Int J CARS (2016) 11:1937–1949
DOI 10.1007/s11548-016-1431-2

ORIGINAL ARTICLE

Shot boundary detection in endoscopic surgery videos using a variational Bayesian framework

Constantinos Loukas[1,3] · Nikolaos Nikiteas[1] · Dimitrios Schizas[2] · Evangelos Georgiou[1]

Received: 2 December 2015 / Accepted: 27 May 2016 / Published online: 11 June 2016

Abstract

Purpose Over the last decade, the demand for content management of video recordings of surgical procedures has greatly increased. Although a few research methods have been published toward this direction, the related literature is still in its infancy. In this paper, we address the problem of shot detection in endoscopic surgery videos, a fundamental step in content-based video analysis.

Methods The video is first decomposed into short clips that are processed sequentially. After feature extraction, we employ spatiotemporal Gaussian mixture models (GMM) for each clip and apply a variational Bayesian (VB) algorithm to approximate the posterior distribution of the model parameters. The proper number of components is handled automatically by the VBGMM algorithm. The estimated components are matched along the video sequence via their Kullback–Leibler divergence. Shot borders are defined when component tracking fails, signifying a different visual appearance of the surgical scene.

Results Experimental evaluation was performed on laparoscopic videos containing a variable number of shots. Performance was measured via precision, recall, coverage and overflow metrics. The proposed method was compared with GMM and a shot detection method based on spatiotemporal motion differences (MotionDiff). The results demonstrate that VBGMM has higher performance than all other methods for most assessment metrics: precision and recall >80 %, coverage: 84 %. Overflow for VBGMM was worse than MotionDiff (37 vs. 27 %).

Conclusions The proposed method generated promising results for shot border detection. Spatiotemporal modeling via VBGMMs provides a means to explore additional applications such as component tracking.

Keywords Video content analysis · Shot detection · Border detection · Variational Bayes · Tracking · Surgery

Introduction

Minimally invasive surgery (MIS) is a widespread therapeutic procedure with well-documented benefits for the patient such as shorter hospital stays, less pain and shorter recovery. Another advantage is the effortless video recording via the endoscope used to visualize the anatomical area in operation. The videos may be stored in the hospital's server or uploaded into dedicated Web-based multimedia resources for reasons such as: as an educational material to teach junior surgeons [1], for retrospective evaluation and improvement in the applied technique [2], and to provide patients a personal copy for later review. In addition, digital video recording and archiving of surgeries is considered mandatory in some countries in order to provide evidence for lawsuits in case of malpractice [3]. Alternatively, some surgeons may record specific video segments showing the most critical parts of the operation.

Despite the growing availability of surgical videos, tools and methods for effective organization and management of

✉ Constantinos Loukas
cloukas@med.uoa.gr

[1] Simulation Center, Laboratory of Medical Physics, Medical School, National and Kapodistrian University of Athens, Athens, Greece

[2] 1st Department of Surgery, Laiko General Hospital, University of Athens, Athens, Greece

[3] Medical Physics Lab-Simulation Center, School of Medicine, University of Athens, Mikras Asias 75 str., 11527 Athens, Greece

related databases are still limited. Typically, tasks such as video annotation and content representation are performed manually via tags that provide keywords or a short description about the employed technique. In the general field of multimedia analysis, the literature abounds with algorithms that target video content analysis for various applications such as retrieval, annotation and detection of semantic information [4]. A fundamental prerequisite for most video abstraction and content representation approaches is the detection of shot boundaries. In this context, a shot is considered as a sequence of video frames that represent a continuous spatiotemporal action [5], whereas a boundary is defined as the time where the content of the shot presents a dramatic change. The later may be the result of an abrupt movement of the camera or object(s) being monitored. Therefore, there are significant content correlations between frames within a shot. Shots are considered to be fundamental units in video content organization and the primitives for higher-level semantic annotation and retrieval tasks [6]. After shot segmentation, it is then straightforward to establish the overall video context as a collection of representations arisen from the analysis of the individual shots (e.g., shot-based feature descriptors, keyframes, etc.). Hence, retrieval of frames or even tasks relevant to a video content query may lead to more effective management of video databases.

The aforementioned benefits also apply to the field of surgical video analysis, where software systems for video database organization are still limited. Representing a video as a group of descriptors, keyframes or scenes, extracted from each shot detected, may lead to a number of interesting applications such as video summarization and retrieval of shots with content similar to a query shot, where the similarity may refer to semantic information such as: the tool used, organ/anatomy contained or task performed, in the query shot. Moreover, extracting relevant keyframes or keyshots may be considered as a first step for video-based assessment of surgical skills. For example, a higher number of shots may be correlated to more erratic camera movements (and consequently worse performance). Although this information on its own does not define entirely ‘surgical dexterity,’ it may well be used in conjunction with other information extracted from the shots, such as the amount of specular reflections. Alternatively, given the extracted number of shots and associated keyframes, surgical educators and performance reviewers could concentrate on particular shots. As another example, consider a surgical educator or trainee who for didactic purposes would like to retrieve segments that show how a predefined activity is performed or retrieve shots of an important task performed with a specific instrument. Manual annotation of digital media provides only a minor way to describe what has really happened in a surgery. Currently, users have to go through the whole video stream in order to visualize such events.

In the area of multimedia analysis, there are plenty of methodologies for video shot detection [7]. A typical approach includes the computation of a dissimilarity metric between consecutive frames based on various representations, such as motion, color, texture and edge features as well as combinations [8–10]. These algorithms are mostly designed to detect cuts and gradual transitions (e.g., fade, dissolve, etc.), which are commonly found in movie videos and TV news. However, in endoscopic surgery there are fundamental challenges for shot boundary detection. First, the recordings typically contain one shot per video file, unless the camera is removed/reinserted several times (e.g., for removing haze from the lenses). Second, there is significant color similarity between the consecutive frames. Third, the camera is close to the area in operation, which is heavily magnified by the lenses, and thus, even a small movement of the camera could result in a significant change of the field of view. Fourth, the lighting conditions are not constant since the light source is attached to the front lenses of the camera, which is operated manually. Finally, there is high amount of noise due to specular reflections and frequent movement of the camera, the instruments and the operated tissue.

In endoscopic surgery, shot detection techniques based on video content analysis are still limited. The most relevant work is that of Primus et al. [3], where a method based on differences of motion is proposed. Using the well-known Kanade–Lucas–Tomasi tracker, an aggregate movement vector was extracted separately for nine areas of each frame. The segmentation border was based on the spatiotemporal standard deviation of these vectors. In another related work [11], the extended Kalman filter was applied to identify key episodes by encoding motion of tracked salient features on video data from robotically assisted MIS procedures. Episode borders were defined when feature tracking fails. Moreover, probabilistic motion modeling of tracked features was used for episode representation. Video content analysis in MIS has attracted increased interest lately with applications in various directions such as: task boundary detection [12], surgery classification [13], detection of irrelevant segments [14], keyframe extraction [15], skills assessment [16] and video retrieval [17]. However, the overall methodology in these works (except keyframe extraction) is based on the design of appropriate similarity criteria with regard to the query video/frame, which is different to shot detection where there is considerable difficulty in selecting reference data from optimal video shots.

