

Online Video Segmentation by Bayesian Split-Merge Clustering

Juho Lee¹, Suha Kwak¹, Bohyung Han^{1,3}, and Seungjin Choi^{1,2,3}

¹ Department of Computer Science and Engineering

² Division of IT Convergence Engineering

³ Department of Creative IT Excellence Engineering,

Pohang University of Science and Technology,

77 Cheongam-ro, Nam-gu, Pohang 790-784, Korea

{stonecold,mercury3,bhhan,seungjin}@postech.ac.kr

Abstract. We present an online video segmentation algorithm based on a novel nonparametric Bayesian clustering method called Bayesian Split-Merge Clustering (BSMC). BSMC can efficiently cluster dynamically changing data through split and merge processes at each time step, where the decision for splitting and merging is made by approximate posterior distributions over partitions with Dirichlet Process (DP) priors. Moreover, BSMC sidesteps the difficult problem of finding the proper number of clusters by virtue of the flexibility of nonparametric Bayesian models. We naturally apply BSMC to online video segmentation, which is composed of three steps—pixel clustering, histogram-based merging and temporal matching. We demonstrate the performance of our algorithm on complex real video sequences compared to other existing methods.

1 Introduction

Clustering is a primitive problem widely used in many computer vision applications. While clustering algorithms have typically been invented for static data, some applications involve dynamic data evolving over time, which often makes the problem much more difficult; clustering results should be consistent in the temporal domain and adaptive to the changes of existing data and the arrivals of new data. Clustering with such constraints is called *evolutionary clustering* [1] and most of existing algorithms are limited to simple extensions of standard clustering techniques by enforcing temporal smoothness [1, 2].

In computer vision, video segmentation is an important example of evolutionary clustering. As a generalization of image segmentation, it aims to cluster the pixels into related groups throughout an input video. However, video segmentation is not straightforward to be handled by ordinary evolutionary clustering techniques because natural videos often involve drastic changes and complex cluster structures. Due to this challenge, many video segmentation algorithms are designed in batch method, which process the entire spatio-temporal video volume offline [3–5]. However, batch processing on the spatio-temporal volume is generally expensive in time and space, and often intractable; the development

of fast and robust online video segmentation algorithm would be essential for the situations with limited resources and real-time requirements.

On the other hand, many video segmentation algorithms suffer from the choice of the proper number of segments as it dynamically changes over time. One possible solution is using nonparametric Bayesian methods such as Dirichlet Process Mixture (DPM) [6] based on the Dirichlet process [7]. There is a prior work to apply the DPM to adapting the number of clusters over time in evolving datasets [8]. For video segmentation, a DPM based algorithm was proposed by extending the static DPM using MCMC for inference [9]. However, both generalizations assume slow evolutions of data. Especially in [9], videos are assumed to be moderately changing and relatively simple because of the limitation of the expensive MCMC steps for inference.

In this paper, we propose an online video segmentation technique based on a novel clustering algorithm called Bayesian Split-Merge Clustering (BSMC). BSMC efficiently organizes clusters through split and merge processes and determines the number of clusters in evolving data, based on the Dirichlet process. It is inspired from Bayesian Hierarchical Clustering (BHC) [10]—a probabilistic version of agglomerative hierarchical clustering. BSMC is a probabilistic version of top-down and bottom-up split-merge clustering, where the initial clustering of the current data is given by the model at the previous time step. The proposed algorithm efficiently handles the temporal variations of data by incremental update of clustering through split and merge operations from the initial clusters at each time step; it maintains structural consistencies in time and adapts to substantial changes from old clusters. Note that BHC is a bottom-up clustering algorithm, which is not easily extended for evolving data. BSMC is nicely applied to the online video segmentation problem and efficiently handles the drastic variations in real-world video sequences with greater accuracy compared to other online segmentation method [9]. The advantages of our video segmentation algorithm are as follows:

- Contrary to many existing algorithms, the proposed algorithm is an online algorithm.
- It performs cluster-wise split-merge inference for clustering in contrast to point-wise inference in DP mixture models; at each time step, it can rapidly adapt to dynamic changes in video, while MCMC methods require many iterations to converge to the solution.
- It sidesteps the difficult problem of finding the proper number of segments by employing flexible nonparametric Bayesian models.

This paper is organized as follows. We first describe general nonparametric Bayesian clustering in Section 2 and discuss BSMC algorithm in Section 3. Section 4 describes the application of our algorithm to video segmentation. Our technique is tested on synthetic data and real video sequences, and its performance is illustrated in Section 5.

