs in various natural and man-made settings such as supernova explosions, the interiors and wakes of jet engines, combustion chambers, etc. It is therefore important to understand and model this instability accurately. In recent years, researchers have been able to produce the Richtmeyer-Meshkov instability in high-quality experiments. This data is now being used to validate simulation codes in order to determine the numerical techniques that best match the results in the experiments [3, 4].

As the first step toward code validation in the context of this particular problem, we consider the more general task of identifying similar "objects" in simulation data. This is motivated by the fact that the image of two fluids mixing, as shown in Figure 1, has clearly identifiable "mushroom" shaped objects. If we could quantitatively measure the similarity of these objects taken in isolation, we could then combine this measure with additional information such as the number and locations of the mushrooms to quantitatively compare the experimental image with the ones from simulations. In fact, a system that can automatically identify visually similar objects in simulation data has other applications beyond code validation. For example, by combining such a system with visualization software, scientists can quickly focus on selected regions in large dataset that are visually similar to a pre-defined object of interest. In this paper, we focus on the task of identifying objects in the simulation data that are similar to a given query object. Our primary approach is to represent objects in terms of carefully-designed numerical features such that features of similar objects in different orientations, scales, and resolutions are close to each other. This paper is organized as follows. After reviewing related work in Section 2, we describe a preliminary implementation of our Similarity-Based Object Recognition (SBOR) system for simulation data. This system provides a test-bed for evaluating dif-

ferent features in retrieving similar objects. Initial experiments with a number of simple features using data from turbulence simulation are reported in Section 4. We conclude this paper and highlight some of our ongoing work in Section 5.

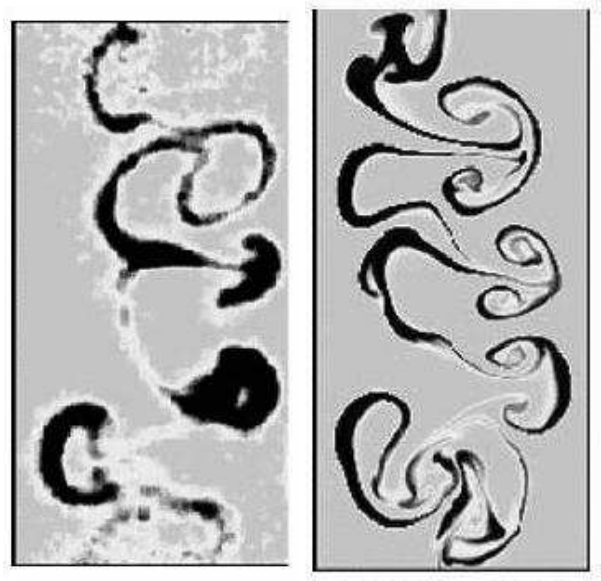


Figure 1: *The left image shows the flow pattern of a Richtmyer-Meshkov experiment performed at Los Alamos National Laboratory. The same experiment is simulated by high resolution numerical methods and the result is shown in the right image [3].*

2 Related Work

Much of the research on pattern recognition for turbulence data has been focused on extracting and tracking topological features such as flow lines and vortexes for visualization [5, 6, 7], and identifying high-level events such as bursting and shock waves based on these features [8, 9]. These works typically assume a single large fluid dataset, and the goal is to allow scientists to visualize the large amount of data available and build models to explain the underlying phenomenon. On the other hand, our goal of code validation is to compare and validate datasets from simulations with experiments. In general, these datasets contain different physical measurements, and vary greatly in resolution and precision. As such, we need to construct a system that can support a large array of different features and provide robust methods for feature extraction and comparison. The similarity-based approach described in this paper is inspired by the recent progress in the area of Content-Based Image Retrieval (CBIR).