In the neighboring field of surgical process modeling, there has also been a small number of vison-based techniques. For example, the potential of using video data for segmentation of a surgery into its key phases was addressed in [18]. Using color information and canonical correlation analysis, the average accuracy was about 60 %. In [19], an algorithm based on kinematic feature analysis was recently proposed

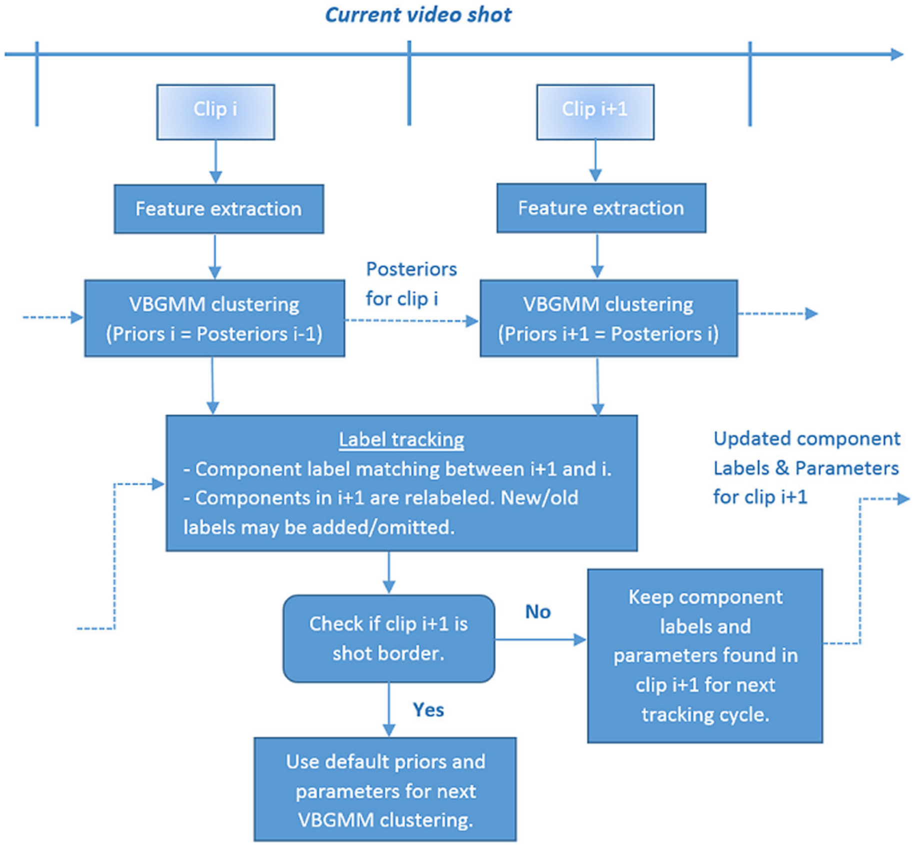
Current video shot
Clip i
Clip i+1
Feature extraction
Feature extraction
VBGMM clustering (Priors i = Posteriors i-1)
Posteriors for clip i
VBGMM clustering (Priors i+1 = Posteriors i)
Label tracking
- Component label matching between i+1 and i.
- Components in i+1 are relabeled. New/old labels may be added/omitted.
Updated component Labels & Parameters for clip i+1
Check if clip i+1 is shot border.
No
Keep component labels and parameters found in clip i+1 for next tracking cycle.
Yes
Use default priors and parameters for next VBGMM clustering.

Fig. 1 A graphical overview of the main steps of the proposed shot border detection technique

for segmentation of endoscopic videos into ‘smoke events.’ In another study [20], a method based on user-selected visual features and appropriate dissimilarity metrics was proposed for video summarization of MIS operations.

Content-based video analysis techniques are also encountered in medical domains other than endoscopic surgery. For example, a method employing visual features extracted from compressed videos together with audio analysis was proposed in [21] for detecting semantic units in colonoscopy. A novel methodology for semantic encoding of endoscopic visual content is described in [22]. Two techniques (edge based and clustering based) for detection of uninformative endoscopy frames are proposed in [23]. Video content representation of low-level tasks in eye surgery has also attracted some interest lately [24]. However, the visual content of the medical procedures in the aforementioned studies is significantly different to that encountered in MIS where there is significant tissue deformation and camera/instrument motion, and the lighting conditions are variable.

The purpose of this paper is to propose a content-based methodology for shot border detection in laparoscopic videos. The first contribution lies in the application of a variational Bayesian (VB) framework for computing the posterior distribution of spatiotemporal Gaussian mixture models (GMMs). In particular, the video is first decomposed into a series of consecutive clips. The VBGMM algorithm is applied on feature vectors extracted from each clip. The proper number of components is estimated automatically via the sparseness of a Dirichlet prior on the mixture weights. The second contribution lies in the tracking of the mixture components, which is accomplished by estimating the shortest Kullback–Leibler distance between the posteriors of the components found in each pair of consecutive clips. Component entry and exit are handled by the resulting number of components. The final step of the method examines whether a clip is a border of the current shot. This is based on the failure of the component tracking process, via a criterion that signifies the appearance of a different visual content with regard to the past clips. A graphical overview of the main steps is presented in Fig. 1. The method was tested on video segments containing a variable number of shots. Comparison among VBGMM, GMM and a recently published method [3] was also performed.

Methods

Video processing

Given a video segment, we first decompose into consecutive clips of fixed duration: $\mathcal{U} = (u_1, \ldots, u_t, \ldots)$. Each clip is processed sequentially as a separate spatiotemporal volume. A GMM is employed, where a feature vector, y, extracted from each pixel in the volume, is assumed to have been drawn from a set of K components:

$$p\left(\boldsymbol{y}/\theta, \boldsymbol{\pi}\right) = \sum_{j=1}^{K} \pi_j p\left(\boldsymbol{y}|\boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j^{-1}\right) \tag{1}$$

where π_j is the weight for the jth component, K is the number of components and $p\left(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}\right)$ is a multivariate Gaussian distribution with parameters $\boldsymbol{\mu}$ (mean) and $\boldsymbol{\Lambda}$ (inverse covariance or precision). In this context, a component constitutes a cluster of pixels with similar characteristics described by the entries of the feature vector.

The overall GMM parameters are denoted as: $\Theta = \left\{\theta_j\right\}$, where $\theta_j = \left\{\boldsymbol{\pi}_j, \boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j\right\}$. A common approach in computing these parameters is to introduce a latent variable, $\boldsymbol{z} = \left\{z_j\right\}$, that denotes one of the K components and then apply expectation maximization (EM) to find a maximum likelihood (ML) solution. In particular, the E step evaluates the posterior of the latent variables, $p\left(z_j/\boldsymbol{y}\right)$, using the current model parameter values. Then, the M step re-estimates the parameters using the current posteriors. The equations for the model parameters are obtained by maximizing the expectation of the complete data likelihood:

$$Q\left(\Theta, \Theta^{\text{old}}\right) = \sum\nolimits_{\boldsymbol{Z}} p\left(\boldsymbol{Z}/\boldsymbol{Y}, \boldsymbol{\Theta}^{\text{old}}\right) \ln p(\boldsymbol{Y}, \boldsymbol{Z}|\Theta) \tag{2}$$

where $\boldsymbol{Z} = \{z_i\}$, $\boldsymbol{Y} = \left\{\boldsymbol{y}_i\right\}$, $i = 1, \ldots, N$, with N denoting the total number of feature vectors in the video clip.

Variational Bayes framework

A significant limitation of the aforementioned approach is that it requires setting K in advance. Techniques such as the maximum value of the Bayesian Information Criterion (BIC) are not always suitable for finding the optimum value for K [25]. Moreover, ML estimation does not provide a means for excluding redundant components, typically when initially setting large K. Another reason is that in the case of time-varying data, such as those employed in this study, it is not easy to find component correspondence between video clips, especially when a component disappears in the upcoming clip, or a new one is introduced.

A Bayesian treatment of the mixture model resolves many of these obstacles. In this paper, we employ spatiotemporal GMMs for each video clip and apply VB inference to approximate the full posterior distribution on the model parameters [26,27]. Based on the VB framework, the parameters $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$ are modeled with conjugate priors:

$$p\left(\boldsymbol{\pi}\right) = \text{Dir}\left(\boldsymbol{\pi}|\alpha_0\right) \tag{3}$$

$$p\left(\boldsymbol{\mu}, \boldsymbol{\Lambda}\right) = \prod\nolimits_{j=1}^{K} p(\boldsymbol{\mu}_j|\boldsymbol{\Lambda}_j) p\left(\boldsymbol{\Lambda}_j\right)$$
$$= \prod\nolimits_{j=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_j|\boldsymbol{m}_0, \left(\beta_0\boldsymbol{\Lambda}_j\right)^{-1}\right) \mathcal{W}\left(\boldsymbol{\Lambda}_j|\boldsymbol{W}_0, \nu_0\right) \tag{4}$$

where Dir denotes the Dirichlet distribution with an associated parameter α_0, $p(\boldsymbol{\mu}_j|\boldsymbol{\Lambda}_j)$ and $p\left(\boldsymbol{\Lambda}_j\right)$ are the normal and Wishart distributions, respectively, $\boldsymbol{m}_0$, β_0 are prior parameters for the distribution of mean $\boldsymbol{\mu}_j$ and $\boldsymbol{W}_0, \nu_0$ are prior parameters for the distribution of precision $\boldsymbol{\Lambda}_j$.

