Int J Comput Vis (2009) 84: 146–162
DOI 10.1007/s11263-008-0146-4

Cooperative Object Segmentation and Behavior Inference in Image Sequences

Laura Gui · Jean-Philippe Thiran · Nikos Paragios

Received: 4 October 2007 / Accepted: 27 May 2008 / Published online: 17 June 2008

Abstract In this paper, we propose a general framework for fusing bottom-up segmentation with top-down object behavior inference over an image sequence. This approach is beneficial for both tasks, since it enables them to cooperate so that knowledge relevant to each can aid in the resolution of the other, thus enhancing the final result. In particular, the behavior inference process offers dynamic probabilistic priors to guide segmentation. At the same time, segmentation supplies its results to the inference process, ensuring that they are consistent both with prior knowledge and with new image information. The prior models are learned from training data and they adapt dynamically, based on newly analyzed images. We demonstrate the effectiveness of our framework via particular implementations that we have employed in the resolution of two hand gesture recognition applications. Our experimental results illustrate the robustness of our joint approach to segmentation and behavior inference in challenging conditions involving complex backgrounds and occlusions of the target object.

Keywords Image segmentation · Behavior inference · Gesture recognition

L. Gui (✉) · J.-P. Thiran
Signal Processing Institute, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
e-mail: laura.gui@epfl.ch

J.-P. Thiran
e-mail: jp.thiran@epfl.ch

N. Paragios
Laboratoire MAS, Ecole Centrale de Paris, Chatenay-Malabry, France
e-mail: nikos.paragios@ecp.fr

1 Introduction

In the classical computer vision paradigm, image segmentation and object behavior inference lie at different levels of abstraction. At a basic level, segmentation aims at extracting relevant objects from the target image(s). A higher level image understanding task is to infer the behavior of the extracted object(s), based on prior knowledge about typical object behavior. By "behavior", we mean the temporal evolution of the object, as observed in the image sequence. The inference of object behavior from an image sequence requires the determination of the appropriate behavior class for each object evolution instance throughout the sequence. For instance, one may want to make an inference about a sequence of object motions (e.g., car turn directions at an intersection), motions and deformations (e.g., hand gestures, body motions), or about a sequence of intensity changes in a brain activation map for diagnostic purposes. Generally, such an inference is formulated in terms of a set of relevant attributes (e.g., color histogram, object position, orientation, shape, size, etc.), which have been extracted from the image sequence in a preceding phase. Thus, attribute extraction is conventionally performed separately from behavior inference.

This paper pursues a joint solution to the problems of image segmentation and object behavior inference. Clearly, a precise segmentation of the target object would greatly facilitate behavior inference by offering access to object attributes relevant to the inference task. Moreover, image segmentation could be drastically improved by exploiting the knowledge which is available to the behavior inference task. This knowledge is typically represented in the form of probabilistic attribute models corresponding to behavior classes. Such models can be used to guide the segmentation of the

target object(s) in challenging conditions (e.g., images affected by noise, occlusions or cluttered background).

These considerations motivate us to introduce a general framework for cooperative object segmentation and behavior inference in image sequences. We formulate the segmentation in a variational setting, which enables the smooth integration of both prior knowledge (in the form of behavior class models) and specific segmentation criteria for the target images. This paper reviews our general framework formulation (Gui et al. 2007a, 2007b), and further develops it in order to deal with more complex behavior scenarios, as demonstrated in a new application pertaining to sign language recognition.

Variational methods offer a solid mathematical basis for the formulation and solution of many computer vision problems. In particular, the image segmentation problem has been formulated in terms of energy minimization, allowing the seamless blending of various criteria describing the desired solution, such as smoothness, region homogeneity, edge correspondence, etc. Starting with the original active contour model (Kass et al. 1987), variational segmentation has been steadily advancing through the introduction of the Mumford-Shah model (Mumford and Shah 1989), the level set approach (Osher and Sethian 1988), geodesic active contours (Caselles et al. 1995; Kichenassamy et al. 1995; Malladi et al. 1995) and, more recently, versatile segmentation approaches such as Vese and Chan (2002), Paragios and Deriche (2002). The segmentation of familiarly shaped objects in difficult cases was facilitated by the introduction of statistical shape priors into active contours (Cootes et al. 1999), into level set active contours (Leventon et al. 2000; Chen et al. 2002; Rousson and Paragios 2002) and in the Mumford-Shah model (Cremers et al. 2006c; Bresson et al. 2006). Variational methods have also been adapted to object tracking (e.g., Kass et al. 1987; Paragios and Deriche 2005; Cremers et al. 2006b). The coherence between frames has been exploited by approaches based on Kalman filtering (Terzopoulos and Szeliski 1992), particle filtering (Rathi et al. 2007), and autoregressive models (Cremers 2006).