CBIR systems exploit various features derived from the images and model visual similarity by mathematical distance functions between feature vectors. Extensive research has been performed to derive compact and representative features and distance functions to model visual cues such as color, texture, and shapes. Excellent reviews of different CBIR systems can be found in [10, 11, 12, 13, 14, 15, 16, 17]. All of these systems focus on photographic imagery, remotely sensed images, medical images or geologic images. To the best of our knowledge, our work is the first to consider the application of content-based approach to turbulence simulation data. Turbulence data differs from other types of imagery in that they do not have clear object definitions, and there are a multitude of physical quantities associated with each physical location. In addition, much of the existing research focus on capturing the salient features of the entire image. A related problem arises when the query object is not an image, but a part of an image. For example, instead of using the entire image in Figure 1 as a query, a scientist might outline just one of the "mushroom"-shape structures as the object of interest. In this case, the problem of CBIR becomes more complex as we now need to find sub-images that are a close match. To clearly identify this added level of complexity, we refer to this problem as similarity-based object retrieval (SBOR). There are two approaches to the SBOR problem: data-independent and data-dependent [18]. In the data-independent approach, images are divided into overlapping or non-overlapping rectangular regions or tiles, and feature vectors are extracted from each tile and stored in a database for similarity search [19, 20]. Data-dependent approaches, on the other hand, apply object segmentation algorithms to extract objects from images and perform similarity search on feature vectors representing individual objects [21, 22]. Due to the small size and fine granularity of tile images, the data-independent approach typically generate much larger amount of feature data than the data-dependent approach. On the other hand, the data-independent approach is more flexible and accurate as it is feasible to incorporate the query object as part of the input to the object segmentation and extraction algorithms. Our work will primarily focus on the data-independent approach.

3 Proposed System

In this section, we describe a preliminary implementation of a SBOR system for simulation data. There

are three major modules in the system: graphical user interface, feature extraction, and similarity search. Their relationship is depicted in Figure 2. In a typical similarity search, a user first opens an image from the image database and defines a rectangular tile on the image as the query image. An example is shown in Figure 3, in which the top right corner of the image is used as a query image. Then, the user specifies the types of features to be used in the similarity search. The user can select from a large array of features, ranging from simple pixel statistics to complicated visual attributes such as shape and texture. Based on the user's familiarity of the system, the user may start with simple features to obtain a quick response, and then refine the results with more sophisticated features. On the other hand, the user may choose a particular combination of features to exactly pinpoint the characteristics of interest.

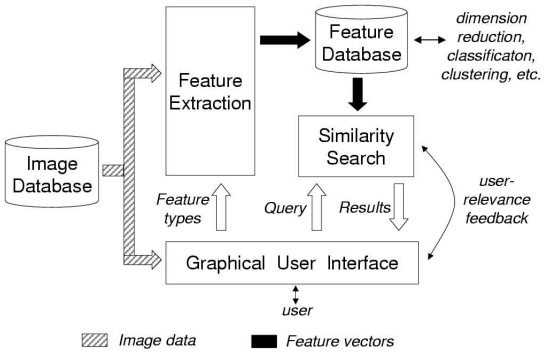


Figure 2: *Schematic diagram of the proposed similarity-based object recognition system.*

Based on the types of features chosen by the user, the feature extraction module populates the feature database with feature vectors extracted from images in the database. We adopt a simple sliding-window approach in generating feature vectors from images. A tile window, with dimensions same as the query image, is moved across each image in a fixed step-size. A feature vector, which contains all features selected by the user, is computed for the part of the image under the tile window at each location. In the experiments reported here, a small step-size of two pixels is used for both the horizontal and vertical directions in order to capture spatial variations of the data. This results in overlapping tiles. Other step-sizes are also possible. The feature vector, the location of the tile image, and the ID of the original image in the image database are stored in the feature database. Even though our focus for this paper is the design of features for similarity search, it should be noted that this feature database is also amenable to more sophisticated pattern recognition tools such as dimension reduction, classification, and clustering.