In order to find the posterior distribution of the model parameters given data $\boldsymbol{Y}$, one needs to compute:

$$p(\Theta|\boldsymbol{Y}) = \frac{p(\boldsymbol{\Upsilon}|\boldsymbol{\Theta})\, p\left(\boldsymbol{\Theta}\right)}{\int p\left(\boldsymbol{Y}, \boldsymbol{\Theta}\right) \mathrm{d}\boldsymbol{\Theta}}$$
$$= \frac{p(\boldsymbol{Y}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\boldsymbol{Z}|\boldsymbol{\pi}) p\left(\boldsymbol{\pi}\right) p(\boldsymbol{\mu}|\boldsymbol{\Lambda}) p\left(\boldsymbol{\Lambda}\right)}{\int p\left(\boldsymbol{Y}, \boldsymbol{\Theta}\right) \mathrm{d}\boldsymbol{\Theta}} \tag{5}$$

The main difficulty in the previous equation is that the marginal likelihood, or evidence, in the denominator is analytically intractable. VB approximation is an efficient method used for estimating such integrals. In brief, using an approximation distribution $q(\Theta|\boldsymbol{\Upsilon})$ the logarithm of the evidence can be written as:

$$\ln\, p\left(Y\right) = \mathcal{L}\left(q\right) + KL(q||p) \tag{6}$$

where $\mathcal{L}\left(q\right)$ is a functional of the distribution $q\left(\Theta/\boldsymbol{Y}\right)$:

$$\mathcal{L}\left(q\right) = \int q\left(\Theta|\boldsymbol{Y}\right)\, \ln \frac{p\left(\boldsymbol{Y},\, \Theta\right)}{q(\Theta|\boldsymbol{\Upsilon})} \mathrm{d}\Theta \tag{7}$$

and $KL(q||p)$ is the Kullback–Liebler divergence from $q\left(\Theta|\boldsymbol{\Upsilon}\right)$ to the true posterior distribution $p(\Theta|\boldsymbol{\Upsilon})$:

$$KL\left(q||p\right) = -\int q\left(\Theta|\boldsymbol{Y}\right)\, \ln \frac{p(\Theta|\boldsymbol{Y})}{q(\Theta|\boldsymbol{Y})} \mathrm{d}\Theta \tag{8}$$

Because $KL(q||p) \geq 0$, it follows that $\ln p\left(Y\right) \geq \mathcal{L}\left(q\right)$, with equality if and only if $q = p$. The variational distribution q is found by maximizing $\mathcal{L}\left(q\right)$. The only assumption made is that q is chosen to be factorizable between the latent variables and the parameters: $q\left(\boldsymbol{Z},\, \boldsymbol{p},\, \boldsymbol{\mu},\, \boldsymbol{\Lambda}\right) = q\left(\boldsymbol{Z}\right) q\left(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}\right)$. Thus, the true posterior distribution of a model parameter may be replaced with its variational approximation, which has an equivalent form but with different parameters:

$$q\left(\boldsymbol{\pi}\right) = \text{Dir}\left(\boldsymbol{\pi}|\boldsymbol{\alpha}\right) \tag{9}$$

$$q(\boldsymbol{\mu}_j|\boldsymbol{\Lambda}_j) q\left(\boldsymbol{\Lambda}_j\right) = \mathcal{N}\left(\boldsymbol{\mu}_j|\boldsymbol{m}_j, \left(\beta_j\boldsymbol{\Lambda}_j\right)^{-1}\right) \mathcal{W}\left(\boldsymbol{\Lambda}_j|\boldsymbol{W}_j, \nu_j\right) \tag{10}$$

The hyperparameters of interest for each component: $\omega_j = \left\{\boldsymbol{m}_j, \boldsymbol{W}_j, \nu_j, \beta_j, \alpha_j\right\}$, $\Omega = \left\{\omega_j\right\}$, are obtained under the VB framework using a set of update equations (a detailed derivation may be found in [28]). Based on these equations, the posterior distribution of the latent variables, also known as *responsibility*, that denotes the probability of the jth mixture component given a data vector, y_i, is calculated as:

$$\gamma_{ij} = \frac{\rho_{ij}}{\sum_{j=1}^{K} \rho_{ij}} \tag{11}$$

where

$$\begin{aligned} \ln \rho_{ij} \approx\ & \psi\left(\alpha_j\right) - \psi\left(\sum\nolimits_j \alpha_j\right) \\ & + 0.5 \left[\sum_{n=1}^{d} \psi\left(\frac{\nu_j + 1 - n}{2}\right) \right. \\ & \left. + \ln\left|\boldsymbol{W}_j\right| - d\beta_j^{-1} - \nu_j \left(\boldsymbol{y}_i - \boldsymbol{m}_j\right)^T \boldsymbol{W}_j \left(\boldsymbol{y}_i - \boldsymbol{m}_j\right) \right] \end{aligned} \tag{12}$$

The symbol T denotes the transpose operator, d is the dimensionality of the feature vector and $\psi(.)$ is the digamma function. The previous equation is essentially used to assign each pixel in u_t to the component with the highest responsibility value. The set of assigned labels for all pixels in u_t is denoted as L_t. Hence, the components essentially denote clusters of pixels in the video clip with similar feature values, whereas the labels are the 'tags' (indices) of these components.

Label tracking

Having obtained the labels L_t for all pixels in u_t, we next seek to determine the labels L_{t+1} for all pixels in the forthcoming clip u_{t+1}. Recall that the labels essentially indicate the indices of the components clustered by the VBGMM algorithm, and the relationship between labels and components is one to one for a specific video clip. Consequently, the model parameters of a certain component (that defines a cluster of pixels in a clip) are also linked to the label assigned to that component.

Now, to compute L_{t+1} one needs to compute the new responsibilities for all pixels in u_{t+1}, using the new data vectors $\boldsymbol{Y}_{t+1}$. This process is essentially the same to that described in the previous section, with the only difference being the initialization step: The priors for the mixture labels in u_{t+1} are set equal to the posteriors from u_t.

The remaining step is to find the correspondence of the labels L_{t+1} with regard to the previous ones L_t. This is achieved by computing the *KL* distance between the posteriors on the mixture parameters at time $t+1$ and t. Hence, a new feature vector $\boldsymbol{y}$ in u_{t+1} with label l_k is assigned a label $l_{k'}$ from the previous clip u_t when:

$$l_{k'} = \arg\min_{j \in L_{t+1}} KL\left(p(\boldsymbol{y}|\Theta_{t+1}, l_j)\right) || p(\boldsymbol{y}|\Theta_t, l_k)) \tag{13}$$

Note that the aforementioned process is followed only for those components that have significant amplitude weights. In fact, an important advantage of the VB framework is that the components with near-zero amplitude weights can be easily determined by calculating the expectation values $\mathbb{E}\left[\pi_j\right]$, $\forall j \in K$:

$$\mathbb{E}\left[\pi_j\right] = \frac{\alpha_j}{\sum_{k=1}^{K} \alpha_k} \tag{14}$$

where:

$$\alpha_j = \alpha_0 + \sum_{i=1}^{N} \gamma_{ij} \tag{15}$$

Hence, components with a weight smaller than threshold ε_π are deemed insignificant and are removed, whereas components with higher weights are considered significant and are kept for the next label tracking step ($\varepsilon_\pi = 0.01$ was used here).

Component entry and exit

As the video content changes through the consecutive clips, it is expectable that some components may disappear in the upcoming clip, or new ones may appear. In this case, it is essential to find these components and remove or add their labels, respectively.

