Efficient segmentation of spatio-temporal data from simulations

Imola K. Fodor and Chandrika Kamath

Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA, USA 94551

ABSTRACT

Detecting and tracking objects in spatio-temporal datasets is an active research area with applications in many domains. A common approach is to segment the 2D frames in order to separate the objects of interest from the background, then estimate the motion of the objects and track them over time. Most existing algorithms assume that the objects to be tracked are rigid. In many scientific simulations, however, the objects of interest evolve over time and thus pose additional challenges for the segmentation and tracking tasks. We investigate efficient segmentation methods in the context of scientific simulation data. Instead of segmenting each frame separately, we propose an incremental approach which incorporates the segmentation result from the previous time frame when segmenting the data at the current time frame. We start with the simple K-means method, then we study more complicated segmentation techniques based on Markov random fields. We compare the incremental methods to the corresponding sequential ones both in terms of the quality of the results, as well as computational complexity.

Keywords: Image segmentation, K-means, Markov random field, simulation data

1. INTRODUCTION

Detecting and tracking objects in spatio-temporal datasets is an active research area with applications in many domains. Usually, the first step in such problems is to detect the objects in the data by segmenting the frames corresponding to the different time steps, and then estimating the motion of the objects from frame to frame.

In this paper, we investigate efficient segmentation methods in the context of scientific simulation data. Instead of segmenting each frame separately, we propose an incremental approach which incorporates the segmentation result from the previous time frame when segmenting the data at the current time frame. We start with the simple K-means method, then we study more complicated segmentation techniques based on Markov random fields. We compare the incremental methods to the corresponding sequential ones both in terms of the quality of the results, as well as computational complexity.

The rest of this paper is organized as follows. Section 2 describes the simulation dataset we use as an example in this study. Section 3 outlines methods we consider: the K-means and the Markov random field segmentation techniques in Sections 3.1 and 3.2, respectively, and explains our incremental approach in Section 3.3. Section 4 presents our results, and finally, Section 5 concludes with a summary.

2. THE DATA

Understanding turbulence is an important scientific problem, but one that has not yet been fully solved. To answer the remaining challenges, it is important to analyze carefully the results of different turbulent flow simulations. We consider 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.[1] Initially, 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.

In this paper, we illustrate our methods using only a small subset of the data, as shown in Fig. 1. The twelve

Further author information: (Send correspondence to I.K.F.)

I.K.F.: E-mail: fodor1@llnl.gov, Telephone: 1 925 424 5420, Address: Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA, USA 94551

C.K.: E-mail: kamath2@llnl.gov, Telephone: 1 925 423 3768, Address: Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA, USA 94551