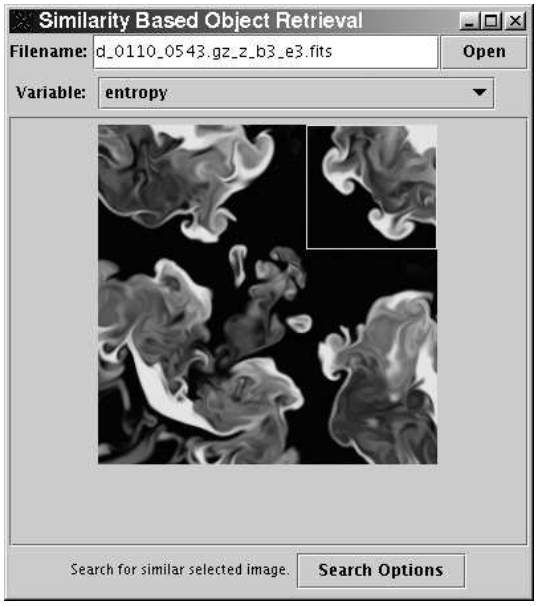


Figure 3: *Screen-shot on how to specify a query tile image.*

With the feature database in place, the similarity search module seeks out the feature vectors in the database that are "similar" to the feature vector corresponding to the query image. To properly define the notion of similarity, we assume that there is a distance, or dissimilarity, function associated with each type of feature. Two feature vectors that are a small distance apart are regarded to be more similar to each other than those with a large distance between them. Some of the most commonly-used distance functions are described in detail in [18] and [23, ch. 11]. Based on the distances between the query feature and the features in the feature database, the similarity search module supports two types of search functions: ϵ-search and k-Nearest-Neighbor (k-NN) search. In a ϵ-search, the module returns all feature vectors in the database whose distance from the query feature is within a positive threshold ϵ. ϵ-search is intended for experienced users who can correlate distance values with the level of similarity. In a k-NN search, the k feature vectors in the database closest to the query feature are returned. A small k in a k-NN search allows a user to quickly examine the small set of returned results and assess how relevant a particular feature is in a similarity-search task. When more than one feature is used a similarity search,

the results on individual features can be combined by conjunction and/or disjunction. Figure 4 shows an example of how to combine two features in a single similarity search.

Figure 4: *This figure illustrates how a similarity search is defined. The top left image is the query image specified in Figure 3. Two features, "Histogram" and "ART2D", are used in this search, and their respective search parameters are specified in the two rectangular regions shaded in different gray levels.*

The returned results are finally presented to the user via the graphical user interface. Based on the returned results, the user can refine the search by modifying different search parameters and specifying different sets of features to be used. Except for expert users who are very familiar with the system, search refinement can be a daunting task due to the large number of parameters and feature available. A more intuitive approach, called "user-relevance feedback" is to have the user identified relevant and non-relevant entries among all the returned results, and apply machine-learning techniques to infer appropriate modifications in search parameters. We are currently investigating a number of user-relevance feedback techniques that are pertinent to our system.

4 Experiments

For the work in this paper, we consider the data from a high resolution 3-D shock tube simulation performed on a $2048 \times 2048 \times 1920$ grid over 27,000 time steps, obtained on 960 nodes of the IBM-SP Sustained Stewardship TeraOp system at Lawrence Livermore National Laboratory [24]. At the beginning of the simulation, two gases are separated by a membrane in a tube; then the membrane is pushed against a wire mesh. The simulation models the resulting mixing of the two gases.

Several variables are output by the simulation at each grid point at each time step. These include pressure, density, velocity, etc. In our initial work, we focus on the entropy which is available in Brick-of-Byte (BOB) files, with one byte of information per grid point. This information is the entropy scaled linearly with a minimum of 0 and a maximum of 255. Each node of the IBM-SP is responsible for a 256×256 piece of the data, and there is a BOB file for each of these pieces. Each node of the system generates a file for the output from each time step. For a preliminary testing of our SBOR system, we randomly select two 2-D slices along the wire mesh perpendicular to the direction of the impact. The following transformations are then applied to these two slices:

- Anti-clockwise rotation by 36^o (rot36), 90^o (rot90), and 150^o (rot150).
- Reflection about the vertical axis in the middle (flip).
- Morphological erosion (erode) and dilation (dilate) by a 3×3 cross-shaped element.

Examples of these transformation together with the original image are shown in Figure 5.