Component removal is encountered when the number of significant components in u_t is greater than that found in u_{t+1}. Hence, those labels from u_t that are not reassigned a label in u_{t+1}, during label tracking, are removed and may be used for label assignment in a future step. The entry of new components is decided when the number of significant components in u_{t+1} is greater than that found in u_t. Any component that has not been given a label during tracking at this step is assigned a random one from the initial set of labels, with the only restriction being that this label is not already assigned to a component in the current clip u_{t+1}. As an example, consider a label set for clip u_t as: $L_t = \{6,\ 3,\ 2\}$. If, for the next clip u_{t+1}, VBGMM yields 4 significant labels, then the three of them will be assigned a unique label from the set L_t (based on the *KL* distance matching), whereas the fourth one can take any label from the initial set K, as long as it is different to the three ones already assigned (so it may be $L_{t+1} = \{6,\ 3,\ 2,\ 8\}$). Note that the new label added may have been used before in another clip. However, as will be described next, this does not affect the shot border detection process since the proposed method does not take into account the entire label correspondence history.

Shot border detection

The proposed shot boundary detection method is based on the label assignment process performed via the sequential processing of the video clips. In particular, we monitor the number of significant components tracked from the first clip of the current shot, up to the current clip, say u_{t+1}. Due to the variation of the video content, as a result of the camera and the instruments' movement, it is expectable that the tracking process will fail at some point. Tracking failure is considered when the number of significant components found in clip u_{t+1} falls below a minimum threshold value ε_L:

$$\varepsilon_L = \min\left(0.5\,av\{L_\tau^{sign}\}\right), 2),\ \tau = t, \ldots t-2 \tag{16}$$

where av is the average operator and $\{L_\tau^{sign}\}$ is the set containing past number of significant components, back to $t-2$.

The previous equation essentially denotes that the border of the shot occurs when the threshold is the minimum between two parameters: The 50 % of the average number of the significant components found in the last three video clips and the number 2. The first parameter was used to take into account some past history of the label assignment process; a 50 % deviation from the three most recent video clips is considered significant here. Number 2 was used essentially to denote that in the current clip tracking fails to provide more than two components (e.g., background and foreground).

Finally, after a border is detected, the whole process is reinitialized for the next video clip (say u_{t+2}, if u_{t+1} denotes the shot border found). Initial experimentation on individual clips showed that $K = 10$ was sufficient to represent the maximum number of components encountered in a video clip. With regard to the VB framework, the components' weight, mean and covariance were initialized via a standard GMM algorithm. The initial parameters for the prior distributions were selected so as to be sufficiently uninformative ($\boldsymbol{m}_0 = \boldsymbol{0}, \boldsymbol{W}_0 = 100\boldsymbol{I}, \nu_0 = 10, \beta_0 = 1, \alpha_0 = 0.001$, where $\boldsymbol{0}$ is the zero matrix and $\boldsymbol{I}$ the identity matrix).

Results

Dataset

The initial video collection included 8 laparoscopic cholecystectomy operations performed by two different surgeons over a period of one year. Each video corresponded to an operation performed on a different patient. The video resolution of the endoscope camera was 720 × 576, and the frame rate was 25 fps. From the initial collection, an experimental dataset of 53 video segments was randomly selected by the surgeons, $\boldsymbol{\mathcal{U}} = \{\mathcal{U}_M\}_{M=1}^{53}$, so that each segment contained from 0 up to 4 shot borders (i.e., 1 to 5 video shots). The video segments had duration: 204 ± 47 s (mean ± SD), and contained views from various phases of the surgery, such as gallbladder inspection, dissection, clipping and coagulation.

Each segment was decomposed into consecutive clips: $\mathcal{U} = (u_1, \ldots, u_t, \ldots)$; the clips had fixed duration equal to the frame rate of the camera. To reduce computational cost, the clips were temporally and spatially downsampled by a factor of 5 and 4, respectively. From each pixel in the downsampled volume, a feature vector containing the RGB color values and the 2D optical flow was extracted (five-dimensional feature vector). All clips from each segment were processed sequentially by the proposed method.

The ground truth for the shot borders was created by experienced operators, using a scheme similar to that reported in [3]. In particular, each video segment was annotated for events that relate to a combination of two activities: camera/instrument movement and different view of the surgical scene. Specifically, we were interested in identifying motion activities that also bring a significant change in the surgical scene. Examples included: camera removal/insertion, camera panning and expose of anatomical structures with the instruments. Such events not only relate to a movement activity, but also introduce a change in the timeline view of the surgery. The timings of these events were matched with the corresponding video clips, which defined the shot borders.

Label tracking

Figure 2 shows an example of the label tracking process for a series of 12 clips selected from a video segment (only the first frame from each clip is shown). The surgical scene shows the abdominal wall and the upper surface of the liver. In the first 5 clips, the camera is slightly moving to the right. The VBGMM framework detects 5 components that are firmly tracked by label tracking (label color coding is shown at the bottom). In the sixth frame, the obturator of the trocar is inserted from the right, and a new label (L7) is assigned due to the different color of this component. Clips 7, 8 show the entry of a larger portion of the obturator, which is successfully tracked, along with the previous components. In clips 9–12, we have the entry of another component, the cannula of the trocar. The algorithm successfully identifies its spatial region, although it does not assign a new label, mainly due to color similarity with a component found previously. In clip 12, the obturator is removed and so its label by the algorithm. Note that throughout the sequence all common components among the clips are successfully reassigned the same label number.

Figure 3 provides the tracking results using the GMM method. To have comparable results with VBGMM, any component for which its total volume was <5 % the clip volume was merged into its nearest component using the *KL* distance. Clearly, GMM detects more components than VBGMM, which may be considered as over-segmentation. Moreover, the label correspondence among the clips is not robust. For example, in the first 5 clips, where there are small