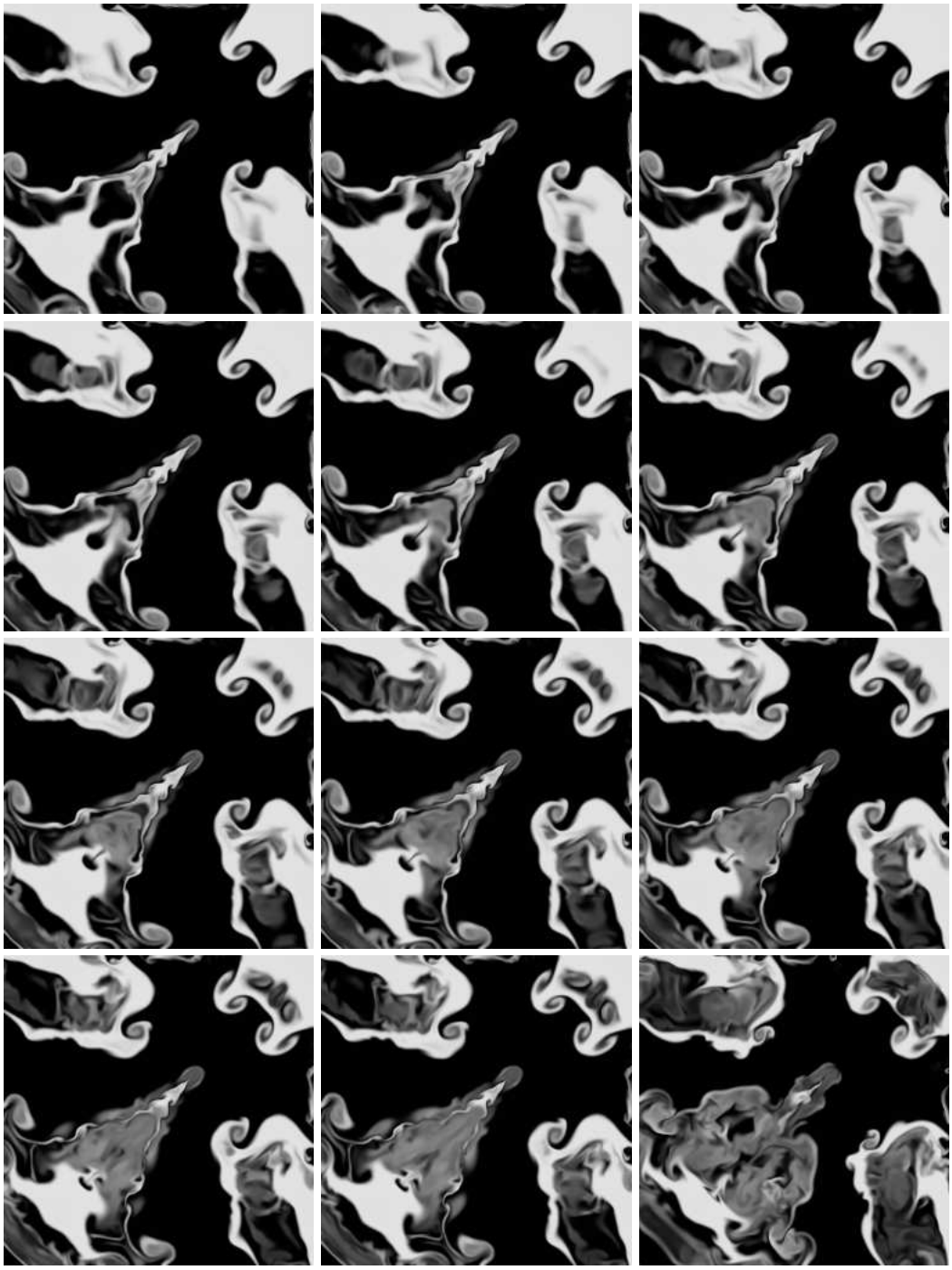


Figure 1. Example turbulence images, each of size 256×256 pixels, corresponding to different time steps in the simulation. The time evolution is ordered by row. The first eleven correspond to consecutive time steps, while the last one is ten time steps after the eleventh time frame.

images are each of size 256×256 pixels, and correspond to twelve time steps taken at a certain cross-section of a typical $256 \times 256 \times 128$ data cube from a node. The time evolution is by row, that is, the three images in the top row correspond to time steps 1 through 3, the three images in the second row to time steps 4 through 6, and so on. The first eleven images correspond to consecutive time steps, while the twelfth one is ten time steps after the eleventh.

3. THE METHODS

There are numerous techniques for segmenting images. A frequently updated extensive computer vision bibliography available online at `http://iris.usc/edu/Vision-Notes/bibliography/contents.html` lists several examples, including global threshold based methods, region growing algorithms, split and merge algorithms, model based techniques, deformable curves (snakes), level set methods, and texture based implementations.

In rest of this section, we describe the segmentation algorithms we consider. Section 3.1 outlines the K-means clustering algorithm as used for segmentation, Section 3.2 explains segmentation based on Markov random fields, and Section 3.3 presents our incremental approach.

3.1. Segmentation using K-means

As a first segmentation method, we apply the simple K-means clustering algorithm.[2,3] Given N observations $\{\mathbf{z}_i\}_{i=1}^N$, from a d-dimensional dataset, and a specified number of classes K, the K-means algorithm divides the observations into K clusters by minimizing the total within-class sum of squares. For a given class, the within-class sum of squares is simply the sum of the L_2 distances of the objects in the class and the class mean. If μ_k denotes the p-dimensional mean corresponding to class k, denoted by Cl_k for $k = 1, \ldots, K$, the within-class sum of squares for class k is defined as

$$\mathrm{SSQ}_k = \sum_{\{i:\mathbf{z}_i \in Cl_k\}} (\mathbf{z}_i - \mu_k)^2. \tag{1}$$

The total within-class sum of squares is the sum of the K class sum of squares

$$\mathrm{TSSQ} = \sum_{k=1}^{K} \mathrm{SSQ}_k. \tag{2}$$

In our results, instead of reporting the rather large sums of squares, we provide the class and overall clustering summaries in terms of the number of items per cluster and root mean square (RMS) deviations. For class k, n_k denotes the number of elements in it, and RMS_k the associated class RMS. We have,

$$n_k = \#\{i : \mathbf{z}_i \in Cl_k\}, \tag{3}$$

and

$$\mathrm{RMS}_k = \left(\frac{1}{n_k} \sum_{\{i:\mathbf{z}_i \in Cl_k\}} (\mathbf{z}_i - \mu_k)^2 \right)^{1/2}. \tag{4}$$

For the overall clustering, the total number of elements N is

$$N = \sum_{k=1}^{K} n_k, \tag{5}$$

and the total RMS is

$$\mathrm{RMS} = \left(\frac{1}{N} \sum_{k=1}^{K} \sum_{\{i:\mathbf{z}_i \in Cl_k\}} (\mathbf{z}_i - \mu_k)^2 \right)^{1/2} = \left(\sum_{k=1}^{K} \frac{n_k}{N} \mathrm{RMS}_k^2 \right)^{1/2}. \tag{6}$$

The K-means algorithm is easily extensible to other cluster centers and distance measures: instead of using the mean and Euclidean distance, one could also use the median and L_1 norm, for example.

