Retrieval of Similar Objects in Simulation Data Using Machine Learning Techniques

Erick Cantú-Paz,[a] Sen-ching S. Cheung,[a] Chandrika Kamath[a]

[a]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-551, Livermore, CA 94551

ABSTRACT

Comparing the output of a physics simulation with an experiment 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. This paper describes 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 using classification techniques from machine learning to retrieve the objects of interest in the simulation data. The early results reported in this paper suggest that machine learning techniques can retrieve more objects that are similar to the query than distance-based similarity methods.

Keywords: Machine learning, classification, similarity-based object retrieval, simulation data, turbulence

1. INTRODUCTION

Computer simulations are increasingly being seen as the third mode of science, complementing theory and experiments. Simulations provide qualitative and quantitative insights into physical phenomena. Simulations are particularly useful when the phenomena are too complex to be studied analytically or too expensive, impractical, or dangerous to study experimentally. In order to validate the physics models in these simulations, their results must be compared with an experiment using quantitative measures—this process is known 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 known 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, and combustion chambers. 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 "objects" similar to a query that the user finds interesting 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. Our

Send correspondence to cantupaz@llnl.gov

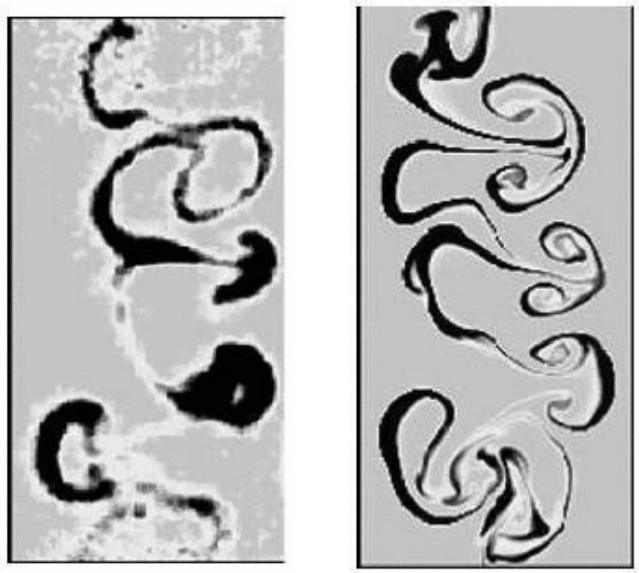

Figure 1. *The left image shows the flow pattern of a Richtmyer-Meshkov experiment performed at Los Alamos National Laboratory. The same experiment is simulated by high resolution numerical methods and the result is shown in the right image.*[3]

objective in this paper is to show that inductive classification algorithms can be used to identify objects similar to a query object. We compare the results obtained using a naive Bayes classifier with the results of ϵ-search, which consists on returning objects whose distance from the query object is within the positive threshold ϵ.

This paper is organized as follows. After reviewing related work in Section 2, we describe a preliminary implementation of our Similarity-Based Object Retrieval (SBOR) system for simulation data. This system provides a test-bed for evaluating different features in retrieving similar objects. Section 3 describes the operation of the SBOR system and how machine learning methods can be integrated into it. Section 4 describes the data used in this study. Initial experiments with a number of simple features using data from turbulence simulation are reported in Section 5. We present conclusions and highlight some of our ongoing work in Section 6.

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,[5–7] 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 references 10–17. 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. SBOR AND MACHINE LEARNING

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. 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. Later in this section we describe the feature types in detail.

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-size step. A feature vector 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. 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.

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 references 18 and 23. For the experiments in this paper, we used ϵ-search, which consists on returning all feature vectors in the database whose distance from the query feature is within a positive threshold ϵ. Other types of distance-based searches are possible.

To use inductive machine learning techniques to retrieve similar objects, we first need to create a training set with positive and negative examples. The output of the similarity-based search is used as the positive examples and a random sample of 200 tiles from the feature database are used as the negative examples. It is possible that the similarity search incorrectly identifies some tiles as being similar to the query object when they are not. It is also possible (but generally unlikely) that sampling randomly to obtain the negative examples would include tiles that are visually similar to the query object, but were not retrieved by the ϵ-search. This would include examples incorrectly mislabeled as negative into the training set. In any case, the training set is presented to the user, who can examine and relabel the examples if necessary.