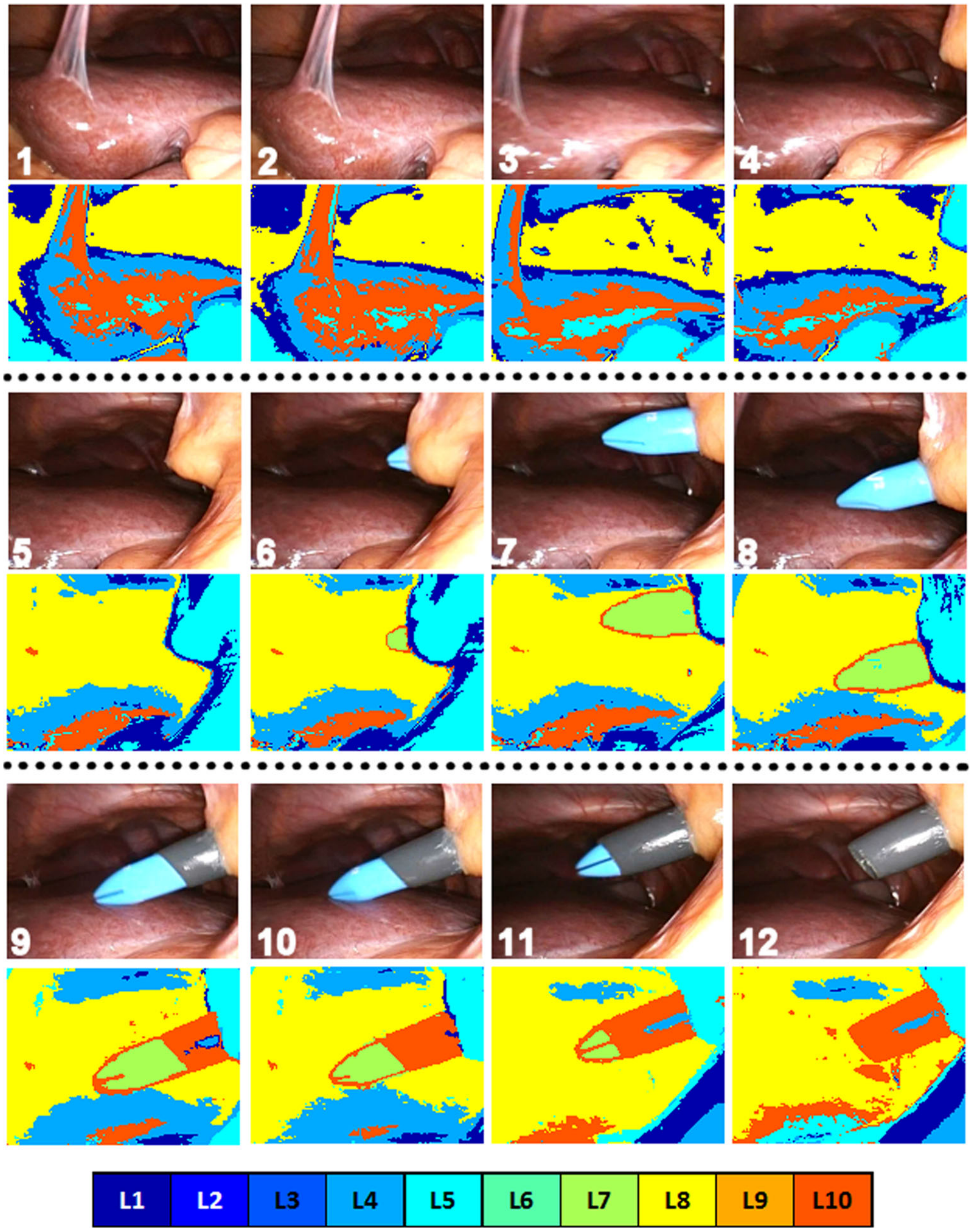

Fig. 2 First frames and the corresponding component classification output, from a series of laparoscopic video clips processed by the VBGMM algorithm. Label *color coding* is shown at the *bottom*

changes in the scene, GMM fails to track the same components. For example, compare L7 among clips 1–3. The components of the trocar are successfully tracked, but the surrounding regions are mistracked (see for example L1, L8 between clip 10 and clip 11).

To further demonstrate the value of VBGMM, Fig. 4a, b shows the total number of significant components in the sequence and the mixture weights for the first clip, respectively. As previously mentioned, when the obturator appears in the scene, VBGMM assigns a new label, and hence, the number of labels is increased to 6. In clip 12 the obturator is removed and so its label. For GMM, the total number of labels varies greatly along the sequence. In particular for clip 6, the entry of the new component seems to negatively affect label tracking, resulting in 2 less labels with respect to the previous clip.

As shown in Fig. 4b, the VBGMM framework allows the usage of error bars in the mixture weights, while for GMM this is not valid. Moreover, using a Dirichlet prior for the weights leads to several insignificant components the weights of which are almost zero. In contrast, for GMM all 10 initial components are denoted as significant and their weights are

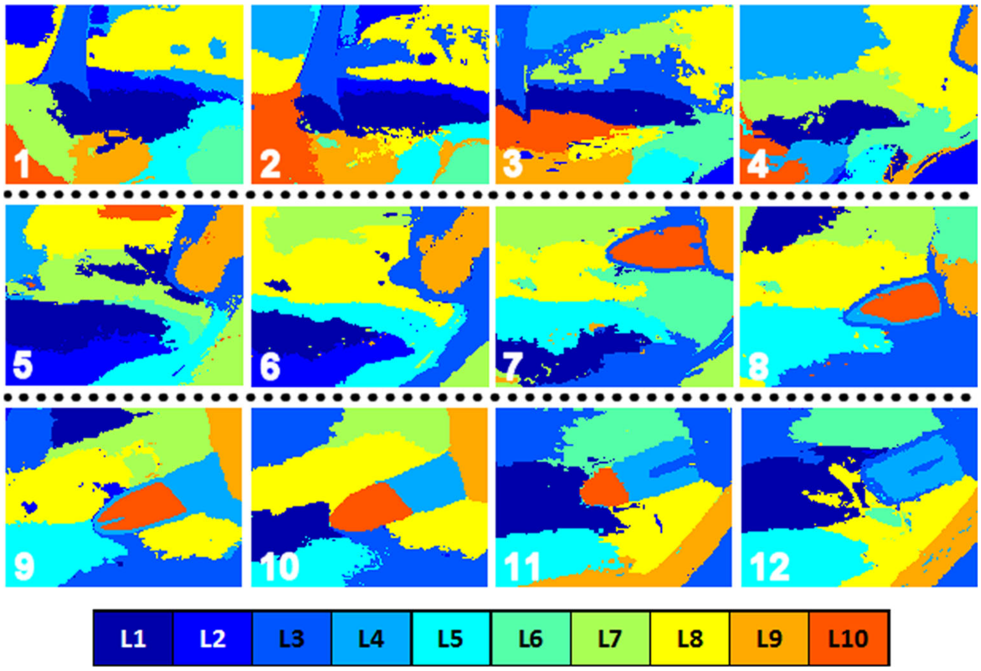

Fig. 3 GMM-based component classification output for the clips shown in Fig. 2

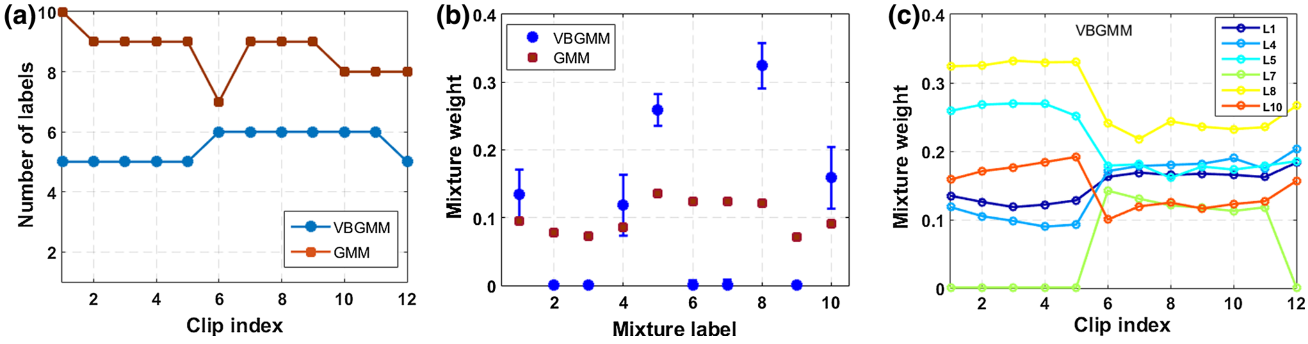

Fig. 4 **a** Total number of significant components generated by VBGMM and GMM for the clips shown in Fig. 2. **b** The corresponding mixture weights for the first clip. **c** The sequence of the mixture weights as found by the VBGMM algorithm; the *line colors* are in accordance with the *color coding* shown in Fig. 2

variable, although none is near zero. In Fig. 4c, we plot the sequence of the weights for the components that are considered significant. The components labeled as L1, L4, L5, L8 and L10 are tracked throughout the 12 clips. In clip 6, there is an extra component (obturator) with a significant weight, and it is assigned a previously unused label (L7). The weight of this component is significant up to clip 11. In clip 12, the algorithm finds 5 components that are matched with those in the previous clip. Label L7 is dropped since its weight did not find a close match with the previous components.