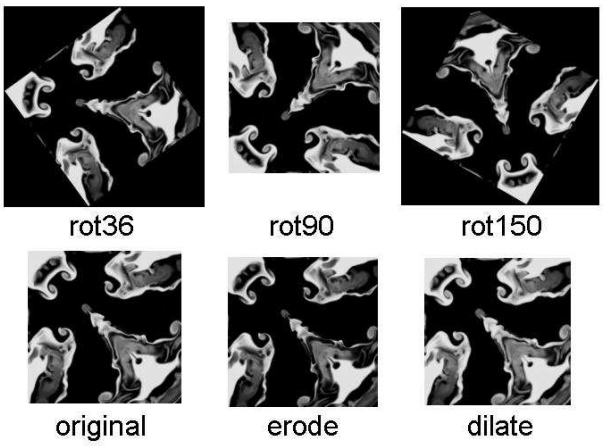


Figure 5: *One of the original 2-D slices and the corresponding five spatial transformations.*

Five query tile images of distinctive patterns are selected from the two original slices. All query tile images are of dimension 64×64, and their locations within the slices are shown in Figure 6. The goal of the experiment is to test how well different features

can identify these query patterns in all the transformed slices described above. As explained in Section 1, it is desirable to use features that can robustly handle geometrical and spatial transformations. Morphological erosion and dilation are included to mimic the resolution difference between turbulence data obtained from simulations and physical experiments.

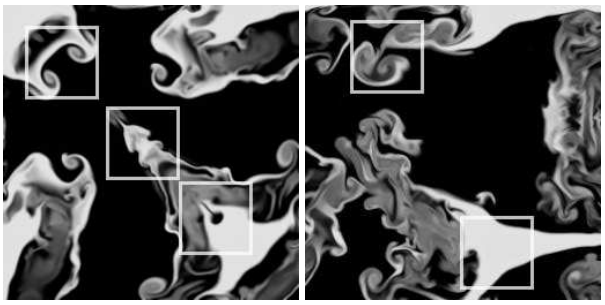


Figure 6: *2-D slices of entropy data from simulations. The five bounding boxes indicate the query images used for testing.*

4.1 Feature Definitions

This section describes the features tested in our experiments. We focus primarily on features that provide a general and compact description of how pixel values are distributed inside a tile image. For the experiments reported in this paper, the l_1 metric, or sum of absolute differences between coordinates, is used as the distance function for all features. The list of features used in our experiment include:

Simple Features This is a four dimensional vector that consists of the mean, the standard deviation, the maximum, and the minimum of all pixel values in a tile image.

Histogram This is a 16-bin histogram of pixel values in a tile image. The bins are uniform across the dynamic range.

ART Angular Radial Transform (ART) belongs to a broad class of shape analysis tools based on moments [25]. Our implementation of ART is based on the region-shape descriptor defined in MPEG-7 [26, ch. 15]. ART projects a 2-D signal within the unit circle onto a set of complex orthonormal basis functions. The ART basis function of angular order n and radial order n in polar coordinates is given by

$$V_{nm}(\rho,\theta) = \begin{cases} \exp(jm\theta)/2 & n = 0 \\ \exp(jm\theta)\cos(\pi n\rho) & n \neq 0 \end{cases}$$

The ART coefficient F_{mn} of a 2-D signal $f(\rho,\theta)$ is defined by

$$F_{mn} = \frac{1}{\pi}\int_0^{2\pi}\int_0^1 V^*_{nm}(\rho,\theta) f(\rho,\theta)\rho\, d\rho\, d\theta \quad (1)$$

The actual implementation of ART feature is a discretization of Equation (1). The origin of the functions is set at the centroid of the image. Rotational invariance is achieved by using only the magnitude of F_{mn}, and scale invariance is achieved by normalizing all F_{mn} by the area of the image, i.e. F_{00}. Following the MPEG-7 standard, twelve angular basis and three radial basis are used. This results in a 35-dimensional feature vector as the normalized F_{00} is always one and thus dropped from the representation. Unlike MPEG-7, we do not quantize the coefficients and retain the full floating-point precision for similarity search.

BART As alluded to in Section 1, we believe that shape is a very important attribute in identifying similar objects in turbulence data. To provide a description of the shape of a 2-D object independent of the internal pixel values, we propose a slight modification of ART called the Binary ART (BART) feature. A simple adaptive thresholding scheme is first applied to the input tile image to convert it to a binary image, with the foreground pixels set to 255 and the background pixels to zero. The threshold is chosen to provide a good definition of the object boundary. It is set to be the first minima of a 32-bin histogram of all pixel values in the input tile image. The BART feature is defined to the ART feature of the resultant binary image.

