Learning to Predict Salient Regions from Disjoint and Skewed Training Sets

Larry Shoemaker, Robert E. Banfield, Lawrence O. Hall
University of South Florida
Department of Computer Science and Engineering
4202 E. Fowler Ave, Tampa, FL 33620-9951
{ lwshoema, rbanfiel, hall }@cse.usf.edu

Kevin W. Bowyer
University of Notre Dame
Department of Computer Science and Engineering
South Bend, IN 46556, USA
kwb@cse.nd.edu

W. Philip Kegelmeyer
Sandia National Laboratories
Computational Science and Math Research Department
P.O. Box 969, MS 9291
Livermore, CA 94551-0969
wpk@ca.sandia.gov

Abstract

We present an ensemble learning approach that achieves accurate predictions from arbitrarily partitioned data. The partitions come from the distributed processing requirements of a large scale simulation where the volume of the data is such that classifiers can train only on data local to a given partition. As a result of the partition reflecting the need for efficient simulation analysis, rather than the needs of data mining, the class statistics vary across partitions; indeed some classes will likely be absent from some partitions. We combine a fast ensemble learning algorithm with majority voting to generate an accurate working model of the simulation. Results from several simulations show that regions of interest are successfully identified in spite of training set class imbalances. Accuracy is analyzed both at the level of nodes in the simulation data structure, and in terms of higher-level regions of interest. It is shown that over 98% of salient regions are found in independent test sets. Hence, this approach will be a significant time saver for simulation users and developers.

1. Introduction

We consider the problem of dealing with training data that is too large to fit in the memory of a typical computer (or compute node), too large to cycle between computers, and that does not have a uniform distribution of classes of data across nodes [5]. Such a problem exists for the United States Department of Energy's Advanced Simulation and Computing program (ASC) [1], wherein a supercomputer simulates real-world events of strategic importance [9]. The simulation data is partitioned and distributed across separate processors, to facilitate parallel computation. Of concern for learning is that the storage allocation optimizes for efficient computation of the simulation, without regard to conditions that might make it easy or difficult for a machine learning algorithm to use the resulting data. Physical objects in the simulation may well be distributed across processors such that adjacent regions that would not normally be separable are, in fact, separated.

In analyzing the results of these simulations, developers and users want to spot anomalies that may take days or weeks to find by hand. Developers look for anomalies that arise due to bugs in the simulation. Users look for anomalies that represent important events. Therefore, manually marking some areas of interest and automatically finding others in the same or similar types of simulations can greatly reduce debugging and analysis time.

In this paper, we give examples of learning from several simulations of a storage canister being crushed by an impactor bar from above. We perform separate experiments using vertical and horizontal partitioning of the canister. These two different partitioning schemes are designed to illustrate the problem of an arbitrary data partitioning, precisely because they complicate the use of a learning algorithm. An illustration of the simulation model with vertical partitions appears in Figure 1, where the different shades of gray represent the partitioning of the simulation in a distributed environment. Note that pieces of the impactor bar crushing the canister are also broken up spatially according to the partition. A visualization of the horizontal partitions is shown in Figure 2.

As a result of the partitioning, areas of saliency in some partitions may be limited to only a few points in the simulation mesh (called nodes). Each node typically has a set of physical characteristics associated with it, as well as a mesh location. Salient nodes, being few in number, exhibit a pathological minority-class classification problem. In the case of a partition having zero salient points, the only classifier that can be learned is a trivial one that always predicts unknown.

We show that it is possible to obtain an accurate prediction of salient points even when the data is partitioned arbitrarily in 3D space with no particular relation to feature space. Results on the canister data sets indicate that experts working with much larger simulations can benefit from the predictive guidance obtained from only a small amount of relevant data. As both designers and simulation users are most interested in finding a salient region rather than individual salient nodes, we also evaluate how well our approach can detect connected groups of salient nodes.

2 Data description

In the canister-crush simulation, an impactor bar crushes a canister from above. The walls of the canister buckle under the pressure and the top of the canister travels downward until it meets the bottom. Typical simulations have from 25 and 44 timesteps depending on the impactor speed and simulation completion.