Shot border detection

Figure 5 shows the first frames from various clips extracted from a video segment processed with the VBGMM algorithm. The sequence of the total number of labels, along with the different shots found by our method, is shown at the bottom. As described previously, a border is defined when label tracking fails (clips 14 and 45). As can be seen from clips 5 and 10, during the first shot the surgeon attempts to lift the gallbladder. The content of the scene does not change much, which is captured by label tracking. In clip 14, the camera is pulled out abruptly, and consequently label tracking fails, signifying the end of the current shot. In shot 2, there is a variable change in the number of labels, but the surgical scene is similar as may be seen from clips 20 and 30. From clip 43, the camera starts moving to the right and tracking fails in clip 45 where a new border is defined. Note the difference between clips 20, 30 and 45 where the gallbladder is absent. The following frames (clips 50, 55) are repre-

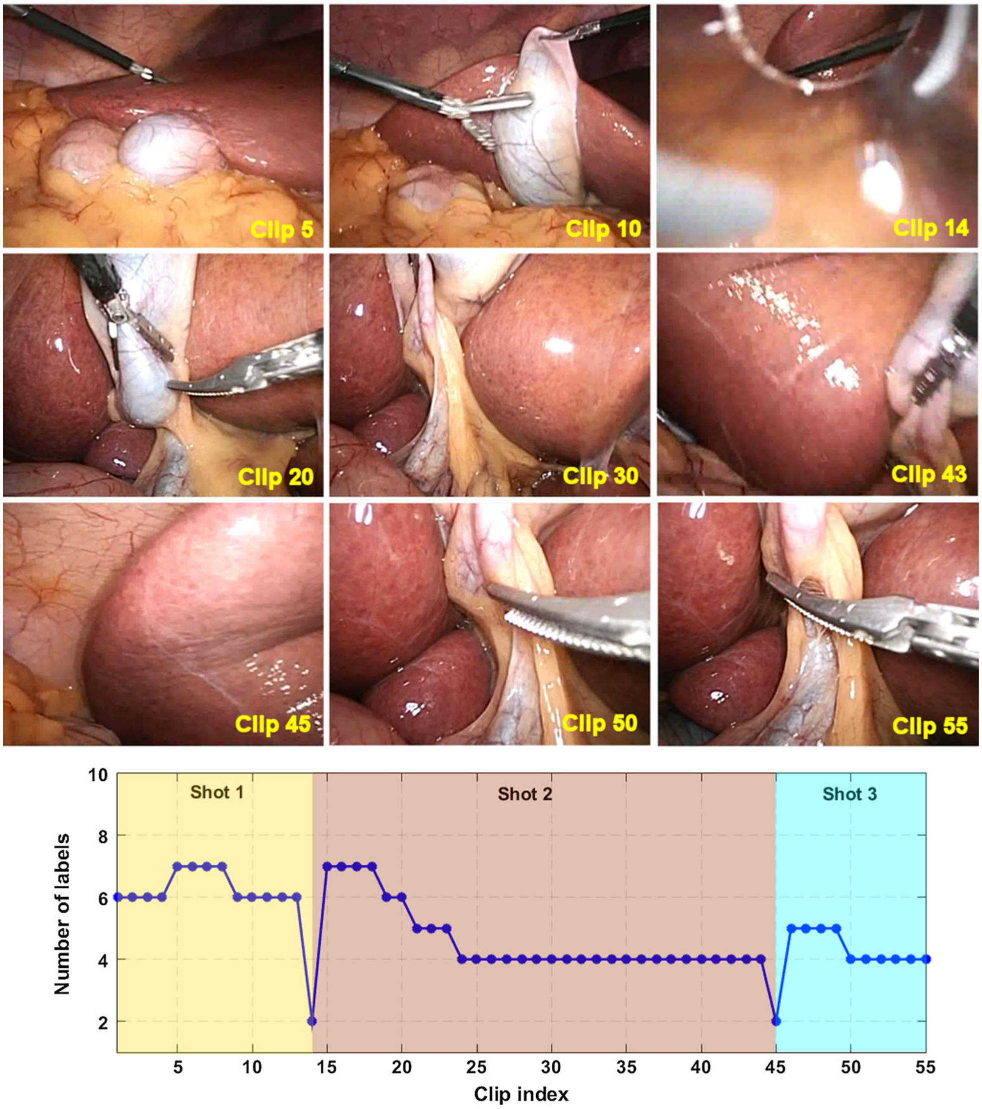

Fig. 5 First frames from various clips along a video segment. The *bottom panel* shows the number of the significant components estimated by the VBGMM algorithm and the corresponding shots defined by the proposed method

sentative of shot 3, where the surgeon starts dissecting the gallbladder.

Figure 6 shows another example from a lengthier video segment that is split into 3 shots. The two borders were found at clips 48 and 65. To obtain better understanding about the video content, for the two shot borders we present two consecutive frames (t and $t + \Delta t$) for the clip before the shot border, and the clip defined as the border. For the first one (clip 48), it is clear that the camera movement results in a completely different visual content, leading to a tracking failure. However, for the second border, the surgical scene does not change much, despite the maneuver of the surgeon to lift the gallbladder. In this case, the instrument movement seems to negatively affect label tracking, and hence, clip 65 is detected as a shot border. This is an example of a false hit since according to the reviewer the overall content of the scene did change much.

Figure 7 presents results from a video segment that consists of a single shot. The top row shows frames from various clips along the segment, and the next row shows the labels assigned by the tracking algorithm. At the bottom, there is the sequence with the total number of significant components found. The label images are the pure output of the VBGMM-based tracking algorithm (i.e., without postprocessing). Note that our purpose here is to highlight the output of label tracking for a lengthy video segment rather than presenting multilabel image classification results. As can be seen from the labeled images, there is a consistency in the component label matching throughout the sequence. For example, the background tissue (liver) is constantly assigned

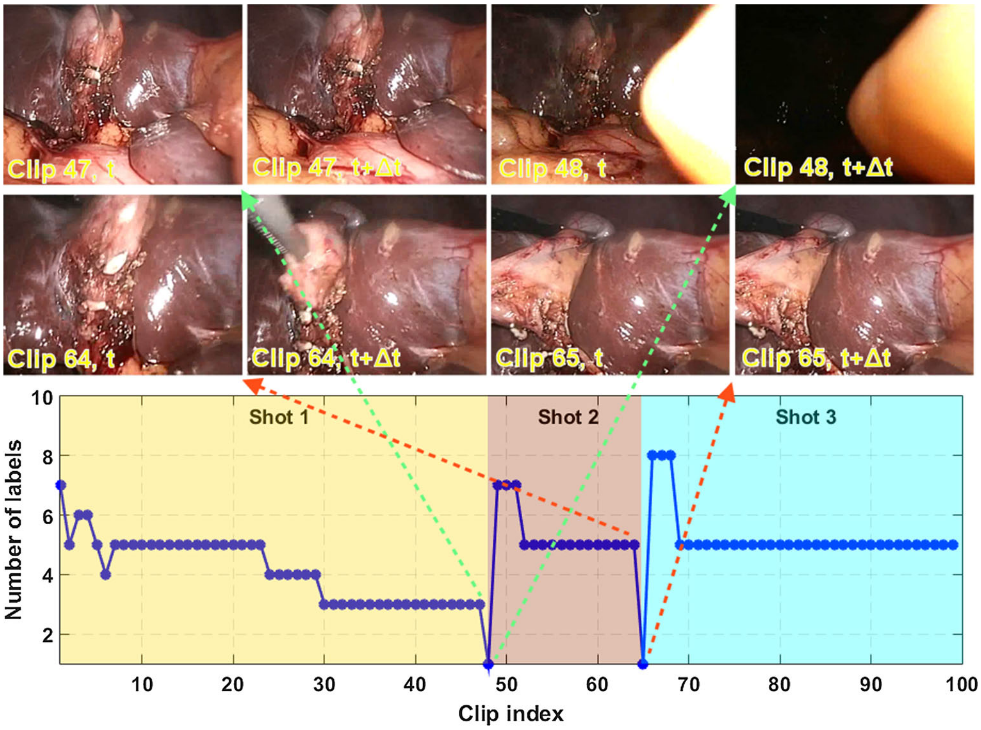

Fig. 6 Another example of shot detection. Between the shots, the first two frames from the clip before the border, and the actual border found, are shown