2 Nonparametric Bayesian Clustering

2.1 Mixture over Partitions

Clustering on the input dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ is a task to find a mutually-exclusive partition $\{\mathbf{X}_1, \dots, \mathbf{X}_K\}$ of \mathbf{X} , where K can vary from 1 to N . The number of possible partitions is $\mathcal{O}(N^N)$. In nonparametric Bayesian clustering models, each partition of \mathbf{X} is given a probability that measures how well the partition reflects the structure of a dataset. Hence, one can write the marginal probability of \mathbf{X} as a mixture over partitions as

$$p(\mathbf{X}) = \sum_{\phi \in \Phi(\mathbf{X})} p(\mathbf{X}, \phi) = \sum_{\phi \in \Phi(\mathbf{X})} p(\mathbf{X}|\phi)p(\phi), \quad (1)$$

where $\Phi(\mathbf{X})$ is a set of all partitions of \mathbf{X} , and $p(\phi)$ is a prior distribution over partition ϕ . $p(\mathbf{X}|\phi)$ is a likelihood for \mathbf{X} given a partition ϕ , which is given by

$$p(\mathbf{X}|\phi) = \prod_{k=1}^{K_\phi} p(\mathbf{X}_k^\phi), \quad (2)$$

where $\{\mathbf{X}_k^\phi\}_{k=1}^{K_\phi}$ is a set of K_ϕ clusters corresponding to ϕ . Each cluster is characterized by its parameter θ_k , which defines a probabilistic model generating the data that belong to the k th cluster.¹ In non-Bayesian models, we find the optimal parameters for all clusters by point estimation. In Bayesian models, we place a prior distribution over parameters and integrate them out. Therefore, the probability of cluster $p(\mathbf{X}_k^\phi)$ —in other words, the probability that the data in \mathbf{X}_k^ϕ are independently drawn from the same model—is computed as

$$p(\mathbf{X}_k^\phi) = \int \left\{ \prod_{\mathbf{x}_n \in \mathbf{X}_k^\phi} p(\mathbf{x}_n|\theta_k) \right\} p(\theta_k) d\theta_k, \quad (3)$$

which is computed easily provided that $p(\theta_k)$ is a conjugate prior for $p(\mathbf{x}_n|\theta_k)$. Using these probabilities, we compute a score for a partition ϕ by the joint probability $p(\mathbf{X}, \phi)$. As a result, finding the optimal partition of \mathbf{X} reduces to finding the partition with maximum joint probability as

$$\phi^* = \arg \max_{\phi \in \Phi(\mathbf{X})} p(\mathbf{X}, \phi). \quad (4)$$

Note that we do not place any hypothesis on the number of clusters, which means that solving Eq. (4) bypasses the model selection problem. However, finding ϕ^* is often impractical because of the huge search space and the intractable computation of posterior $p(\mathbf{X}|\phi)$. The most popular approach to solve the problem is MCMC sampling, which draws indefinite number of samples from $p(\phi|\mathbf{X})$ and finds reasonable partitions based on the samples.

¹ For example, if the underlying probabilistic model is Gaussian, the parameter would be the mean and covariance of a cluster.

2.2 Prior for Partitions

To define the joint probability $p(\mathbf{X}, \phi)$, we need a prior $p(\phi)$ that is a probability distribution over partitions ϕ . One of choices for the prior is Dirichlet process (DP) [7], which is a random measure on discrete distributions with infinite supports; Dirichlet Process Mixture (DPM) refers to the nonparametric Bayesian models with the DP prior. Under the DP, a random partition of dataset is easily drawn by Chinese restaurant process [6], which is a predictive distribution of DP. Suppose that $\mathbf{x}_{<n} = \{\mathbf{x}_1, \dots, \mathbf{x}_{n-1}\}$ are partitioned into K clusters $\{\mathbf{X}_k\}_{k=1}^K$. Then, for the n th point \mathbf{x}_n ,

$$p(\mathbf{x}_n \in \mathbf{X}_k, 1 \leq k \leq K | \mathbf{x}_{<n}) = \frac{N_k}{n + \alpha - 1} \quad (5)$$

$$p(\mathbf{x}_n \in \mathbf{X}_{K+1} | \mathbf{x}_{<n}) = \frac{\alpha}{n + \alpha - 1}, \quad (6)$$

where $N_k = |\mathbf{X}_k|$. This implies that \mathbf{x}_n may belong to the existing clusters or create a new cluster. Here, α is a *concentration parameter* that controls the tendency to create a new cluster. Using these conditional distributions, the joint distribution of ϕ is given as

$$p(\phi) = \frac{\alpha^{K^\phi} \Gamma(\alpha)}{\Gamma(N + \alpha)} \prod_{k=1}^{K^\phi} \Gamma(N_k^\phi), \quad (7)$$

where Γ denotes the gamma function. Note that this probability is not affected by the ordering of the data, which is referred to as *exchangeability*.