2.1 Physical and spatial characteristics

In the four different instances of the EXODUS II format [10] can-crush simulation, several physical variables are stored for each node within each time step. These variables are the displacement on the X, Y, and Z axes; velocity on the X, Y, and Z axes; and in canister simulation 1 only, acceleration on the X, Y, and Z axes. An ”Equivalent Plastic Strain” variable, which is a metric for the stress on the surface of the canister [8], is also stored for each finite element of eight nodes in each time step. This variable is not used in training or testing, though it contributed as a template in labeling the ground truth. The nodes and finite elements of the simulation model are embedded in a mesh framework. Table 1 shows the parameter settings for each simulation. Table 2 shows the ranges of the features in each simulation.

Figure 3 shows a visualization of ground truth data in the final time step of each simulation. Simulation 2 ends before much of the canister has been crushed. In simulation 4 the impactor bar itself is deformed when a user runs the simulation for too long. This could be an example of something interesting to a simulation designer, as it might violate physical constraints.

The data for each time step is divided spatially according to the compute node to which it is assigned. The vertical partitioning is performed parallel to the Y axis of the canister, dividing the canister into four disjoint spatial partitions with 1,640, 1,886, 1,886, and 1,312 nodes per time step. The horizontal partitioning is performed along the Z axis of the canister, dividing the canister into four disjoint spatial partitions with 1,640, 1,640, 1,640, and 1,804 nodes per time step. Data from the impactor bar is not used for training or testing in either horizontal or vertical experiments. This represents the focus of the simulation designers on the integrity of the storage container.

2.2 Train and test sets

To create labeled training data for every time step, those pieces of the canister that have buckled and been crushed are manually marked as salient, using a custom add-on to ParaView, an open-source visualization tool for scientific data [6]. At the beginning of the simulation, before the impactor bar has made contact, there are no salient nodes within the mesh. As time progresses and the canister collapses, more and more nodes are marked salient. Every node not marked salient receives the label unknown, rather than not salient, to reflect the fact that, in general, the users will indicate only salient regions.

Designating nodal saliency by means of an expert can in principle be as precise as desired, but more precision requires greater effort. There are 6,724 canister nodes in each simulation model, each of which may be labeled differently in each time step. The domain expert will generally want to mark regions in a manner that saves time rather than catering to the nuances of data mining. Thus we have allowed a fair amount of noise in the class labels by using tools which mark areas rather than individual nodes in the simulation model.

A classifier or an ensemble of classifiers is trained on each of the four partitions of a simulation. Testing on each