Our framework fuses segmentation and behavior inference over image sequences. To our knowledge, this idea is novel in the context of variational image sequence analysis, and it capitalizes on existing developments in the use of shape priors. In previous works, segmentation has been combined with object recognition, yielding good results in the case of single, static images, both in variational (Cremers et al. 2006c) and non-variational (Tu et al. 2003; Leibe et al. 2004; Ferrari et al. 2004; Kokkinos and Maragos 2005) settings. For tracking, (Cremers 2006) demonstrates the use of single-class dynamic models of motion and deformation, based on auto-regressive modeling. For image registration, (Cremers et al. 2006a) dynamically chooses the relevant modes of an a-priori joint intensity distribution of registered image pairs, according to their proximity to the current estimated distribution. The novelty of our work is that we address the segmentation problem *over image sequences*, in a *multi-class* scenario, i.e., where the behavior class of the tracked object changes over time. Via a parallel classification strategy, we guide the segmentation dynamically towards the most likely behavior class at the given time. This guidance is based on learning (via Hidden Markov Models) and on accumulated evidence throughout the image sequence. Moreover, these dynamical probabilistic priors offered by classification evolve during the segmentation of each image, adapting to new image content.

Our *general framework* for the cooperative resolution of the two tasks, segmentation and behavior inference, can be employed to resolve a wide range of applications by adapting its components and parameters according to the specific need. In particular, we illustrate the potential of our approach in two gesture recognition applications, where the cooperation of segmentation and behavior inference dramatically increases the tolerance to occlusion and background complexity present in the input image sequences.

The remainder of the paper is organized as follows. Section 2 details the collaborating halves of our general framework: behavior inference and segmentation. In Sects. 3 and 4 we propose particular implementations of our framework, for the resolution of two gesture recognition applications. Experimental results are presented at the end of Sects. 3 and 4, respectively. Section 5 concludes the paper.

2 Formulation of the General Framework

Our general framework for fusing segmentation and behavior inference is based on the idea of cooperation between the two processes along the target image sequence (Fig. 1). During an initial training phase, the inference process learns the dynamic probability models of typical behaviors from training data. Then, segmentation and behavior inference

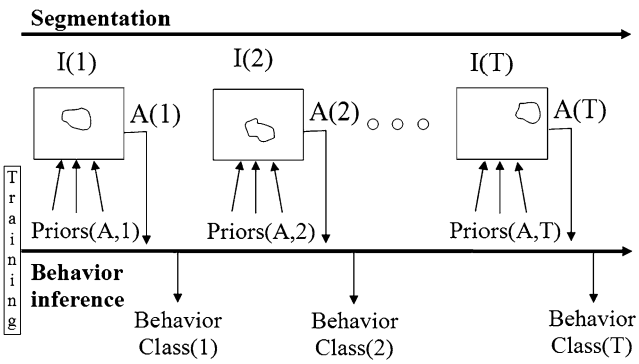


Fig. 1 Our approach: cooperation of segmentation and behavior inference along the image sequence $I(1..T)$

are run cooperatively throughout a new test image sequence. For each image, an inference step is performed, generating probabilistic prior attribute models for each behavior class. These are used by the ensuing segmentation to identify the most likely objects in the current image and subsequently provide their attributes to the next inference step. The priors offered by the inference process are based on learning from training data and are updated dynamically according to newly processed images. The most likely behavior class of each object evolution instance can be extracted at any point within the sequence from the inference process.

By the generic term "attribute" we designate a visual property of an object, definable as a functional $A(C, I)$ of the image I and of the object's segmenting contour C (A is assumed to be differentiable with respect to C). This definition includes many properties computable with boundary- and region-based functionals (e.g. position, orientation, average intensity/color, higher order statistics describing texture). This makes our framework adaptable to the needs of other behavior recognition applications.

2.1 Behavior Inference and Its Cooperation with Image Segmentation

Given a sequence of object attribute values extracted from an image sequence, behavior inference translates to finding the best matching sequence of behavior classes. We address this task using Hidden Markov Models (HMMs) (Rabiner 1989). Having estimated HMM parameters from training attribute sequences, we use them to infer the behavior reflected in new image sequences. Jointly, we segment these sequences, according to the intended collaboration.