In our application, at each time frame, we have $N = n \times m = 256 \times 256 = 65{,}536$ pixel intensity values as the N observations from a $d = 1$ dimensional data.

3.2. Segmentation using Markov random fields

One popular model-based image segmentation method is segmentation with Markov random fields (MRFs). Following Ref. 4, let x_{ij} denote the intensity of the (i,j)th pixel of an $n \times m$ image X, and $y_{ij} \in \{1, ...K\}$ its corresponding class label for $\{(i,j) : i = 1, \ldots, n; j = 1, \ldots, m\}$. The MRF model assumes that the observed intensities depend only on the unobservable (hidden) class labels, and that the distribution of the labels follows an MRF of the form

$$P(Y) \propto e^{\beta V(Y)}, \tag{7}$$

where β is a smoothness parameter and $V(Y)$ is an overall potential function. In Eq. (7), we omitted the normalizing constant that ensures that $\sum_Y P(Y) = 1$. Large values of β correspond to large objects in the data, while small values allow for smaller structures. If the image is homogeneous, it is appropriate to consider one global β for the entire image. However, if there are several regions with spatial structures on different scales, it is recommended to consider spatially varying, or inhomogeneous, MRFs with β_{ij} varying at the different pixel locations.

In this paper, we consider a second order neighborhood model with a location dependent smoothness parameter β_{ij}. The local potential function V_{ij} is equal to $n_{ij}(y_{ij})$, which is the number of pixels labeled y_{ij} in the second order neighborhood $\delta_{ij} = \{y_{kl} : i-1 \leq k \leq i+1, j-1 \leq l \leq j+1\}$ of pixel (i,j). Accordingly, Eq. (7) becomes

$$P(Y) \propto e^{\sum_{i,j} \beta_{ij} n_{ij}(y_{ij})}. \tag{8}$$

We also assume that given the class label y_{ij} at location (i,j), the intensity x_{ij} is a realization from a normal random variable that depends only on the class label y_{ij}, with mean $\mu_{ij} = \mu_{y_{ij}}$ and standard deviation $\sigma_{ij} = \sigma_{y_{ij}}$,

$$P(x_{ij}|y_{ij}) \sim \mathcal{N}(\mu_{ij} = \mu_{y_{ij}}, \sigma_{ij} = \sigma_{y_{ij}}). \tag{9}$$

Combining Eqs. (8) and (9) and using Bayes' theorem, we obtain the posterior distribution of the labels given the intensity observations as

$$P(Y|X) \propto P(Y)P(X|Y) = e^{\sum_{i,j} \beta_{ij} n_{ij}(y_{ij})} \prod_{ij} \frac{1}{\sqrt{2\pi\sigma^2_{y_{ij}}}} e^{-1/(2\sigma^2_{y_{ij}})(x_{ij}-\mu_{y_{ij}})^2}. \tag{10}$$

We estimate the MRF parameters β_{ij} with the method developed in Ref. 4, based on the pseudo likelihood approximation of the maximum likelihood solutions. The spatially homogeneous model corresponds to one global beta at each image pixel, that is, $\beta_{ij} \equiv \beta$. The technique requires an initial guess estimate of the labels. Rather than a random guess, we use the results of a K-means segmentation as the starting point of the algorithm.

3.3. An incremental approach to segmentation

Instead of segmenting anew the images at every time step, we propose to use the segmentation result from an earlier time step as the starting point for segmenting the data at a later time frame.

Given the segmentation result at time t_1, we obtain the segmentation at time $t_2 > t_1$ as follows: first, we take the pixel-wise difference of the image X_{t_2} at time t_2 and X_{t_1} at time t_1, to obtain a difference image D_{t_1,t_2}. Next, for $i, j = 1, \ldots, 256$, if the absolute value of the difference image at pixel (i,j), i.e., $D_{t_1,t_2}(i,j)$, is less than a positive threshold λ, we keep the class of $X_{t_2}(i,j)$ the same as the class of pixel $X_{t_1}(i,j)$. Otherwise, we assign pixel $X_{t_2}(i,j)$ to the class k that minimizes $|(X_{t_2}(i,j) - \mu_k)/RMS_k|$ over $k = 1, \ldots, K$.