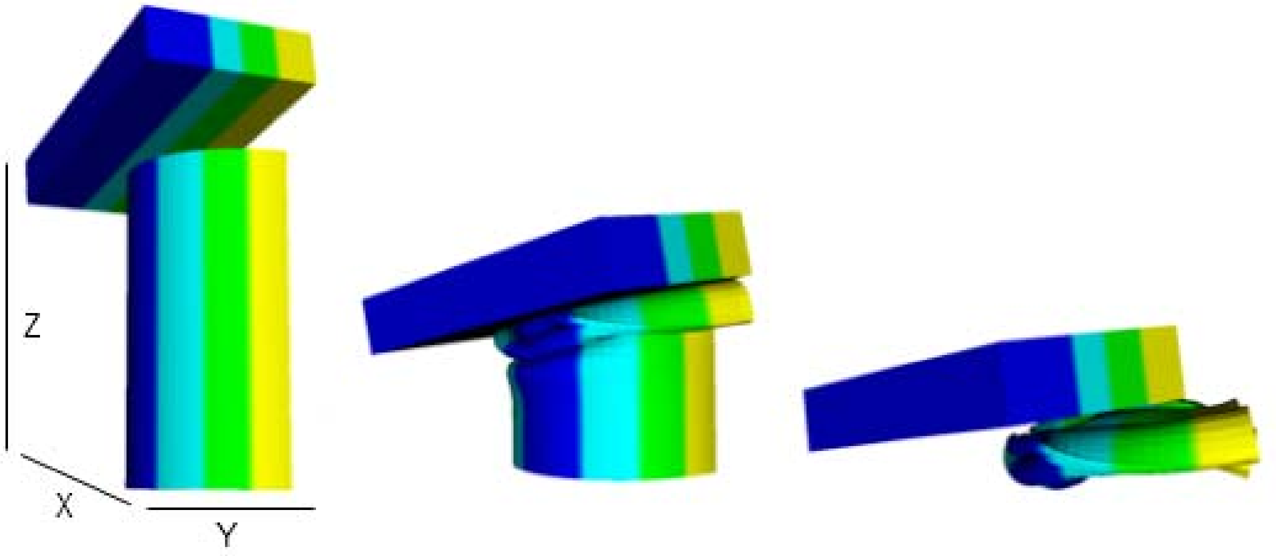

Figure 1. A visualization of the data as distributed across compute nodes for vertical partitions. Four partitions are shown in different gray levels as the storage canister is crushed. Partitions 0 to 3 in numerical order are shown from right to left.

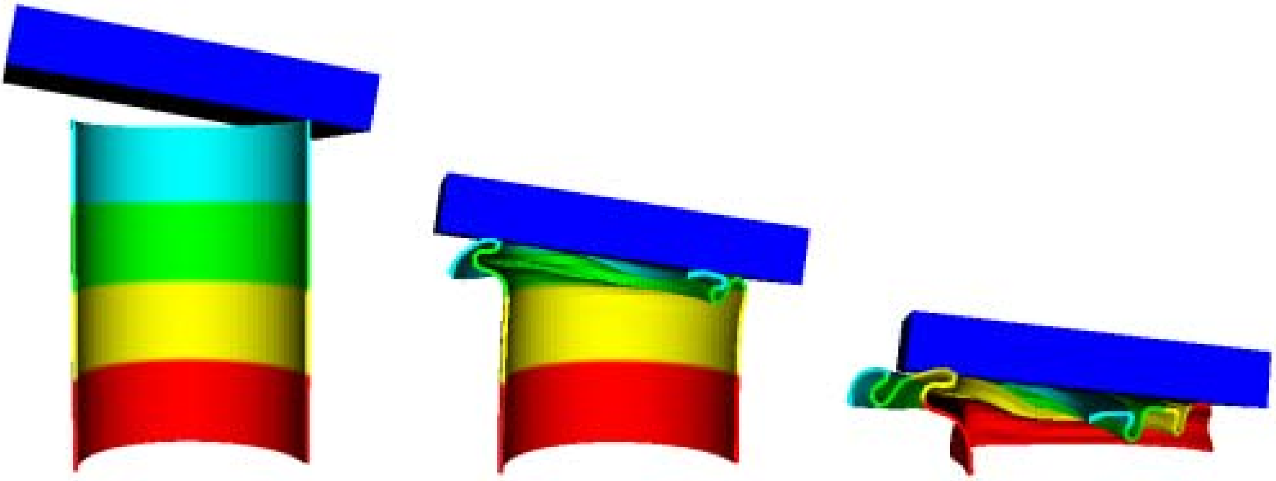

Figure 2. A visualization of the data as distributed across compute nodes for horizontal partitions. Four partitions are shown in different gray levels as the storage canister is crushed. Partitions 0 to 3 in numerical order from top to bottom are beneath the impactor bar in the left view.

Table 1. Physical and spatial characteristics for the canister simulations. Impactor bar velocity is in inches per second. "% salient can nodes" indicates the fraction of the can marked as salient.

Canister simulation	1	2	3	4
Bar initial velocity (in/s)	5,000	2500	5,000	7,500
# nodal variables	9	6	6	6
# can nodes per time step	6,724	6,724	6,724	6,724
# bar nodes per time step	3,364	1,740	1,740	1,740
Total # nodes per time step	10,088	8,464	8,464	8,464
# time steps	44	31	31	25
Total # can nodes	295,856	208,444	208,444	168,100
% salient can nodes	64.7	27.3	51.3	60.7

Table 2. Feature ranges for canister data in simulations 1 to 4. NA denotes not applicable.