For the experiments reported in this paper, a naive Bayesian classifier is trained using the entire training set and applied to label all the feature vectors in the database as being either "similar" or "not-similar" to the query. The feature vectors labeled as "similar" are presented to the user.

Other classifiers, such as decision trees or k-nearest neighbors, could be used instead or the naive Bayes. Future work will address this topic.

4. DATA PREPARATION

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 variables include pressure, density, velocity, etc. In this 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.

We focus primarily on features that provide a general and compact description of how pixel values are distributed inside a tile image. The list of features used in our experiment include:

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

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] ART projects a 2-D signal within the unit circle onto a set of complex orthonormal basis functions.

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 minimum 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.

The feature vectors in the feature database contain fairly low-level features such as ART coefficients and the bin counts for the histograms. We first tried using these low-level features directly as input to machine learning techniques. However, we realized that the individual elements in the feature vectors do not represent anything meaningful in this context. Consider, for example, basing a discrimination on the 10-th bin of the histogram or on the fourth ART coefficient. So, we also computed a set of "derived" features which are different distances between features of the query and features of each element in the database. For example, a derived feature might be the L1 distance between the intensity histogram of the query and the histogram of each example in the feature database. There is a new derived feature vector for each feature vector in the database. The derived features calculate L1, and L2 distances between all the features as well as the Kullback-Leibler and Chi-square distances between histograms.

Calculating derived features also has the advantage of reducing the dimensionality of the problem. For example, the first data set used in the experiments has feature vectors with 90 elements, but only 20 derived features. The derived features corresponding to the same original feature as highly correlated. However, there is no obvious way to choose among the different distances. An exploratory data analysis showed that simple thresholding on any one of these distances would result in several false positive results, but several of these distances considered together would reduce the number of errors. However, using distances as the derived features implies that the features are dependent on the query, and must be recalculated as the query changes.

Figure 2. The three queries used in the experiments.

Figure 3. The results of applying the naive Bayes classifier to retrieve objects similar to the first query. The first two results were also retrieved by the ϵ-search.

5. EXPERIMENTS

We conducted experiments with different query objects. In this section we present the results of the experiments with three queries that illustrate different advantages of using machine learning techniques to retrieve similar objects. The three queries are shown in figure 5.

In the case of the first query, ϵ-search with ϵ=1.5 returned three matches (including the query) and no false matches. The training set was composed of these two matches as the positive examples and 200 negative examples that were sampled randomly from the feature database. The naive Bayesian classifier was trained and used to label all the examples in the feature database as "similar" or "not-similar." Figure 5 presents the three "similar" objects (besides the query) identified by the naive Bayes. Using the naive Bayes resulted in finding one additional similar object.

For the second query, we chose $\epsilon = 4.0$ to minimize the number of false matches returned by the similarity search. With this threshold, the ϵ-search returned two matches (including the query) and no false matches. The naive Bayes classifier was trained using these two matches labeled as positive examples and 200 randomly chosen examples labeled as negative. The results of applying the trained naive Bayes as shown in figure 5. Again, the naive Bayes classifier obtained more true matches than with the ϵ-search and no false matches.

The third query produced interesting results. ϵ-search with $\epsilon = 1.5$ returned 20 true matches, including some rotated examples. Figure 5 has a sample of the results. The naive Bayes was trained with these 20 positive examples and 200 randomly selected negative examples. Applying the trained classifier to label the items in the

Figure 4. The results of applying the naive Bayes classifier to retrieve objects similar to the second query. The first result was also retrieved by the ϵ-search.

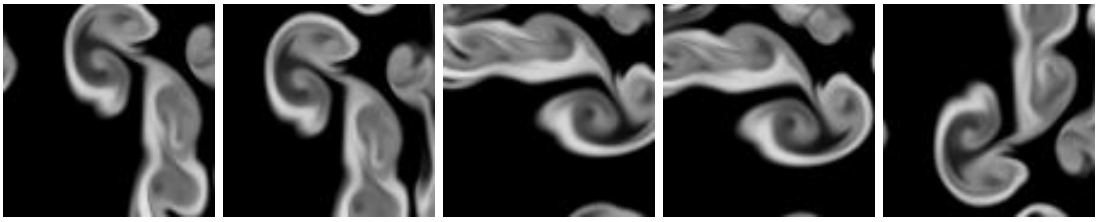

Figure 5. A sample of results of ϵ-search to retrieve objects similar to the third query.