4.2 Experimental Results

We follow the procedure described in Section 3 for testing: a 64×64 tile window is moved across each transformed slice in a step-size of two pixels horizontally and vertically. In order to be robust against rotation at an arbitrary angle, we only consider the pixels within the largest circle inscribed in the tile window for feature extraction. Within each transformed slice, we identify those tile images that overlap more than 90% with a query pattern to be the ground-truth for that particular query. A partial-overlap criteria is used because first, features should be robust against small translations induced by partial overlapping, and second, complete overlap is simply unachievable for some of the transformations such as rot36 and rot150. After establishing the ground-truth, we use two metrics to measure how well the

images in the ground-truth can be identified by a particular feature. These two metrics are Average Normalized Modified Retrieval Rate (ANMRR) and the Average Total Rank (ATR). We prefer these metrics to the standard precision-recall curve [27] as they provide a single score for each feature under each transformation.

ANMRR was originally proposed by the MPEG-7 committee in evaluating visual features [26, ch. 12]. ANMRR is computed as follows. For the particular feature being tested, we first compute the feature vector q for the query image and the feature database D of all the tile images. Let the ground-truth for q be $G(q) \subset D$. We then perform a k-NN search with k equal to twice the size of $G(q)$. For each feature vector $x \in G(q)$, we define a ranking score $r(x)$ to be the rank of x if x is returned in the k-NN search, or $1.25 \cdot k$ if x is not retrieved at all. We define the Normalized Modified Retrieval Rank (NMRR) for query q by averaging the ranking scores for all the feature vectors in $G(q)$, and normalizing the result so that it is within the range of zero and one. Mathematically, NMRR(q) is defined by the following equation:

$$\mathrm{NMRR}(q) = \frac{\sum_{x \in G(q)} r(x)/|G(q)| - 0.5 * (1 + |G(q)|)}{1.25 * k - 0.5 * (1 + |G(q)|)} \quad (2)$$

If feature vectors in the ground-truth are among the top-ranked results from the k-NN search, NMRR(q) is close to zero. On the other hand, if none of the ground-truth are retrieved, NMRR(q) becomes one. ANMRR is simply the average of the NMRR scores of all the testing queries. Figure 7 shows the ANMRR scores for the four features described in Section 4.1 under different transformations. ART produces an average ANMRR value of 0.22, which is the lowest among all the features experimented. Its advantage over other features is especially pronounced in erode and dilate, the two morphological transformations. Unlike simple features and histogram, ART takes into account the spatial relationship between pixel values. Thus, it is more robust against transformations such as erosion and dilation which modify the distribution of pixel values. The similar BART feature performs slightly worse. The reason is that the binarization in BART completely discards the variation in pixel values, which is an important cue to discern similar patterns in the transformed images used in our experiments.

ANMRR is very useful in quantifying the performance of interactive similarity search [28]. However, a similarity search based on a few simple features is unlikely to capture all possible variations of similar objects. Further refinement of the search results