2.3 Bayesian Hierarchical Clustering (BHC)

Instead of drawing indefinite number of samples from $p(\phi | \mathbf{X})$, one can reduce the search space and find the optimal solution by selecting the best among the possible partitions. BHC [10], a probabilistic version of agglomerative hierarchical clustering, reduces the search space using a tree representing the hierarchical structure of the dataset. It computes the *probability* of merging based on the posterior distribution of DPM and merges the pairs whose merging probability is the largest. Unlike traditional hierarchical clustering methods, it automatically determines whether the tree requires additional merging or not by means of the posterior probability. Therefore, it is free from the model selection problem.

More specifically, let \mathbf{X} be a dataset to be clustered. BHC computes $p(\mathbf{X} | T)$, where T is the tree composed of the elements in \mathbf{X} . Instead of summing all possible partitions, BHC sums over the tree-consistent partitions, which are the partitions existing under the tree, representing the hierarchical cluster structure of the dataset. $p(\mathbf{X} | T)$ is computed recursively from the bottom, where each data point corresponds to one node. Let \mathbf{X}_i be a set of data in the subtree rooted by T_i , \mathbf{X}_j be another node in the same level and $\mathbf{X}_k = \mathbf{X}_i \cup \mathbf{X}_j$.

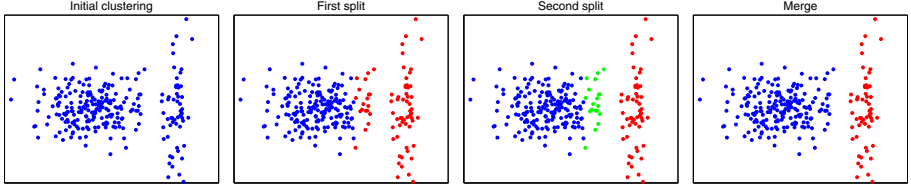


Fig. 1. Example of clustering by BSMC. A single cluster is split and merged through the split and merge stage to perform clustering.

There are two possible options: \mathbf{X}_i and \mathbf{X}_j belong to the one cluster \mathbf{X}_k or they are separate clusters. Therefore, $p(\mathbf{X}_k|T_k)$ is recursively computed as

$$p(\mathbf{X}_k|T_k) = \pi_k p(\mathbf{X}_k|H_k) + (1 - \pi_k) p(\mathbf{X}_i|T_i) p(\mathbf{X}_j|T_j), \quad (8)$$

where H_k is a hypothesis that \mathbf{X}_k is a single cluster and π_k is a prior probability for H_k that is recursively computed from the DP prior. (Note that $p(\mathbf{X}_k|H_k)$ is equivalent to (3).) By the Bayes rule, the posterior probability for H_k is

$$P(H_k|\mathbf{X}_k) = \frac{\pi_k p(\mathbf{X}_k|H_k)}{\pi_k p(\mathbf{X}_k) + (1 - \pi_k) p(\mathbf{X}_i|T_i) p(\mathbf{X}_j|T_j)}, \quad (9)$$

and $p(H_k|\mathbf{X}_k) > 0.5$ means that \mathbf{X}_i and \mathbf{X}_j should be merged. Therefore, the algorithm can determine the stopping level naturally while greedily merging the pair with the largest posterior probability in Eq. (9) at each iteration.

3 Bayesian Split-Merge Clustering (BSMC)

BHC is a batch clustering algorithm that always starts its merge process from the bottom level; it is not desirable for evolving data since previous clustering results are ignored completely. Therefore, we propose an alternative hierarchical clustering algorithm called *Bayesian Split-Merge Clustering (BSMC)*. BSMC is a probabilistic version of traditional split-merge clustering algorithm such as ISO-DATA [11]. As its name implies, BSMC obtains the optimal partition through split and merge procedures. The decision of splitting or merging depends on the approximate posterior of partitions based on Bayesian clustering model. Therefore, it can bypass the model selection problem. Moreover, BSMC is appropriate for evolving data since it can start clustering from any intermediate level of the propagated tree.

Given an initial partition, we recursively split clusters in so-called the *split stage* as long as the probability of splitting is larger than 0.5. After that, pairs of clusters are merged in a recursive manner as long as the probability of merging is larger than 0.5, which is done in the *merge stage*. The procedure for BSMC is illustrated in Fig. 1.

3.1 Initial Partitions

At each time step, the initial partition ϕ_0 is obtained from the previous clustering result. If a new data point enters, a new cluster is created for the new data point. If no initial partition is given—for example, at the first time step, ϕ_0 is set to a single cluster containing all data.