4. RESULTS

The three images in Fig. 2 display the segmentation results for the first time frame obtained by K-means with $K = 3$, 4, and 5, respectively. The corresponding class summaries are presented in Table 1.

The top row of Fig. 3 and Table 2 display the corresponding segmentation results for the second time frame obtained by K-means with $K = 3$, 4, and 5, respectively. The three images in the middle row of Fig. 3 show the

(a) (b) (c)

Figure 2. Results of the K-mean segmentation algorithm applied to the first time frame in Fig. 1. (a) K=3, (b) K=4, (c) K=5.

Table 1. Class summaries for the three K-means segmentations displayed in Fig. 2.

	RMS	k	μ_k	n_k	RMS_k
K=3	15.02	1	7.16	40,858	10.24
		2	117.72	6,362	31.28
		3	219.85	18,316	15.28
K=4	10.37	1	5.54	39,409	5.66
		2	74.55	4,222	20.90
		3	151.63	4,912	20.99
		4	223.0	16,993	10.22
K=5	8.15	1	4.99	38,719	3.99
		2	58.61	3,706	16.61
		3	120.77	3,386	16.90
		4	174.61	3,723	15.47
		5	224.90	16,002	7.33

incremental segmentation result of the second time frame, based on the K-means segmentation of the first time frame and the difference information, as explained in Sect. 3.3. For a given K, the value of the threshold λ for each of the images Fig. 3(d-f) is chosen as the largest class RMS_k corresponding to the K-means segmentation of the first frame. A pixel in the second image is re-classified only if its difference, in absolute value, from the corresponding pixel in the first frame exceeds the largest class RMS_k. The differences between the corresponding K-means segmentations of the second frame, displayed in the top row of Fig. 3, and the ones obtained by the incremental shortcut, shown in the middle row Fig. 3, are presented in the third row of Fig. 3.

In our comparisons, we consider the K-means segmentations performed at each time step to be the ground truth for the corresponding time frame. We evaluate the accuracy of the incremental segmentation only in comparison with the K-means result. Alternative measures,[5] weighting objective errors, that is, differences between ground truth and segmented image, by human perception could also be used.

Table 3 presents the influence of the thresholds on the accuracy of the incremental results, for select values of the threshold and different time separations between the segmented frames. The thresholds were selected based on the class RMS values in Table 1. The $N_{\text{rc}}(a, b)$ values denote the number of pixels that needed to be re-classified in the incremental segmentation of the image at $t_2 = b$ based on the K-means segmentation of the image at time $t_1 = a$. Essentially, it is the number of pixels where the absolute value of the difference between the image at $t_2 = b$ and at $t_1 = b$ exceeds the threshold λ. The $N_{\text{mc}}(a, b)$ columns indicate the number of pixels that are mis-classified in the incremental segmentation of the image at time at $t_2 = b$ based on the K-means