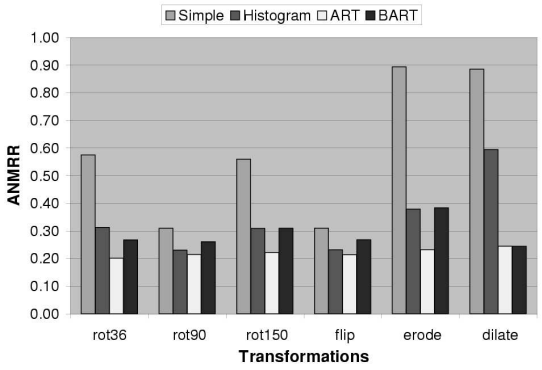


Figure 7: *ANMRR of the tested features under various transformations.*

based on computationally intensive pattern recognition techniques is often required. Thus, it is important not to prematurely prune away any possible matches in the early stages of similarity search. At the same time, it is desirable to keep the set of search results as small as possible in order to support interactive exploration and reduce complex computation for refinement. To this end, we define an alternative metric called Total Rank to measure how well a feature can capture all similar objects without an excessive number of false positives. We first perform an ϵ-search on a query q by setting ϵ equal to the maximum distance between q and all the feature vectors in the ground-truth $G(q)$. Denote the set of feature vectors retrieved by $T(q)$. We define the total rank to be the ratio between the sizes of $T(q)$ and $G(q)$. A small total rank indicates a small number of false positives. ATR is the average of the total ranks for all five testing queries. Figure 8 shows the ATR for the four feature vectors under different transformations. Histogram produces the lowest ATR among all the features. On the contrary, ART and BART, which show very good performance in terms of ANMRR, are the two worst performers in this experiment. We attribute this reversal of performance to our approach in constructing the ground-truth set. Recall that some of the tile images in our ground-truth have only 90% overlap with the actual query object. Due to the large overlap, the histograms of these tile images and the query object are still very close to each other. On the other hand, every ART coefficient, as defined in Equation (1), depends on all the pixels in an image. Depending on which 10% of query object pixels are missing in the tile images, the corresponding ART coefficients may change significantly. As a result, the ART distance between the

query object and some of the tile images in its ground-truth set may become exceedingly large, leading to a large ATR value.

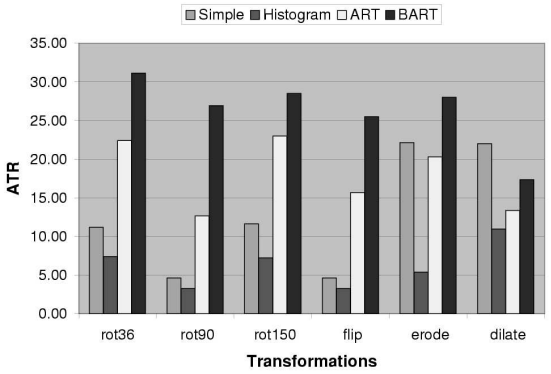


Figure 8: *Total rank of the tested features under various transformations.*

5 Conclusions and Future Work

In this paper, we have described our initial effort to build a SBOR system for code validation of simulation data. Our basic approach is to first capture the salient features of the local structure or object as a multi-dimensional feature vector, and then identify similar objects based on distances between the corresponding feature vectors. We have compared four different types feature vectors, including simple, histogram, ART, and BART, in identifying objects under various spatial transformations. ART provides the best overall retrieval results. Histogram, on the other hand, outperforms others in retaining the entire ground-truth with the least number of false positives.

Our initial design generates feature vectors on-the-fly using a tile size identical to that of the query. Even though this approach is adaptive to the input query, it does not scale well with the size of the database. We are currently developing a two-step approach to rectify this problem. In the first step, a set of simple but general features are computed offline based on a number of pre-defined tile sizes. Similarity search on these features provides a crude estimate on the locations of the objects of interest. As these features are generated offline, we can apply dimension reduction and indexing structures such as R-Tree to achieve faster-than-linear search performance [18]. In the second stage, the user can refine the search results from the first step by computing query-specific features on the target area, or applying learning algorithms to incorporate user feedback. Another area we are investigating is to extend the current system from handling just a single variable of entropy to multiple variables, as well as from 2-D slices to the entire 3-D dataset.

6 Acknowledgments

We would like to thank Nu Ai Tang for building the graphical user interface module for our system and the second reviewer for the insightful comment.

UCRL-JC-151931: This work was performed under the auspices of the National Laboratory under contract No. W-7405-Eng-48.

References

[1] R. D. Richtmyer, "Taylor instability in shock acceleration of compressible fluids," *Communuinications in Pure and Applied Mathematics*, vol. 13, no. 297, pp. 297–319, 1960.

[2] E.E. Meshkov, "Instability of the interface of two gases accelerated by a shock wave," *Izv. Acad. Sci. USSR Fluid Dynamics*, vol. 4, pp. 101–104, 1969.

[3] W. Rider et al., "Using Richtmyer-Meshkov driven mixing experiments to impact the development of numerical methods for compressible hydrodynamics," in *Proceedings of the Ninth International Conference on Hyperbolic Problems Theory, Numerics, Applications*, 2002, http://www.acm.caltech.edu/hyp2002/-program.html.

[4] C. A. Zoldi, *A numerical and experimental study of a shock-accelereated heavy gas cylinder*, Ph.D. thesis, State University of New York at Stony Brook, 2002.

[5] H. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *Computer*, vol. 22, no. 8, pp. 27–36, 1989.