3.2 Split Stage

Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ be a dataset, ϕ_0 be an initial partition and ϕ be the incumbent solution at a new time step. (The time index is omitted for simplicity.) Initially, we set $\phi = \phi_0$, which corresponds to $\{\mathbf{X}_k^\phi\}_{k=1}^{K_\phi}$. In the split stage, we test whether any of these clusters should be split into two or more clusters. By the Bayesian clustering model, the posterior probability of ϕ is given by

$$p(\phi|\mathbf{X}) = \frac{p(\mathbf{X}|\phi)p(\phi)}{\sum_{\phi' \in \Phi(\mathbf{X})} p(\mathbf{X}|\phi')p(\phi')}. \quad (10)$$

To estimate this posterior without considering all partitions, we test the partitions made by splitting current clusters. Let ϕ^s be a partition that the current cluster \mathbf{X}_k^ϕ is split into two clusters $\mathbf{X}_i^{\phi^s}$ and $\mathbf{X}_j^{\phi^s}$ and other clusters remain unchanged. One can propose ϕ^s by any appropriate bisecting algorithm such as k -means clustering, spectral clustering or graph cut. Then, we obtain

$$p(\phi|\mathbf{X}) < \frac{p(\mathbf{X}|\phi)p(\phi)}{p(\mathbf{X}|\phi)p(\phi) + p(\mathbf{X}|\phi^s)p(\phi^s)} = \left\{ 1 + \frac{p(\mathbf{X}|\phi^s)p(\phi^s)}{p(\mathbf{X}|\phi)p(\phi)} \right\}^{-1}, \quad (11)$$

which computes a loose upper bound of $p(\phi|\mathbf{X})$ using ϕ^s only. The upper bound gets tighter as $p(\mathbf{X}, \phi^s)$ increases. Although the bound is not tight for the accurate computation of $p(\phi|\mathbf{X})$, it is sufficient to check the optimality of ϕ .

Suppose that we define the split probability p_{split} as

$$p_{\text{split}} = 1 - \left\{ 1 + \frac{p(\mathbf{X}|\phi^s)p(\phi^s)}{p(\mathbf{X}|\phi)p(\phi)} \right\}^{-1}. \quad (12)$$

If $p_{\text{split}} > 0.5$, $p(\phi|\mathbf{X}) < 0.5$ by Eq. (11). Therefore, we conclude that ϕ is not optimal. The ratio in p_{split} can easily be computed since the terms for clusters other than \mathbf{X}_k^ϕ are canceled out. Using DP prior in Eq. (7), p_{split} is given by

$$p_{\text{split}} = 1 - \left\{ 1 + \frac{\alpha \Gamma(N_i^{\phi^s}) \Gamma(N_j^{\phi^s}) p(\mathbf{X}_i^{\phi^s}) p(\mathbf{X}_j^{\phi^s})}{\Gamma(N_k^\phi) p(\mathbf{X}_k^\phi)} \right\}^{-1}. \quad (13)$$

If $p_{\text{split}} > 0.5$, we set $\phi = \phi^s$. Then, for the two split clusters $\mathbf{X}_i^{\phi^s}$ and $\mathbf{X}_j^{\phi^s}$, we repeat the same procedure recursively as long as $p_{\text{split}} > 0.5$. The recursion for all initial clusters achieves the partition that is not desirable to split any further.

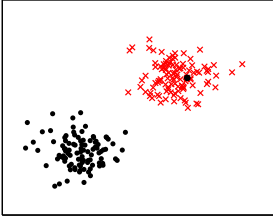


Fig. 2. A partition that requires postprocessing. The isolated black circle in the red cross cluster can simply be allocated to the red cross cluster in the postprocessing stage.

3.3 Merge Stage

In the merge stage, we determine whether any pairs of split clusters should be merged—for example, the [blue] and the [green] clusters in Figure 1 are more natural to be merged after split stage. Let \mathbf{X}_i^ϕ and \mathbf{X}_j^ϕ be a pair of clusters under the current optimal partition. By the similar arguments in Section 3.2, we check whether $p(\phi|\mathbf{X})$ is large enough by proposing a merged partition. Let ϕ^m be a partition that merges \mathbf{X}_i^ϕ and \mathbf{X}_j^ϕ into $\mathbf{X}_k^{\phi^m}$ and leaves other clusters unchanged. Similar to the split stage, p_{merge} is given by

$$p_{\text{merge}} = 1 - \left\{ 1 + \frac{\Gamma(N_k^{\phi^m})p(\mathbf{X}_k^{\phi^m})}{\alpha\Gamma(N_i^\phi)\Gamma(N_j^\phi)p(\mathbf{X}_i^\phi)p(\mathbf{X}_j^\phi)} \right\}^{-1}. \quad (14)$$

If $p_{\text{merge}} > 0.5$, we conclude that ϕ needs to be improved. As in BHC, we compute p_{merge} for all pairs of clusters and merge the pairs with the largest p_{merge} . We repeat the same procedure as long as the largest $p_{\text{merge}} > 0.5$.