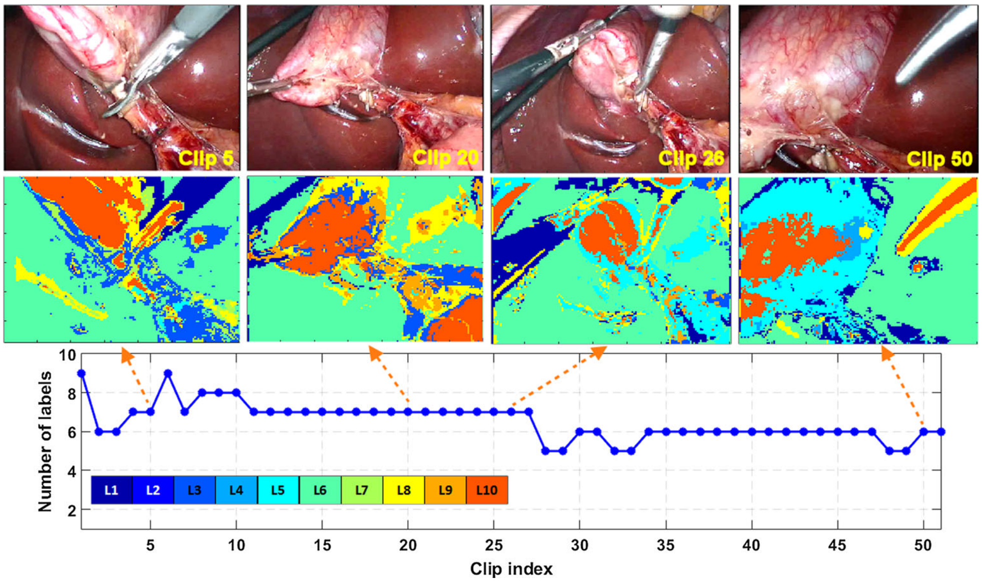

Fig. 7 An example of a continuous shot in a video segment. The *first panel* shows the first frames from various clips along the video sequence. The *second panel* shows the component classification output of the VBGMM algorithm. The *third panel* shows the total number of components found and the corresponding label color coding

label L6. The black-gray instrument shaft takes label L1, the silver-colored instrument tip is assigned label L8, and light- saturated components (for example the gallbladder or the tip when it is close to the camera) are assigned label L10. Other tissue structures are assigned labels L3 or L5. It should be emphasized here that the aforementioned description of the label correspondence is not exclusive since the proposed algorithm considers not only the color content but also optical flow.

Performance evaluation

Evaluation of shot border detection was based on two assessment schemes [3,12]. The first one was represented by mean precision and recall, calculated as:

$$\text{Pre} = 100\frac{\text{Sum}\left\{T_{\text{g}} \cap T_{\text{e}}\right\}}{\text{card}\{T_{\text{e}}\}}, \ \text{rec} = 100\frac{\text{Sum}\left\{T_{\text{g}} \cap T_{\text{e}}\right\}}{\text{card}\left\{T_{\text{g}}\right\}} \quad (17)$$

where T_{g} and T_{e} denote the timings of the estimated and the true (ground truth) shot borders in the video segment, respectively. In particular, we compute the percentage of the total number of intersections between the estimated and the true timings, with respect to the number of estimated timings, for precision, and to the number of true timings, for recall. An intersection was set to 1 if the estimated timing was found within a tolerance period centered on the true timing; otherwise, it was set to 0. For all methods, the tolerance period was set to 5 % of the length of the examined video segment.

The second scheme was based on the computation of coverage and overflow, which are related to the evaluation of the length of the estimated shots. In this context, coverage denotes the percentage of frames of an estimated shot that indeed correspond to the true shot. Overflow denotes the percentage of frames of an estimated shot that exceed the length of the true shot. Both metrics were computed with respect to the length of the true shots. For each video segment, an average value was computed, separately for each metric, for the number of true shots included in this segment. Coverage takes values 0–100 %. Overflow may vary from 0 up to >100 %, since the number of frames of the estimated shot that exceed the true shot may be more than the length of the true shot.

Table 1 summarizes the evaluation results for VBGMM, GMM and ‘MotionDiff’ which is described in [3]. The later method employs spatiotemporal differences of motion and is based on sequential frame processing, so to have comparable results the estimated boundary was matched to the nearest clip. The results demonstrate that VBGMM has higher performance than all other methods for most assessment metrics. Precision and recall measures are >80 % whereas the other two methods had lower performance. Higher recall than precision implies that the algorithm is better in generating true negatives than true positives, and vice versa. Our results show that VBGMM generates similar precision and recall values, whereas GMM has very low precision which means that a large number of false borders is generated. MotionDiff generates higher recall than precision although its performance was higher than GMM but lower than VBGMM.

Table 1 Shot boundary detection results (% average ± SD)

	Precision	Recall	Coverage	Overflow
VBGMM	83 ± 9	85 ± 7	84 ± 12	37 ± 13
GMM	54 ± 12	76 ± 9	54 ± 9	89 ± 11
MotionDiff	70 ± 11	77 ± 8	58 ± 9	27 ± 6

With regard to coverage and overflow, the shots generated by VBGMM seem to coincide with the true ones by 84 %. The estimated shots had a (false) extra length of about 37 % than that of the true shots. For GMM, the corresponding results are much lower, whereas MotionDiff seems to generate less false positive shot clips than VBGMM (37 vs. 27 %), although the true positive shot clips were less than that of VBGMM (84 vs. 58 %).

Discussion

Our results showed that VBGMM has a consistent performance of >80 % in three of the four assessment metrics (precision, recall and coverage). Provided that in general terms, precision–recall evaluates the number of borders, whereas coverage–overflow evaluates the length of the shots, we can conclude that the proposed method generates shot sequences that are in a good agreement with the external observers’ views. If one could visualize the video sequence as a timeline arrow with vertical bars corresponding to the shot borders (e.g., see Fig. 1), the generated shot sequences coincide (using a tolerance period) with the ground truth ones by ≈80 % in terms of the number of bars and the intersection of the length of the consecutive bars. However, our method has a tendency to overestimate the length of the true shots on average by 37 %. In other words, it may detect the border(s) of a shot quite a few frames beyond the true starting/ending border. Provided that in our method the estimation of the shot border is heavily related to the number of significant components tracked along the sequence, there may be multiple factors contributing to this limitation, such as the number of the past components considered, the spatiotemporal downsampling ratio, the length of the analyzed video clips. Fine-tuning of these parameters may restrict this overflow result.

The selection of the appropriate shot border was based on the failure of the label tracking process. The idea of tracking failure to signify the start/end of an activity is not new and has been employed previously in surgery for various reasons such as surgical navigation [29] and content-based surgical scene representation [11]. However, these works rather present evaluation of novel feature descriptors for Kalman-based tracking in short video segments, with no reference to specific applications such as shot detection. Our method employs simple color and motion features, although more advanced descriptors such as those used in the aforementioned works can also be employed, though at a higher computational cost.

The main idea lies in the application of the VBGMM framework for clip segmentation into spatiotemporal components and the tracking algorithm that employs the posteriors of the previous clip as posteriors for the next processing step. Based on this idea, our results showed that label component correspondence is treated effectively.

Compared to the work presented in [3], our method provides superior results for most assessment metrics. In contrast to shot detection based only on motion differences, VBGMM-based tracking analyzes also the color content of the spatiotemporal volume. Moreover, a border is defined when there are significant changes not only in terms of motion but also in the visual content of the volume examined, which results in smoother results. MotionDiff simply monitors motion changes in subsequent frames, resulting in oversegmentation of the video clip sequence.

The proposed methodology does not employ future data to perform border detection since all clips from a given video are analyzed in a sequential order. This is a notable characteristic that allows the exploitation of additional applications such as the segmentation of an operation into its main workflow phases in real time. Potential benefits include the management of a surgical process and the study of surgical skills, as reported in relevant works [18,30]. However, using a modern computer the VBGMM algorithm takes ≈120 s to analyze a downsampled clip volume consisting of ≈0.5 million pixels, for a maximum number of 20 iterations. Based on this limit, the potential for real-time application is prohibitive, although with faster implementations based on parallel core processing and the exponentially increasing computational power, the speed can be increased.

Although the main focus of this work was to present a proof of concept methodology for shot detection, a potential drawback is that it was not tested to detect irrelevant scenes, usually encountered when the camera is removed from the patient's body. Segmentation of these scenes is desirable due to the waste of storage capacity and the resulting difficulty in retrieving relevant parts. However, based on the current design, it is expectable that the proposed method could easily detect the onset of these scenes since the content between in- and outpatient video clips presents significant changes. Alternatively, one could add a preprocessing that takes into account ad hoc color analysis techniques such as those reported in previous works [14,23].