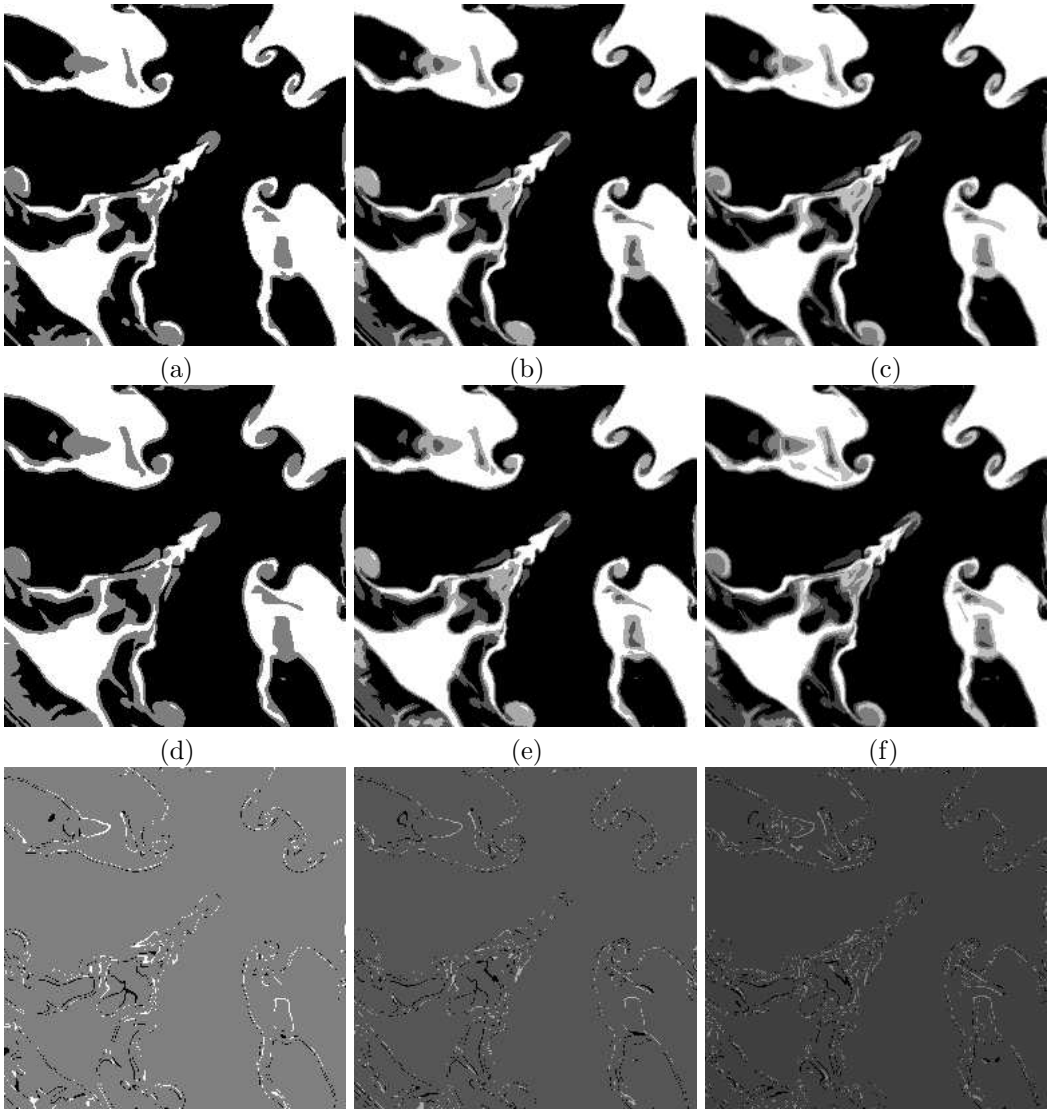


Figure 3. Top row: Results of the K-mean segmentation algorithm applied to the second time frame in Fig. 1. (a) K=3, (b) K=4, (c) K=5. Middle row: Results of the incremental K-means segmentation algorithm on the second time frame ($t_2 = 2$) in Fig. 1, using the corresponding segmentation of the first time frame ($t_1 = 1$) as starting point. (d) K=3, (e) K=4, (f) K=5. Bottom row: Differences between K-means segmentations in top row and corresponding incremental segmentations in middle row. (g) K=3, (h) K=4, (i) K=5.

Table 2. Class summaries for the three K-means segmentations displayed in Fig. 3.

	RMS	k	μ_k	n_k	RMS_k
K=3	15.39	1	7.31	40,613	10.20
		2	115.89	7,037	30.96
		3	219.26	17,886	15.96
K=4	10.77	1	5.80	39,263	6.19
		2	76.86	4,669	21.19
		3	151.66	5,199	21.01
		4	223.02	16,405	10.22
K=5	8.42	1	5.08	38,341	4.12
		2	56.57	3,857	16.21
		3	115.29	3,674	16.52
		4	170.64	4,058	16.076
		5	224.57	15,606	7.73

Table 3. Influence of the thresholds on the accuracy of the incremental segmentation results.

	λ	$N_{\text{rc}}(1,2)$	$N_{\text{mc}}(1,2)$	$N_{\text{rc}}(1,3)$	$N_{\text{mc}}(1,3)$	$N_{\text{rc}}(1,8)$	$N_{\text{mc}}(1,8)$
K=3	0	65,536	3,303	65,536	3,904	65,536	5,050
	10.24	11,137	2,442	15,969	3,353	26,751	4,793
	15.28	8,762	2,341	13,648	3,187	25,294	4,741
	31.28	4,778	2,020	9,079	2,964	20,705	4,493
	50.00	2,344	2,289	5,960	2,880	16,720	3,531
K=4	0	65,536	3,073	65,536	3,422	65,536	6,095
	5.66	14,626	2,567	18,849	3,141	30,335	5,819
	10.22	11,137	2,133	15,969	2,680	26,751	5,969
	20.90	7,080	1,914	11,897	2,340	23,469	5,508
	20.99	7,066	1,909	9,300	2,336	20,959	5,121
	30.00	4,778	2,305	5,960	4,069	16,720	6,424
K=5	0	65,536	3,527	65,536	2,913	65,536	6,366
	3.99	15,554	3,680	19,579	4,412	29,793	4,616
	7.33	14,626	3,573	18,849	4,350	28,777	8,803
	15.47	9,178	2,678	13,648	2,381	25,294	8,803
	16.61	8,415	2,469	13,276	2,907	24,331	5,251
	16.90	7,719	3,508	12,546	3,559	24,038	8,057

segmentation of the image at time $t_1 = a$, when compared to the direct K-means segmentation of the image at time $t_2 = b$. The smaller the number of incorrectly classified pixels, the better the method is.