3.4 Quality of the Solution

We can prove that $p(\phi|\mathbf{X})$ always increases by the splitting and merging:

$$1 - \left\{ 1 + \frac{p(\mathbf{X}|\phi')p(\phi')}{p(\mathbf{X}|\phi)p(\phi)} \right\}^{-1} > \frac{1}{2} \iff p(\mathbf{X}|\phi')p(\phi') > p(\mathbf{X}|\phi)p(\phi), \quad (15)$$

where $\phi' \in \{\phi^s, \phi^m\}$. Although this does not guarantee the optimality, it justifies the use of BSMC for the situations where good initial solutions are given, like video segmentation. According to our observation, BSMC provides quality solutions for complex and fast changing videos.

3.5 Postprocessing

Contrary to other point-wise inference algorithms, BSMC is a cluster-wise algorithm. Although this cluster-wise operations make BSMC efficient, some point-wise errors might occur as presented in Fig. 2. Since the overall cluster structure is found after the split and merge stage, these errors are easily fixed by allocating each point to the clusters having the closest center. The entire procedure of BSMC is summarized in Algorithm 1.

Algorithm 1. Bayesian Split-Merge Clustering**Input:** Initial partition ϕ_0 and dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$.**Output:** Optimal partition ϕ^* .Initialize $\phi = \phi_0$.• **Split stage****for** $k = 1, \dots, K_\phi$ **do**Propose ϕ^s by bisecting \mathbf{X}_k^ϕ into $\mathbf{X}_i^{\phi^s}$ and $\mathbf{X}_j^{\phi^s}$.**if** $p_{\text{split}} > 0.5$ **then**Split \mathbf{X}_k^ϕ into $\mathbf{X}_i^{\phi^s}$ and $\mathbf{X}_j^{\phi^s}$ (Set $\phi \leftarrow \phi^s$.)Recursively split $\mathbf{X}_i^{\phi^s}$ and $\mathbf{X}_j^{\phi^s}$.**end if****end for**• **Merge stage**Compute p_{merge} for all pairs of split clusters.**while** The maximum $p_{\text{merge}} > 0.5$ and $K_\phi > 1$ **do**Merge the maximum p_{merge} pair ($\phi \leftarrow \phi^m$) and update p_{merge} .**end while**• **Postprocessing****for** $n = 1, \dots, N$ **do**Allocate \mathbf{x}_n to the cluster with the closest mean.**end for** $\phi^* \leftarrow \phi$.

4 Video Segmentation

BSMC can be naturally applied to video segmentation in the spatio-temporal domain. In this section, we describe three steps to accomplish video segmentation results perceptually consistent and temporally coherent.

4.1 Pixel Clustering

We first extract RGB color values (or xy -RGB vectors to incorporate spatial constraints) from all pixels in the input image and cluster them. Since our method does not suffer from the problem of choosing the proper number of segments, it can deal with changing number of segments throughout the video. Furthermore, we can provide the clustering result in the previous frame as an initial partition when a new frame arrives. Then, the initial clusters are typically split near the boundaries of moving objects and the split clusters merge to build new clusters. This approach gives segmentation results that are consistent in the major boundaries. We call this procedure *pixel clustering*.

4.2 Second Merge Stage by Histogram Feature

Since pixel clustering employs local features only, clustering results may not be consistent temporally due to the jitters in the regions involving complex textures

and coherent with human perception that often treats semantically related but textured areas as a single segment.

To overcome such limitations, we adopt the idea of region-based segmentation proposed in [4]. We run the second merge stage, based on *histogram features* obtained from regions resulting from the pixel clustering. Using these histogram features, the similarities between regions are measured by color distributions of the regions. Therefore, two textured regions with similar color distributions may have high probability of merging. To define the similarity between histograms, we introduce a probabilistic model for histograms. Let $\mathbf{h} = [h_1 \dots h_K]^\top$ be a K -bin color histogram. Following [12], we use the multinomial distribution for the likelihood of histograms, which is given by

$$p(\mathbf{h}|\boldsymbol{\beta}) = \frac{M!}{\prod_{k=1}^K h_k!} \prod_{k=1}^K \beta_k^{h_k}, \quad (16)$$