Feature	Simulation 1		Simulation 2		Simulation 3		Simulation 4	
	min	max	min	max	min	max	min	max
DISPLX (in)	-7.2	1.4	-4.0	0.5	-4.2	0.4	-4.5	1.0
DISPLY (in)	-5.5	1.5	-1.2	1.5	-0.8	1.6	-1.6	5.9
DISPLZ (in)	-17.8	0.1	-7.0	0.0	-13.2	0.0	-16.1	0.0
VELX (in/s)	-4,820	2,252	-4,529	1,161	-4,562	2,138	-30,840	14,385
VELY (in/s)	-7,891	3,357	-1,327	2,541	-2,113	3,616	-15,703	59,456
VELZ (in/s)	-8,862	3,287	-4,837	493	-8,226	998	-15,980	3,986
ACCLX (in/s^2)	-1.75E+09	2.39E+09	NA	NA	NA	NA	NA	NA
ACCLY (in/s^2)	-2.47E+09	3.38E+09	NA	NA	NA	NA	NA	NA
ACCLZ (in/s^2)	-3.99E+09	3.02E+09	NA	NA	NA	NA	NA	NA

of the other simulations is performed using a probabilistic combination (details to follow) of the votes from the four ensembles. Therefore each test example is classified by using classifiers trained on examples from another simulation.

3 Classification system

For the case of a classifier or ensemble created from the training data at each compute node, we compare different learning algorithms capable of building a predictive model from large amounts of data in a timely fashion. First, a single pruned decision tree is created from data within each compute node to establish a baseline. Then we use Breiman's random forest algorithm [4] with 250 trees per partition and unweighted predictions, which produces a single class vote for the forest. The motivation for using random forests stems from the inherent speed benefit of choosing a split from only a few randomly selected attributes at each branch in the tree. A complete description of the random forest algorithm can be found in [4]. Its accuracy was evaluated in [2] and shown to be comparable with or better than other ensemble generation techniques.

Classification of a test point within the simulation involves prediction by each partition's ensemble. Because our algorithms need to work when only a few compute nodes have salient examples, a simple majority vote algorithm may fail to classify any points as salient if the number of compute nodes trained with salient examples is less than half of the number of compute nodes. In a large-scale simulation it is likely that there will be nodes which have no salient examples in a training set. Therefore we must consider the priors: the probability that any given node contained salient examples during training and therefore is capable of producing a classifier that can predict an example as salient. A description of this algorithm [3] is as follows:
$p(w_1|x)$ = percentage of ensembles voting for class w_1 for example x
$P(w_1)$ = percentage of ensembles capable of predicting class w_1
Classify as w_1 if: $\frac{p(w_1|x)}{P(w_1)} > \frac{p(w_2|x)}{P(w_2)}$
Classify as w_2 if: $\frac{p(w_1|x)}{P(w_1)} < \frac{p(w_2|x)}{P(w_2)}$

Thus, a probabilistic majority vote is applied to a two-class problem. An n-class problem as addressed in [3] can be dealt with as follows:
Classify as w_n : $argmax_n(\frac{p(w_n|x)}{P(w_n)})$

In the case of a tie vote, the unknown class is predicted,

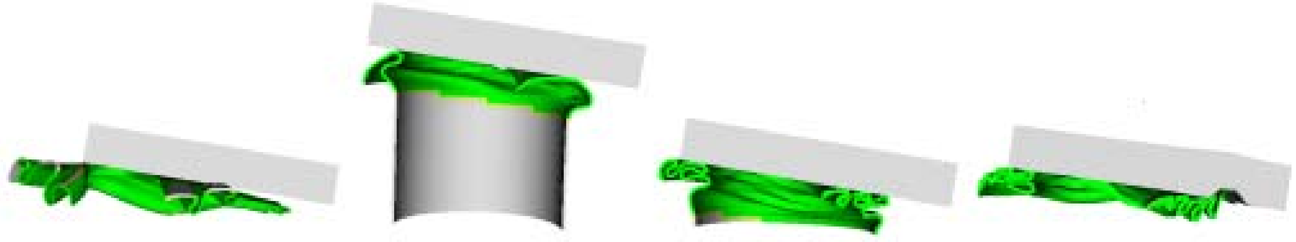

Figure 3. Final time step in simulations 1, 2, 3, and 4 (left to right). Ground truth salient regions are darker than unknown regions.

since a definite salient vote has not been determined. We are interested in directing people to salient regions, that is, connected groups of salient points. Missing a few points within a region due to tie votes for those particular points is unlikely to be important for region detection.