As expected, the accuracy of the results degrades as the time separation increases between the frames. As λ increases, the number of incorrectly classified pixels decreases first, then it starts increasing again. In our preliminary experiments with this dataset, choosing λ automatically as the largest class RMS leads to reasonable results.

Similar to Fig. 3, Fig. 4 shows the incremental segmentation results of the third time frame, based on the K-means segmentation of the first time frame. The corresponding results of the incremental segmentation of the eighth time frame based on the K-means segmentation of the first time frame are presented in Fig. 5. In general, the accuracy of the incremental results decreases as the time separation increases.

The proposed method can be applied without significant loss of accuracy, provided the differences between

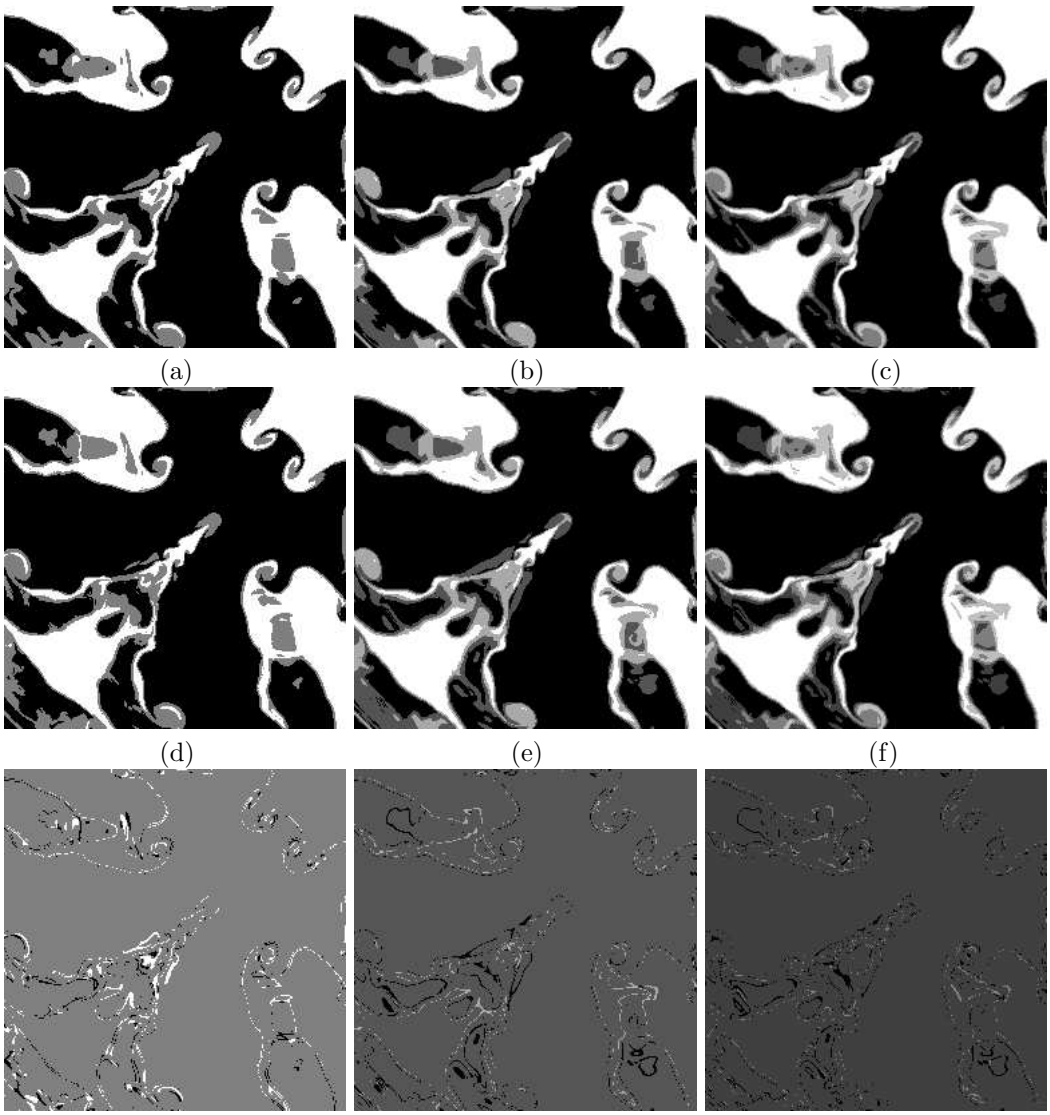


Figure 4. Top row: Results of the K-mean segmentation algorithm applied to the third time frame in Fig. 1. (a) K=3, (b) K=4, (c) K=5. Middle row: Results of the incremental K-means segmentation algorithm on the third time frame ($t_2 = 2$) in Fig. 1, using the corresponding segmentation of the first time frame ($t_1 = 1$) as starting point. (d) K=3, (e) K=4, (f) K=5. Bottom row: Differences between K-means segmentations in top row and corresponding incremental segmentations in middle row. (g) K=3, (h) K=4, (i) K=5.

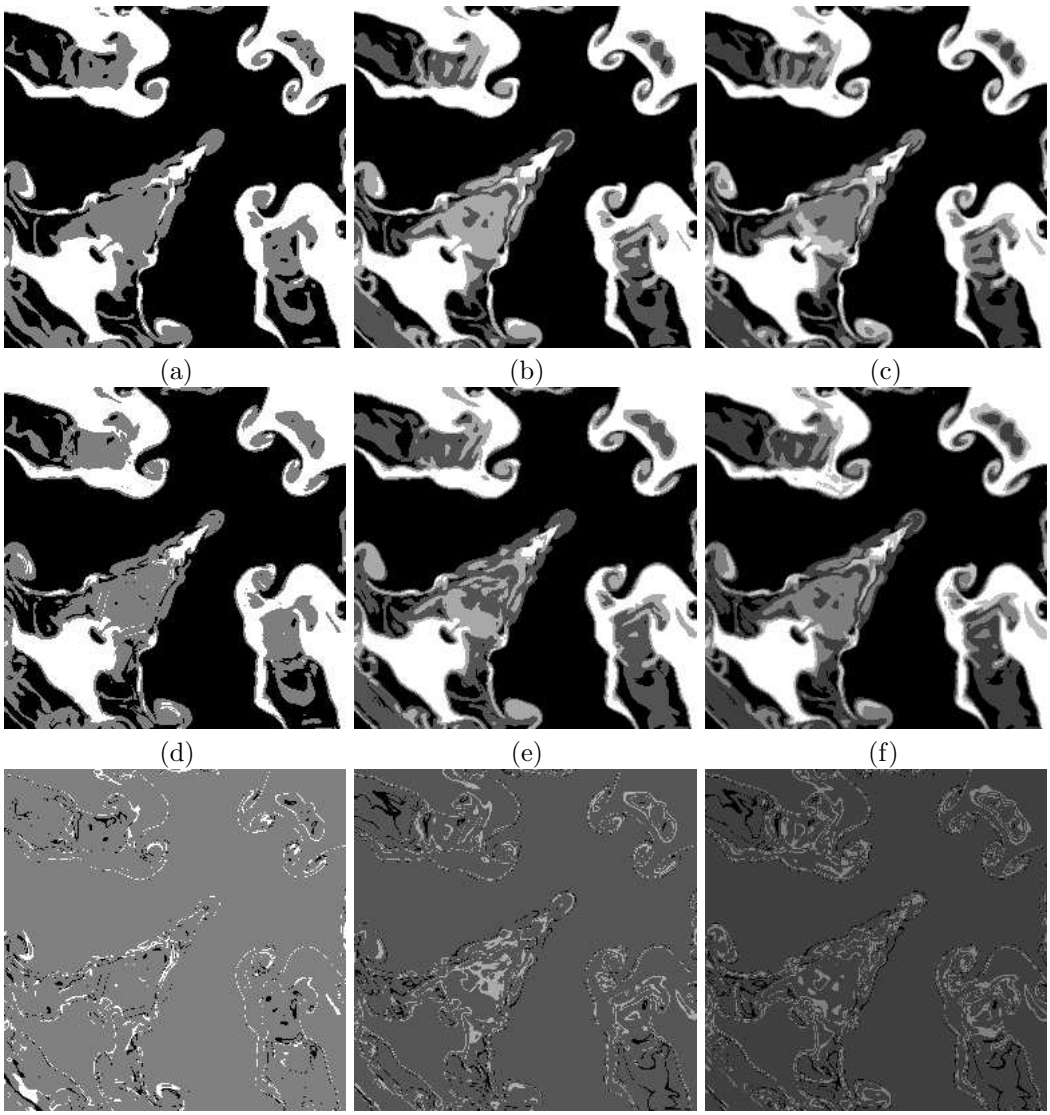
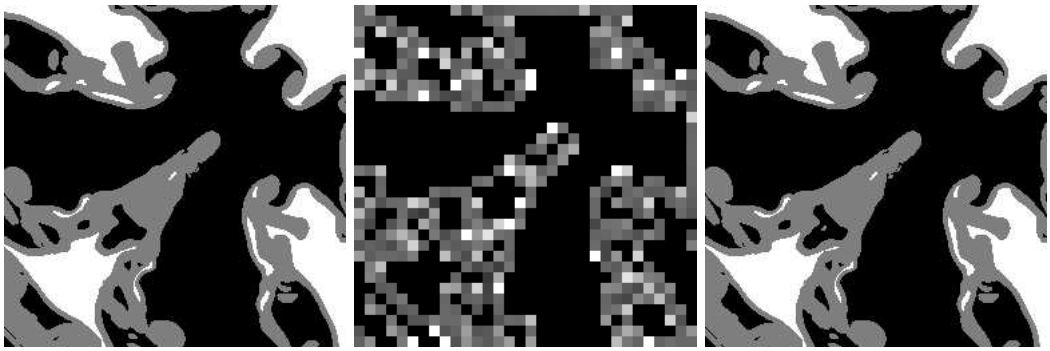