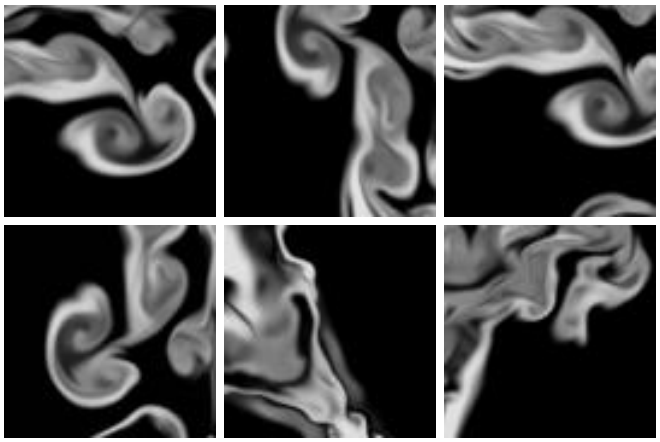

Figure 6. A sample of results of applying the naive Bayes classifier to retrieve objects similar to the third query.

database resulted in 96 items being labeled as "similar" to the query. Of these 96 items, 30 were false positives and 66 were true matches. Figure 5 has a sample of the results.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have described our initial efforts to use machine learning techniques to retrieve objects similar to a query in 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 use ϵ-search to identify an initial set of matches. A training set is created by joining the initial set of matches with a random sample of objects. The training set is used to train a naive Bayes classifier, which is used to retrieve objects similar to the query from the database of feature vectors. We have shown results where the naive Bayes classifier retrieves more true matches than ϵ-search. The system using naive Bayes is also easier to use, since the ϵ threshold must be tuned to each query by experimenting, while the naive Bayes does not have any user-tunable parameters.

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-Trees 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.

ACKNOWLEDGMENTS

We would like to thank Nu Ai Tang for building the graphical user interface module for our system.

UCRL-JC-153866. This work was performed under the auspices of the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

REFERENCES

1. R. D. Richtmyer, "Taylor instability in shock acceleration of compressible fluids," *Commununications in Pure and Applied Mathematics* **13**(297), pp. 297–319, 1960.
2. E. Meshkov, "Instability of the interface of two gases accelerated by a shock wave," *Izv. Acad. Sci. USSR Fluid Dynamics* **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*, pp. 84 – 88, 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*. PhD 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* **22**(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*, pp. 217–225, May 2002.
7. F. Post *et al.*, "Feature extraction and visualization of flow fields," in *State-of-the-Art Proceedings of Eurographics 2002*, pp. 69–100, Sept. 2002.
8. K.-L. Ma, J. van Rosendale, and W. Vermeer, "3d shock wave visualization on unstructured grids," in *Proceedings 1996 Symposium on Volume Visualization*, **104**, pp. 87–94, 1996.
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* **9**(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* **10**, October 2001.
14. B. Perry *et al.*, *Content-based access to multimedia information – from technology trends to state of the art*, ch. 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. Picard, A. Pentland, *et al.*, "Special issue on digital libraries," *IEEE Transactions on Pattern Analysis and Machine intelligence* **18**, 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*, pp. 567–579, (Santa Barbara, CA), Oct. 1997.
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*, **3656**, pp. 2–11, (San Jose, CA), 1999.
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*, pp. 42–49, (San Jan, Puerto Rico), 1997.

22. B. Manjunath, "Image processing in the Alexandria digital library project," in *Proceedings of IEEE International Forum on Research and Technology, Advances in Digital Libraries - ADK'98*, pp. 180–187, (Santa Barbara, CA), 1998.
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. Manjunath, P. Salembier, and T. Sikora, eds., *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley & Sons, Ltd., 2002.