where $M = \sum_k h_k$ is a normalization constant², $\boldsymbol{\beta} = [\beta_1 \dots \beta_K]^\top$ is a parameter that defines the probability of each bin. We use the Dirichlet distribution for $\boldsymbol{\beta}$ that is a conjugate prior of multinomial distribution as

$$p(\boldsymbol{\beta}|\boldsymbol{\pi}) = \frac{\Gamma\left(\sum_{k=1}^K \pi_k\right)}{\prod_{k=1}^K \Gamma(\pi_k)} \prod_{k=1}^K \beta_k^{\pi_k-1}, \quad (17)$$

where $\boldsymbol{\pi}$ is a hyperparameter for Dirichlet distribution. Now, we can define p_{merge} under these probabilistic models. Denoting two sets of histograms by $\mathbf{H}_i = \{\mathbf{h}_{i,1}, \dots, \mathbf{h}_{i,N_i}\}$ and $\mathbf{H}_j = \{\mathbf{h}_{j,1}, \dots, \mathbf{h}_{j,N_j}\}$, which represent two clusters of regions, the probability of merging these two clusters is given by

$$p_{\text{merge}} = 1 - \left\{ 1 + \frac{\Gamma(N_i + N_j)p(\mathbf{H}_i \cup \mathbf{H}_j)}{\alpha \Gamma(N_i)\Gamma(N_j)p(\mathbf{H}_i)p(\mathbf{H}_j)} \right\}^{-1}. \quad (18)$$

We iteratively merge regions as long as the maximum p_{merge} is greater than 0.5. Note that we can restrict candidates pairs to be adjacent to each other to incorporate spatial constraints.

4.3 Matching Clusters between Frames

Since our algorithm is based on the splitting and merging process, maintaining segment identities across frames is not straightforward. We present a simple solution to match clusters between adjacent frames to maintain cluster identity. Suppose that \mathbf{H}_t and \mathbf{H}_{t+1} are the sets of histograms extracted from the regions made by clustering at the frame t and $t+1$, respectively. We perform another merge stage on $\mathbf{H}_t \cup \mathbf{H}_{t+1}$; if $\mathbf{h}_{t,i}$ and $\mathbf{h}_{t+1,j}$ belong to the same cluster, they are matched and identified as a same segment. An additional benefit of this strategy is improved temporal coherency; erroneously separated segments in \mathbf{H}_t can be merged using additional information given by \mathbf{H}_{t+1} . The entire process of segmentation is summarized in Fig. 3.

² We normalize \mathbf{h} and multiply M to compare regions with different sizes.

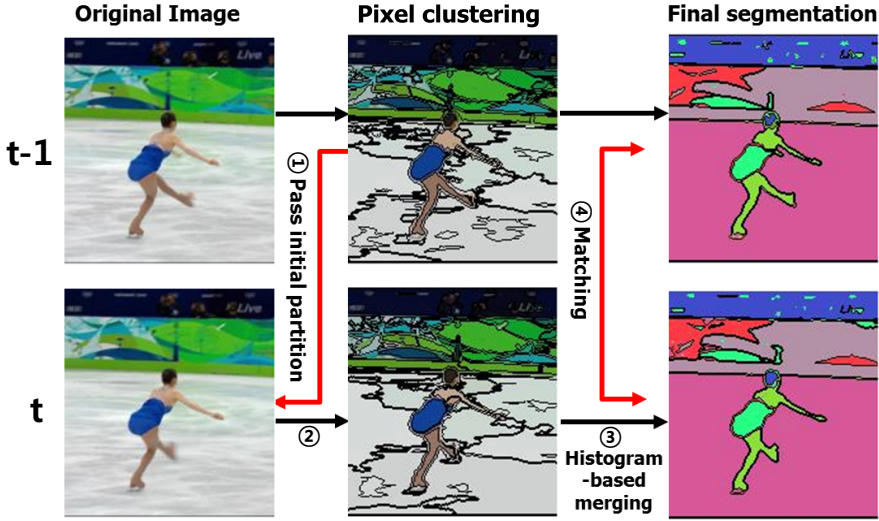


Fig. 3. Video segmentation process. 1. Passing initial partition 2. Pixel segmentation using BSMC 3. Histogram-based merging. 4. Matching two frames.

5 Experiments

5.1 Clustering Simulation

To evaluate the solutions by BSMC, we compared BSMC with collapsed Gibbs sampler for DPM [13], BHC [10] and DPChain [8] on a synthetic dataset. The dataset is composed of 16 frames evolving over time where points in each frame are generated from a Gaussian mixture model (Fig.4(a)). Throughout the sequence, the characteristics of data including the number of clusters change drastically over time, which violates the assumption of temporal smoothness in evolutionary clustering.