4 Experiments

Training is performed on the data contained in each of the four partitions of each of the four simulations 1, 2, 3, and 4 to create both a 250-tree random forest ensemble and a single pruned decision tree for each partition. The decision tree classifier or the random forest ensemble of each partition in a training simulation returns a single prediction for the test example of a test simulation. The four predictions from those classifiers or ensembles are combined into a single prediction for the example in the separate test simulation using a probabilistic majority vote. Predictions on the test examples are compared to the marked saliency of the test examples to determine accuracy.

The salient regions of the data are marked using region-based tools of the ParaView application [6]. The ensembles of classifiers used to classify the test data often produce smaller salient clusters of nodes or even individual isolated salient nodes, which do not correspond well to the larger marked, ground truth regions. In order to improve the regional accuracy of these ensembles, we employ some of the regional tools in the Feature Characterization Library (FCLib-1.2.0) toolkit [7] to process the ensemble prediction data. The numerical class label (0.5 for unknown, 1.0 for salient) of all nodes within a physical radius of two inches of each node is averaged in a smoothing operation. After smoothing, nodes have saliencies in the range from [0.5,1]. A threshold of 0.75 is used to label the nodes as salient, as that is midway between the 0.5 and 1.0 used to signify “Unknown” and “Salient” in the training data. Regions are created of connected components of salient nodes after thresholding. Smoothing tends to remove the smaller salient regions and isolated salient nodes. Another tool is used to generate overlap matrices of ground truth and predicted regions.

5 Results

Results are separated into nodal results and regional results. Please note that results labeled decision tree (”DT” in Table 3) are from building one decision tree per partition. However, when used to classify an example, the decision trees built from the different training partitions form a mini-ensemble.

5.1 Nodal results

Table 3 shows the overall accuracies for cross simulation experiments. Since the final vote is computed using 1 vote from each of the 4 partitions (if each partition has examples of both classes), ties may exist in this 2-class experiment. Ties are assigned to the unknown class. Nodal accuracies are calculated by designating each nodal example as one of the following: true negative, true positive, false negative, or false positive, depending on the ground truth and predicted class of the example. Overall nodal accuracies are calculated by dividing the total sum of true negatives and true positives by the total sum of examples. The highest overall accuracy is generally obtained with random forests for a given train/test pairing. However, the algorithms are often close in accuracy and there is not a consistently best approach.

5.2 Regional results

The goal of the prediction stage is to direct experts to additional salient regions. Assessing the accuracy of an algorithm in finding and classifying regions is more difficult than determining the above node-level accuracy results. We compute a quantitative measure of region detection accuracy. Our regional accuracies designate each regional example as true positive, false negative, or false positive. Only

Table 3. Cross simulation accuracies for canister simulations 1, 2, 3, and 4. DT denotes decision tree. RF denotes random forest.