Figure 5. Top row: Results of the K-mean segmentation algorithm applied to the eight time frame in Fig. 1. (a) K=3, (b) K=4, (c) K=5. Middle row: Results of the incremental K-means segmentation algorithm on the eight time frame ($t_2 = 2$) in Fig. 1, using the corresponding segmentation of the first time frame ($t_1 = 1$) as starting point. (d) K=3, (e) K=4, (f) K=5. Bottom row: Differences between K-means segmentations in top row and corresponding incremental segmentations in middle row. (g) K=3, (h) K=4, (i) K=5.



(a) (b) (c)

Figure 6. MRF segmentation for the first time frame with K=3 classes. (a) Result with a global β estimate. (b) Spatially inhomogeneous β_{ij} estimates. (c) Result with the spatially inhomogeneous β_{ij} estimates.

the frames under consideration are small enough. If the differences reach a certain level above the tolerance of the application, the method has to be re-started by performing an initial K-means segmentation on the frame which drifted too much from the first frame.

In terms of computational complexity, one iteration of the K-means algorithm requires calculating the K class centers, as well as the $N \times K$ distances of the N data items from the K centers, and ordering of the distances. The algorithm usually converges in a few iterations. In comparison, the incremental approach requires calculating the N differences between the pixels of the two images, and only $N_{\mathrm{rc}} \times K$ additional distances and sorting, where N_{rc} is the number of pixels where the absolute difference between the two images exceeds a user-defined threshold. With the small images we explored in this study, (using our C++ software on a Dell Precision 530 workstation with 32 GB disk, 512 MB memory, and dual 1.5 GHz Intel Xeon processors running the RedHat Linux operating system), there is no noticeable difference between the run-times of the two different methods. As we scale up the algorithms to cover the full range of the simulation data, we expect the incremental method to be significantly faster than the sequential one.

In addition to the K-means segmentation, we also experimented with the more advanced MRF method. Because of the added spatial information in the MRF model, it is expected to perform better than the non-spatial K-means algorithm. Figure 6 shows the results of the MRF segmentation with K=3 classes for the first time frame. As an initial guess for the MRF, we supplied the result of a K-means segmentation. The first image in Fig. 6(a) illustrates the MRF segmentation using a single global smoothness parameter β. The spatially inhomogeneous β_{ij} estimates are shown in Fig. 6(b), and the corresponding segmentation in Fig. 6(c). In our current work, we are exploring an incremental segmentation technique based on combining the K-means and the MRF methods. Our plan is to use the results of a K-means segmentation on previous time frames as starting points for the MRF segmentation of the image corresponding to the current time frame.

5. SUMMARY

In this paper, we described a simple technique to segment consecutive frames in a time-evolving simulation. Our future plans include investigating more effective segmentation methods in order to better isolate the objects of interest in the data.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-ENG, UCRL-JC-148906. The authors thank Nu Ai Tang for assistance with the data.

REFERENCES

1. 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.
2. J. Hartigan, *Clustering Algorithms*, John Wiley and Sons, New York, 1975.
3. J. Hartigan and M. Wong, "A K-means clustering algorithm," *Applied Statistics* **28**, pp. 100–108, 1979.
4. I. Cadez and P. Smyth, "Modeling of inhomogeneous markov random fields with applications to cloud screening," Tech. Rep. UCI-ICS 98-21, University of California, Irvine, 1998.
5. A. Cavallaro, E. Drelie, and T. Ebrahimi, "Objective evaluation of segmentation quality using spatio-temporal context," *Proc. IEEE International Conference on Image Processing, Rochester, New York*, 22-25 September 2002.