For all algorithms, we used the Gaussian likelihood and Gaussian-Wishart prior as parameters:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (19)$$

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \mathcal{N}(\boldsymbol{\mu}|\mathbf{m}, (\tau\boldsymbol{\Lambda})^{-1})\mathcal{W}(\boldsymbol{\Lambda}|\mathbf{W}, \nu) \quad (20)$$

where $\boldsymbol{\mu}$ is the mean of a cluster, $\boldsymbol{\Lambda}$ is a precision and $\{\mathbf{m}, \tau, \nu, \mathbf{W}\}$ are hyperparameters. In all experiments, we set \mathbf{m} and \mathbf{W} to the sample mean and precision of the dataset and fixed $\tau = 0.01$ and $\nu = 15$. BSMC employed k -means clustering for bisection. We iterated 100 times for the collapsed Gibbs sampler and DPChain. For DPChain, initial labels are given by the result of the previous time step. We controlled the smoothness parameter λ to 0.5 (DPChain1) and 1 (DPChain2). Smaller λ means more temporal smoothness. For all algorithms except BHC, we conducted clustering 10 times and averaged the results to handle randomness.

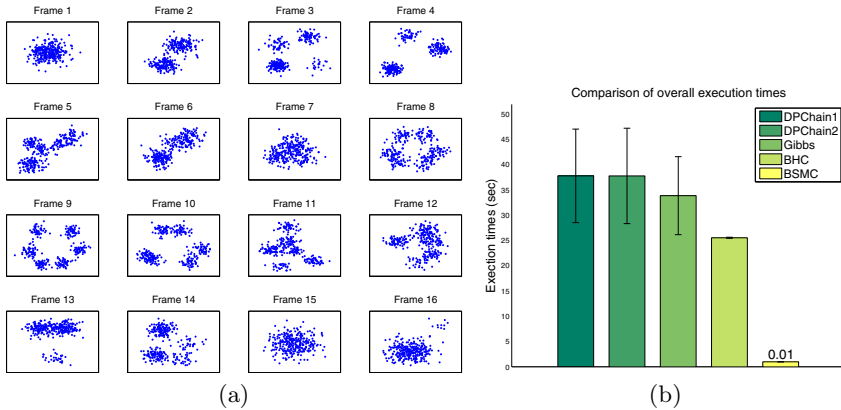


Fig. 4. (a) Synthetic dataset generated using Gaussian mixture models with moving centers. (b) Average computing time in seconds.

According to our experiments, the accuracies of all algorithms are almost identical. However, in terms of running time, BSMC is faster at least by three orders of magnitude than all other algorithms (Fig. 4(b)). The computing time of BSMC is dominated by the bisecting algorithm due to its cluster-wise inference. Therefore, provided that the bisecting algorithm is efficient, BSMC would be significantly faster while maintaining comparable clustering performance.

5.2 Video Segmentations

We tested our algorithm on real world video sequences, which include dynamic movements and complex patterns. We compared our method with a offline algorithm, hierarchical graph-based video segmentation (EHGBVS) [4], and an online algorithm, Bayesian order-adaptive clustering (BOAC) [9].

For pixel clustering, we used k -means clustering in the split stage, and the Gaussian likelihood and Gaussian-Wishart prior for underlying probabilistic models. For color histograms in histogram merging, we employed 3D color histograms. For the BOAC, we used 4-bin RGB histograms for each channel and set the window radius to 2. We iterated 100 times for the first frame and 2 ~ 5 times for the rest of frames. For EHGBVS, we used the default settings provided in the project website.³

Qualitative Comparison. We tested five sequences: *skating* (180×320 , 185 frames), *jump* (224×352 , 157 frames), *sprint* (320×480 , 442 frames), *matrix* (272×480 , 171 frames) and *earth* (170×400 , 98 frames).⁴ Note that, contrary to the online algorithms such as BSMC and BOAC, EHGBVS is a batch algorithm

³ <http://neumann.cc.gt.atl.ga.us/segmentation/>

⁴ All videos are downloaded from YouTube except the *earth* sequence, which is obtained from <http://cpl.cc.gatech.edu/projects/videosegmentation/> [4].



Fig. 5. Comparison of three video segmentation algorithms. From top to bottom, *skating*, *jump*, *sprint*, *matrix* and *earth* sequence are presented. From left to right, original sequence and the results by BSMC, BSMC with spatial constraints, BOAC and EHGBVS are illustrated. Frame numbers are shown at upper-left corners.