Ensemble	Simulation		Four vertical partitions Accuracy (%)			Four horizontal partitions Accuracy (%)		
	Train	Test	Nodal	Regional		Nodal	Regional	
				Unsmoothed	Smoothed		Unsmoothed	Smoothed
DT	1	2	94.5	77.5	96.8	94.0	75.6	96.8
RF	1	2	95.7	86.1	96.8	95.2	96.9	96.8
DT	1	3	95.4	50.8	96.9	91.0	53.4	96.8
RF	1	3	96.6	68.9	96.8	96.0	79.5	100.0
DT	1	4	95.1	64.1	100.0	89.8	59.5	92.3
RF	1	4	95.7	83.3	96.0	96.4	92.6	100.0
DT	2	1	88.5	31.5	96.5	84.0	44.8	88.7
RF	2	1	88.2	46.7	98.2	84.8	51.4	96.5
DT	2	3	92.3	35.2	88.2	88.2	35.2	93.9
RF	2	3	93.2	52.5	96.8	93.3	54.4	96.8
DT	2	4	90.1	50.0	96.2	90.0	25.0	85.7
RF	2	4	93.9	49.0	96.2	95.2	37.9	96.2
DT	3	1	85.3	65.9	98.2	92.6	38.6	100.0
RF	3	1	87.2	76.7	98.2	92.4	59.6	100.0
DT	3	2	95.4	56.4	93.8	95.6	32.3	96.8
RF	3	2	96.1	91.2	96.8	96.2	60.8	93.8
DT	3	4	96.1	71.4	100.0	96.6	35.7	96.2
RF	3	4	97.7	73.5	100.0	96.9	64.1	100.0
DT	4	1	92.5	42.7	94.9	91.6	50.5	87.5
RF	4	1	91.8	70.0	96.6	90.2	47.9	98.2
DT	4	2	95.6	86.1	96.8	83.2	39.7	85.7
RF	4	2	95.9	91.2	93.8	81.1	62.0	96.8
DT	4	3	94.6	52.5	96.9	90.3	48.4	96.9
RF	4	3	95.1	79.5	100.0	89.3	70.5	100.0

the first time step has no salient regions. Hence, for that time step one could say that there is a true negative, if no regions are predicted salient. It is possible that more than one predicted true positive region will intersect with a labeled true positive region. We count this as a single discovery of the ground truth true positive region. For the purposes of people searching for interesting events, this appears sensible because they would be directed to the region. False positives are counted for each region that was predicted salient that does not intersect a ground truth salient region.

This may result in more total predictions than actual salient regions. Overall regional accuracy is calculated as the total sum of true negatives and true positives divided by the total number of all four designations. This approach does not consider the actual node intersection percentage of predicted and ground truth salient regions. It also does not penalize multiple predicted regions that intersect one ground truth salient region. False negatives could each be counted some multiple of times in order to weigh these errors more than false positives. A performance metric that takes all these issues into account will be developed as part of future work. The regional accuracies (after the FCLib processing) are shown in Table 3.

Salient regions are always detected without smoothing. Smoothing removes the smaller but correct single salient region in the second time step in 50% of the experiments, which results in one false negative per simulation in those cases. The smoothing operations increase regional accuracies by reducing the number of false positive regions (predicted regions not connected to ground truth) as the radius is increased. The often much lower unsmoothed regional accuracies result from false positive regions, which often consist of one or several nodes. Smaller regions that are predicted salient are either removed or consolidated into larger regions. Figure 4 shows an example of smoothing applied to a single time step of simulation 1 as predicted by the ensembles of random forests that were trained on data of simulation 4 partitions. The leftmost image shows ground truth. The middle image shows some of the eleven false positive regions in this time step without smoothing. The rightmost image is after smoothing with a radius of 2 inches and contains one false positive region (not visible).

While random forests more often have a higher unsmoothed regional accuracy than decision tree ensembles, smoothing in vertical partition experiments removes this advantage. In general, with a smoothing radius of 2, over 98% of the salient regions are correctly identified with either decision tree or random forests ensembles.

6 Summary and discussion

Some simulations must be partitioned across multiple processors in order to obtain results in a reasonable amount of time. The method of breaking data into pieces is not designed with data mining in mind, as it violates the usual assumption of independent and identically distributed data sets. In this paper we show how such data may be nonetheless effectively used for data mining. Our approach uses fast ensemble learning algorithms and probabilistic majority voting.

Our results from several simulations indicate that our approach has the ability to find most nodes in regions of interest. In our experiments using the data from several different runs of the canister crush simulation, the resultant predictions appear more accurate (in terms of matching the physical processes in the simulation) than the training data, which has been labeled approximately in accordance with the time constraints placed upon experts. This provides confidence that the algorithm is learning the underlying function which determines which points are salient, with the overlap of uninteresting points outweighing the very large number of uninteresting points overall.