Conclusions

In this paper, we have presented a method for grouping consecutive video clips into shots, based on spatiotemporal changes occurred in the timeline of surgery. The core idea was based on the application of the VBGMM algorithm for clip segmentation into components that share similar color–motion characteristics and then track these components along the clip sequence of the video. An important advantage of the variational approach is that by employing Dirichlet priors for the GMM weights, the problem of model-order selection is handled effectively. Hence, based on an arbitrary initial setting, only those components that have significant weights are selected for further processing. In contrary, GMM is unable to provide the proper number of components since the segmentation is based solely on the initial setting.

In the future, we aim to investigate two additional applications of the proposed method. The first one will target grouping of the segmented shots without considering their temporal order. As it is now, the algorithm segments the video sequentially, without considering similarity with previous or subsequent shots. Post-processing for identifying similar shots irrespective to their time stamp may provide a valuable tool for applications related to video summarization. Second, keyframe extraction could be investigated by analyzing further the generated shots and extract keyframes (or key clips) from the plateaus of the label sequence generated by the tracking algorithm. For example, the most correlated keyframe could be selected, since the video content is mostly uniform for the clips in the plateau of the estimated label sequence. We expect that the investigation of these technological challenges will unfold new pathways in the computer-based understanding of surgical interventions.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest in relation to this article.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Statement of informed consent was not applicable since the manuscript does not contain any patient data.

References

1. Abdelsattar JM, Pandian TK, Finnesgard EJ, El Khatib MM, Rowse PG, Buckarma EH, Gas BL, Heller SF, Farley DR (2015) Do you see what I see? How we use video as an adjunct to general surgery resident education. J Surg Educ 72:e145–e150. doi:10.1016/j.jsurg.2015.07.012
2. Zevin B, Bonrath EM, Aggarwal R, Dedy NJ, Ahmed N, Grantcharov TP (2013) Development, feasibility, validity, and reliability of a scale for objective assessment of operative performance in laparoscopic gastric bypass surgery. J Am Coll Surg 216:955–965.e8; quiz 1029–31, 1033. doi:10.1016/j.jamcollsurg.2013.01.003
3. Primus MJ, Schoeffmann K, Böszörmenyi L (2013) Segmentation of recorded endoscopic videos by detecting significant motion changes. In: 11th International work. Content-Based Multimed. Index. Veszprem, Hungary, pp 223–228
4. Priya R, Shanmugam TN (2013) A comprehensive review of significant researches on content based indexing and retrieval of

visual information. Front Comput Sci 7:782–799. doi:10.1007/s11704-013-1276-6

5. Gao X, Li J, Shi Y (2006) A video shot boundary detection algorithm based on feature tracking. Lect Notes Comput Sci 4062:651–658
6. Hu W, Xie N, Li L, Zeng X, Maybank S (2011) A survey on visual content-based video indexing and retrieval. IEEE Trans Syst Man Cybern C Appl Rev 41:797–819. doi:10.1109/TSMCC.2011.2109710
7. Cotsaces C, Nikolaidis N, Pitas I (2006) Video shot detection and condensed representation. A review. IEEE Signal Process Mag 23:28–37. doi:10.1109/MSP.2006.1621446
8. Jacobs A, Miene A, Ioannidis GT, Herzog O (2004) Automatic shot boundary detection combining color, edge, and motion features of adjacent frames. In: TRECVID 2004 Work. Notebook papers, pp 197–206
9. Hanjalic A (2002) Shot-boundary detection: unraveled and resolved? IEEE Trans Circuits Syst Video Technol 12:90–105. doi:10.1109/76.988656
10. Del Fabro M, Böszörmenyi L (2013) State-of-the-art and future challenges in video scene detection: a survey. Multimed Syst 19:427–454. doi:10.1007/s00530-013-0306-4
11. Giannarou S, Yang G (2010) Content-based surgical workflow representation using probabilistic motion modeling. Lect Notes Comput Sci 6326:314–323
12. Twinanda AP, De Mathelin M, Padoy N (2014) Fisher kernel based task boundary retrieval in laparoscopic database with single video query. In: Golland P, Hata N, Barillot C, Hornegger J, Howe R (eds) Medical image computing and computer-assisted intervention–MICCAI 2014, Springer International Publishing, Switzerland, pp 409–416
13. Twinanda AP, Marescaux J, de Mathelin M, Padoy N (2015) Classification approach for automatic laparoscopic video database organization. Int J Comput Assist Radiol Surg 10:1449–1460. doi:10.1007/s11548-015-1183-4
14. Munzer B, Schoeffmann K, Boszormenyi L (2013) Relevance segmentation of laparoscopic videos. In: IEEE international symposium on multimedia, IEEE, Anaheim, CA, USA, pp 84–91
15. Schoeffmann K, Del Fabro M, Szkaliczki T, Böszörmenyi L, Keckstein J (2014) Keyframe extraction in endoscopic video. Multimed Tools Appl 74:11187–11206. doi:10.1007/s11042-014-2224-7
16. Loukas C, Georgiou E (2015) Performance comparison of various feature detector-descriptors and temporal models for video-based assessment of laparoscopic skills. Int J Med Robot Comput Assist Surg. doi:10.1002/rcs.1702
17. Beecks C, Schoeffmann K, Lux M, Uysal MS, Seidl T (2015) Endoscopic video retrieval: a signature-based approach for linking endoscopic images with video segments. In: Del Bimbo A, Chen S-C, Wang H, Yu H, Zimmermann R (eds) IEEE Proceedings of international symposium on multimedia Miami, FL, USA, pp 1–6
18. Blum T, Feussner H, Navab N (2010) Modeling and segmentation of surgical workflow from laparoscopic video. Lect Notes Comput Sci 6363:400–407
19. Loukas C, Georgiou E (2015) Smoke detection in endoscopic surgery videos: a first step towards retrieval of semantic events. Int J Med Robot Comput Assist Surg 11:80–94. doi:10.1002/rcs.1578
20. Lux M, Marques O, Schöffmann K, Böszörmenyi L, Lajtai G (2009) A novel tool for summarization of arthroscopic videos. Multimed Tools Appl 46:521–544. doi:10.1007/s11042-009-0353-1
21. Cao Y, Tavanapong W, Li D (2004) A visual model approach for parsing colonoscopy videos. Lect Notes Comput Sci 3115:160–169. doi:10.1007/978-3-540-27814-6_22
22. Kwitt R, Vasconcelos N, Rasiwasia N, Uhl a, Davis B, Häfner M, Wrba F (2012) Endoscopic image analysis in semantic space. Med Image Anal 16:1415–1422. doi:10.1016/j.media.2012.04.010
23. Oh J, Hwang S, Lee J, Tavanapong W, Wong J, de Groen PC (2007) Informative frame classification for endoscopy video. Med Image Anal 11:110–127. doi:10.1016/j.media.2006.10.003
24. Lalys F, Bouget D, Riffaud L, Jannin P (2013) Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures. Int J Comput Assist Radiol Surg 8:39–49. doi:10.1007/s11548-012-0685-6
25. Hamerly G, Elkan C (2003) Learning the K in K-Means. In: Thrun S, Saul LK, Schölkopf B (eds) Adv. Neural Inf. Process. Syst. MIT press, Whistler BC, Canada, pp 281–288
26. Attias H (2000) A variational Bayesian framework for graphical models. In: Advanced neural information processing system. Neural Information Processing Systems Foundation, pp 209–215
27. Corduneanu A, Bishop CM (2001) Variational Bayesian model selection for mixture distributions. In: Proceedings of 8th international conference on AI statistics. Key West, FL, USA, pp 27–34
28. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York, pp 474–480
29. Giannarou S, Visentini-Scarzanella M, Yang GZ (2013) Probabilistic tracking of affine-invariant anisotropic regions. IEEE Trans Pattern Anal Mach Intell 35:130–143. doi:10.1109/TPAMI.2012.81
30. Loukas C, Georgiou E (2013) Surgical workflow analysis with Gaussian mixture multivariate autoregressive (GMMAR) models: a simulation study. Comput Aided Surg 18:47–62. doi:10.3109/10929088.2012.762944