[6] M. Jiang, R. Machiraju, and D. Thompson, "A novel approach to vortex core region detection," in *Proceedings of Joint Eurographics-IEEE TCVG Symposium on Visualization*, May 2002, pp. 217–225.

[7] F. Post et al., "Feature extraction and visualization of flow fields," in *State-of-the-Art Proceedings of Eurographics 2002*, Sept. 2002, pp. 69–100.

[8] K.-L. Ma, J. van Rosendale, and W. Vermeer, "3d shock wave visualization on unstructured grids," in *Proceedings 1996 Symposium on Volume Visualization*, 1996, vol. 104, pp. 87–94.

[9] E.-H. Han, G. Karypis, and V. Kumar, "Data mining for turbulent flows," in *Data mining for scientific and engineering applications*, R. Grossman et al., Eds., pp. 239–256. Kluwer Acadmic Publishers, 2001.

[10] V. Castelli and D. Bergman, Eds., *Image Databases: Search and Retrieval of Digital Imagery*, John Wiley & Sons, Inc., 2002.

[11] C. Djeraba et al., "Special issue on content-based multimedia indexing and retrieval," *IEEE Multimedia Magazine*, vol. 9, no. 2, pp. 18–60, 2002.

[12] R. C. Veltkamp, H. Burkhardt, and H.-P. Kriegel, Eds., *State-of-the-Art in Content-Based Image and Video Retrieval*, Kluwer Academic publishers, 2001.

[13] M.Yeung et al., "Special section on storage, processing, and retrieval of digital media," *Journal of Electronic Imaging*, vol. 10, no. 4, October 2001.

[14] B. Perry et al., *Content-based access to multimedia information – from technology trends to state of the art*, chapter 4.3, Kluwer Academic Publishers, Massachusetts, U.S.A., 1999.

[15] D. Forsyth, J. Malik, and R. Wilensky, "Searching for digital pictures," *Scientific American*, pp. 88–93, June 1997.

[16] C. Faloutsos, *Searching Multimedia Databases by Content*, Kluwer Academic Publishers, 1996.

[17] R.W. Picard, A.P. Pentland, et al., "Special issue on digital libraries," *IEEE Transactions on Pattern Analysis and Machine intelligence*, vol. 18, no. 8, August 1996.

[18] V. Castelli, "Multidimensional indexing structures for content-based retrieval," in *Image Databases: Search and Retrieval of Digital Imagery*, V. Castelli and L. D. Bergman, Eds. John Wiley & Sons, Inc., 2002.

[19] C.-S. Li and V. Castelli, "Deriving texture feature set for content-based retrieval of satellite image database," in *Proceedings of IEEE International Conference Image Processing, ICIP'97*, Santa Barbara, CA, Oct. 1997, pp. 567–579.

[20] C.-S. Li et al., "Comparing texture feature sets for retrieving core images in petroleum applications," in *Proc. SPIE Storage Retrieval Image Video Database VII*, San Jose, CA, 1999, vol. 3656, pp. 2–11.

[21] C. Carson et al., "Region-based image querying," in *Proc. of IEEE CVPR'97 Workshop on Content-Based Access of Image and Video Libraries*, San Jan, Puerto Rico, 1997, pp. 42–49.

[22] B.S. Manjunath, "Image processing in the Alexandria digital library project," in *Proceedings of IEEE International Forum on Researchand Technology, advances in Digital Libraries - ADK'98*, Santa Barbara, CA, 1998, pp. 180–187.

[23] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

[24] A. Mirin et al., "Very high resolution simulation of compressible turbulence on the IBM-SP system," Tech. Rep. UCRL-JC-134237, Lawrence Livermore National Laboratory, 1999.

[25] R. Mukundan and K. R. Ramakrishnan, *Moment Functions In Image Analysis: Theory and Applications*, World Scientific, 1988.

[26] B.S. Manjunath, P. Salembier, and T. Sikora, Eds., *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley & Sons, Ltd., 2002.

[27] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill computer science series, 1983.

[28] P. Ndjiki-Nya et al., "Subjective evaluation of the mpeg-7 retrieval accuracy measure (ANMRR)," Tech. Rep. M6029, ISO/IEC JTC1/SC29/WG11 (MPEG), May 2000, Geneva.