We evaluate how well regions of salience are found. After smoothing the results of random forests prediction, there were at most one false negative and/or two false positive regions per test simulation. Overall 98% of the salient regions are correctly identified. So, this is a promising result in terms of the utility of the approach. The results indicate that simulation developers and users would be accurately directed to regions of interest with only occasional misdirection. This has the potential for saving significant time during debugging and use by allowing for a much improved focus of attention on areas of interest without highly time-consuming search.

We believe the rapid generation of ensemble classifiers will make it tractable to predict saliency in much larger data sets. The general problem of creating an ensemble from data that was partitioned without regard to the effect on the machine learning algorithm is an important practical problem that merits additional attention.

7 Acknowledgments

This research was partially supported by the Department of Energy through the Advanced Simulation and Computing program (ASC) Pre- and Post-Processing Environments (PPPE) Data Discovery Program Contract number: DE-AC04-76DO00789.

References

[1] ASC, National Nuclear Security Administration in collaboration with Sandia, Lawrence Livermore, and Los Alamos National Laboratories, http://www.sandia.gov/nnsa/asc/.

[2] R. E. Banfield, L. Hall, K. Bowyer, D. Bhadoria, W. Kegelmeyer, and S. Eschrich. A comparison of ensemble creation techniques. In *The Fifth International Conference on Multiple Classifier Systems, Cagliari, Italy*, pages 223–232, 2004.

[3] R. E. Banfield, L. Hall, K. Bowyer, and W. Kegelmeyer. Ensembles of classifiers from spatially disjoint data. In *Sixth International Workshop on Multiple Classifier Systems*, pages 196–205, 2005.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] L. Hall, D. Bhadoria, and K. Bowyer. Learning a model from spatially disjoint data. In *2004 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2*, pages 1447–1451, October 2004.

[6] A. Henderson. *The ParaView Guide*. Kitware, Inc., United States, 2004.

[7] W. S. Koegler and W. P. Kegelmeyer. FCLib: A library for building data analysis and data discovery tools. *Advances in Intelligent Data Analysis VI*, IDA 2005:192–203, 2005.

[8] B. S. Lee, R. R. Snapp, and R. Musick. Toward a query language on simulation mesh data: an object oriented approach. In *Proceedings of the International Conference on Database Systems for Advanced Applications*, pages 533–536, 2001.

[9] M. Pilch. Science of prediction, DOE CSGF yearly report. Technical report, Krell Institute, www.krellinst.org/csgf/deixis/2004/research.cgi?id=201, 2004–2005.

[10] L. A. Schoof and V. R. Yarberry. EXODUS II: A Finite Element Data Model. Technical report, Sandia National Labs, Albuquerque, NM 87185, 1998.

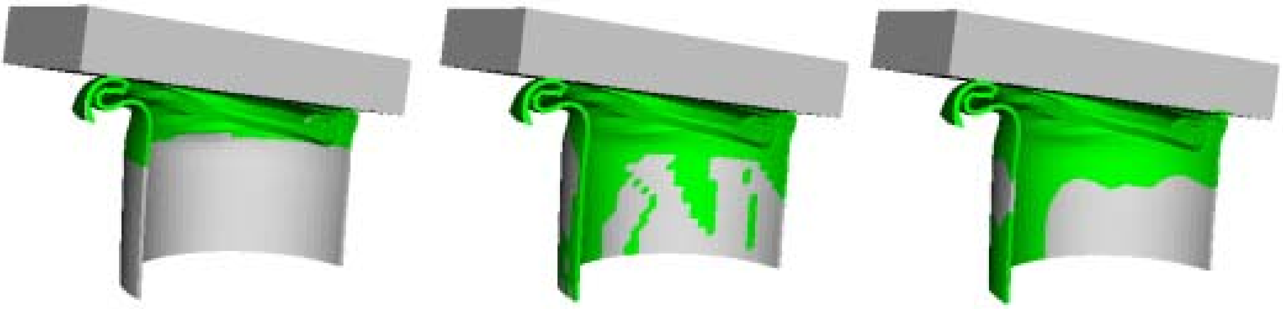

Figure 4. Left: Ground truth as labeled in time step 18 of Simulation 4. Center: Predicted salient regions including false positives (smaller regions) before smoothing. Right: Salient regions after smoothing with one false positive (not visible).