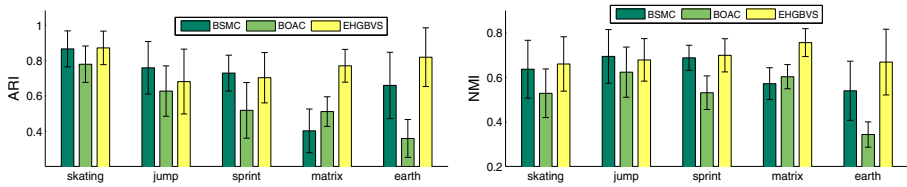


Fig. 6. Average ARI and NMI values of three algorithms for five sequences

that performs a global optimization for segmentation. Also, it can maintain segment identities in 3D spatio-temporal volume and has advantage to visualize results with less flickering. However, our algorithm still demonstrates visually good performance compared to EHGBVS with consistency in region boundaries while BOAC produces many noisy segments (Fig. 5). As the tested videos involve nontrivial patterns and drastic motions, BOAC requires many iterations for convergence. BSMC was approximately 4 ~ 8 times faster than BOAC in our MATLAB implementation; for the *skating* sequence, ours took 207 secs while BOAC took 1647 secs. EHGBVS is implemented and run on a completely different systems with parallel architecture; direct comparison of running time is unavailable.

Quantitative Evaluation. We compared three algorithms quantitatively based on ground-truths manually constructed by five people. We evaluated the segmentation result by Adjusted Rand Index (ARI) [14] and Normalized Mutual Information (NMI) [15] for randomly selected frames from each sequence. We emphasize again that EHGBVS is an offline method which is expected to outperform online methods since it clusters past, present and future frames simultaneously. BSMC outperforms the BOAC except for the *matrix* sequence for both ARI and NMI, while being comparable to EHGBVS as illustrated in Fig. 6.

6 Conclusion

We proposed a novel on-line clustering algorithm called Bayesian split-merge clustering. BSMC can cluster evolving data efficiently and flexibly, while preserving temporal consistency and adapting to drastic changes. We applied our algorithm to online video segmentation through three steps—pixel clustering, merge by histogram, and temporal matching—and obtained good segmentation results with significantly improved efficiency.

Acknowledgments. This work was supported by MEST Basic Science Research Program through the NRF of Korea (2012-0003697), NIPA ITRC Support Program (NIPA-2012-H0301-12-3002), NIPA Program of Software Engineering Technologies Development and Experts Education, MEST Converging Research Center Program (2012K001343), MKE and NIPA “IT Consilience Creative Program” (C1515-1121-0003), and NRF WCU Program (R31-10100).

References

1. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary clustering. In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, PA (2006)
2. Chi, Y., Song, X., Zhou, D., Hino, K.: Evolutionary spectral clustering by incorporating temporal smoothness. In: *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, San Jose, CA (2007)
3. DeMenthon, D., Megret, R.: Sptio-temporal segmentation of video by hierarchical mean shift analysis. Technical Report LAMP-TR-090,CAR-TR-978,CS-TR-4388,UMIACS-TR-2002-68, University of Maryland, College Park (2002)
4. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA (2010)
5. Wang, J., Thieson, B., Xu, Y., Cohen, M.: Image and Video Segmentation by Anisotropic Kernel Mean Shift. In: Pajdla, T., Matas, J. (eds.) *ECCV 2004*. LNCS, vol. 3022, pp. 238–249. Springer, Heidelberg (2004)
6. Antoniak, C.E.: Mixtures of Dirichlet processes with applications to Bayesian non-parametric problems. *The Annals of Statistics* 2, 1152–1174 (1974)
7. Ferguson, T.S.: A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* 1, 209–230 (1973)
8. Xu, T., Zhang, Z., Yu, P.S., Long, B.: Dirichlet process based evolutionary clustering. In: *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Pisa, Italy (2008)
9. Orbanz, P., Braendle, S., Buhmann, J.M.: Bayesian Order-Adaptive Clustering for Video Segmentation. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) *EMMVCPR 2007*. LNCS, vol. 4679, pp. 334–349. Springer, Heidelberg (2007)
10. Heller, K.A., Ghahrahmani, Z.: Bayesian hierarchical clustering. In: *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany (2005)
11. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. *ACM Computing Surveys* 31, 264–323 (1999)
12. Orbanz, P., Buhmann, J.M.: Smooth Image Segmentation by Nonparametric Bayesian Inference. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) *ECCV 2006*. LNCS, vol. 3951, pp. 444–457. Springer, Heidelberg (2006)
13. Escobar, M.D., West, M.: Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* 90, 577–588 (1995)
14. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* 2, 193–218 (1985)
15. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, Toronto, Canada (2003)