An HMM (Rabiner 1989) is a doubly embedded stochastic process. It consists of an underlying hidden process, observable via a set of stochastic processes (the HMM states) that produce a sequence of observations. In our case, the observations are the attribute values extracted from the image sequence, while the states correspond to the behavior classes. In our framework, HMMs model the dynamics of behavior by encapsulating the most likely successions of basic actions (corresponding to behavior classes) which compose the behaviors under study. Depending on the complexity of the application and on the available prior knowledge regarding typical behavior, we can employ either one or several HMMs to capture behavior dynamics. We describe both cases in the following and illustrate their use in two different application scenarios in Sects. 3 and 4.

We briefly introduce HMMs and their notation (for details see Rabiner 1989). We denote the HMM states by $S = \{S_1, S_2, \ldots, S_M\}$, the state at time t by q_t and the attribute value at time t by $A(t)$. The HMM parameters are:

1. the initial state distribution $\pi = \{\pi_i\}$, with $\pi_i = P(q_1 = S_i), i = 1..M$,
2. the state transition probability distribution $T = \{t_{ij}\}$, with $t_{ij} = P(q_{t+1} = S_j | q_t = S_i), i, j = 1..M$, and
3. the state observation probability distributions (behavior class likelihoods):

$$P(A(t) \mid q_t = S_i) = P_i(A(t)), \quad i = 1..M. \tag{1}$$

The class likelihoods $P_i(A(t))$ are another free parameter of our framework. They can be adapted to the application at hand, subject only to the condition that they be differentiable with respect to $A(t)$.

2.1.1 Behavior Modeling with One HMM

In simple application cases with few and relatively well differentiated behavior classes, the use of a single HMM is sufficient to model behavior dynamics and perform inference about object behavior. During the training phase, the ensemble of HMM parameters are estimated from typical behavior-class-labeled attribute sequences (see Rabiner 1989 for details). In this way, the HMM encodes constraints regarding preferred successions of behavior classes, as well as typical values of object attributes corresponding to the different behavior classes.

Having estimated the HMM parameters from training data, we can perform behavior inference on new attribute sequences using the Viterbi algorithm (Rabiner 1989). For a new observation sequence $A_{1..T} = \{A(1), A(2), \ldots, A(T)\}$, the algorithm estimates the most likely state (behavior class) sequence $q_{1..T}^{\text{opt}} = \{q_1, q_2, \ldots, q_T\}^{\text{opt}}$ that generated it, as follows:

$$q_{1..T}^{\text{opt}} = \arg\max_{q_{1..T}} P(q_{1..T} | A_{1..T}) = \arg\max_{q_{1..T}} P(q_{1..T}, A_{1..T}). \tag{2}$$

This estimation is equivalent to the evaluation—for each time step t and for each state S_i—of the quantity

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(q_{1..t-1}, q_t = S_i, A_{1..t}). \tag{3}$$

It represents the highest probability at time t, along a state sequence which accounts for the first t observations and ends in state S_i. After initialization

$$\delta_1(i) = \pi_i P_i(A(1)), \quad i = 1..M, \tag{4}$$

the following recursion is used to compute the δ-s at each time step t:

$$\delta_t(i) = \left(\max_{j=1..M} \delta_{t-1}(j) t_{ji} \right) P_i(A(t)), \quad i = 1..M. \tag{5}$$

For each $\delta_t(i)$, the state which maximizes (5) is stored in a variable $\psi_t(i)$

$$\psi_t(i) = \arg\max_{j=1..M} \delta_{t-1}(j) t_{ji}, \quad i = 1..M, \tag{6}$$

initialized by

$$\psi_1(i) = 0, \quad i = 1..M. \tag{7}$$

The ensemble of δ and ψ variables can be used at any time instance T to retrieve the (currently) optimal state sequence by backtracking:

$$\begin{aligned} q_T^{\text{opt}} &= \arg\max_{i=1..M} \delta_T(i), \\ q_t^{\text{opt}} &= \psi_{t+1}(q_{t+1}^{\text{opt}}), \quad t = T-1, T-2, \ldots, 1. \end{aligned} \tag{8}$$

The probability of this optimal state sequence is given by

$$P^{\text{opt}} = \max_{i=1..M} \delta_T(i). \tag{9}$$

We couple behavior inference and segmentation by using the probability estimates of the Viterbi algorithm at each step to guide the segmentation of the corresponding image. To this end, we run the algorithm and segmentation in an interleaved manner along the image sequence, using as observations the attributes of newly segmented images as soon as they become available. Suppose that we have completed step $t-1$ of our framework, so that $A_{1..t-1}$ and $\delta_{t-1}(j)$, $j = 1..M$ are available. To guide the segmentation of $I(t)$, we use the maximum amount of a priori knowledge offered by the inference process:

1. the predictions of each class i for the next attribute $A(t)$; i.e., the likelihood functions $P_i(A(t))$, $i = 1..M$ (1), and
2. our relative confidence in the prediction of each class i, given by the Viterbi algorithm, i.e., the maximum probability of reaching state S_i at time step t, after having observed attributes $A_{1..t-1}$:

$$\begin{aligned} w_t(i) &= \max_{j=1..M} \delta_{t-1}(j) t_{ji} \\ &= \max_{q_1, q_2, \ldots, q_{t-1}} P(q_{1..t-1}, q_t = S_i, A_{1..t-1}). \end{aligned} \tag{10}$$

We define the prior information offered by class i about the next attribute $A(t)$ as the product of the two quantities above. According to (5), this is

$$\delta_t(A(t), i) = w_t(i) P_i(A(t)), \quad i = 1..M; \tag{11}$$

i.e., δ_t as a function of the unknown attribute $A(t)$. In Sect. 2.2 we explain how we introduce these class contributions into the segmentation framework.

2.1.2 Behavior Modeling with Multiple HMMs

Many practical applications require analysis of complex behavior scenarios, involving numerous classes, often poorly discriminated in terms of the available attributes. In such cases, the behavior inference process can be greatly aided by imposing coherence conditions on the resulting succession of behavior classes, stemming from prior knowledge about possible behaviors. Common scenarios in human-to-computer interaction applications require discrimination among a number of behavior types. In this context, each behavior is made up of a different succession of basic actions, belonging to different behavior classes, which are shared among the behavior types (e.g. letter classes shared among words). We model such cases via multiple HMMs. Each HMM accounts for a different behavior type and they all share the same state models, corresponding to the basic behavior classes. To perform behavior inference, we estimate the probability of an attribute sequence on the most likely state path in each HMM. Then, we choose the winner HMM (thus, behavior type) as the one with the highest probability for the given attribute sequence. Its most likely state path yields the most likely succession of behavior classes for the given attribute sequence.

Our framework can be easily adapted to incorporate multiple HMMs. To distinguish among K behavior types, we employ K HMMs. These models share their states and state models $P_i(A(t)), i = 1..M$, while having different initial $\pi^k = \{\pi_i^k\}$ and state transition probabilities $T^k = \{t_{ij}^k\}, k = 1..K$. Running the Viterbi algorithm in parallel for all HMMs requires the use of K sets of variables δ and ψ, denoted by δ^k and $\psi^k, k = 1..K$, respectively. Similarly to the single HMM case, the analysis of a sequence $A_{1..T}$ starts with variable initialization:

$$\begin{aligned} &\delta_1^k(i) = \pi_i^k P_i(A(1)), \\ &\psi_1^k(i) = 0, \quad i = 1..M, k = 1..K. \end{aligned} \tag{12}$$

For each time step $t = 2..T$, a recursion step is performed:

$$\begin{aligned} &\delta_t^k(i) = \left(\max_{j=1..M} \delta_{t-1}^k(j) t_{ji}^k \right) P_i(A(t)), \\ &\psi_t^k(i) = \arg\max_{j=1..M} \delta_{t-1}^k(j) t_{ji}^k, \quad i = 1..M, k = 1..K. \end{aligned} \tag{13}$$

Finally, the probability of the attribute sequence given the most likely path in each HMM k is given by:

$$P_k^{\text{opt}} = \max_{i=1..M} \delta_T^k(i). \tag{14}$$

The winner HMM (thus, behavior type) maximizes the probability P_k^{opt}:

$$k^{\text{opt}} = \arg\max_{k=1..K} P_k^{\text{opt}}. \tag{15}$$

The most likely state (behavior class) sequence, corresponding to the given attribute sequence, can be retrieved by backtracking from the δ-s and ψ-s of the winner HMM:

$$\begin{aligned} q_T^{\text{opt}} &= \arg\max_{i=1..M} \delta_T^{k^{\text{opt}}}(i), \\ q_t^{\text{opt}} &= \psi_{t+1}^{k^{\text{opt}}}(q_{t+1}^{\text{opt}}), \quad t = T-1, T-2, \ldots, 1. \end{aligned} \tag{16}$$

Similarly to the single HMM case, we run segmentation and behavior inference cooperatively. After step $t-1$, attributes $A_{1..t-1}$ and variables $\delta^k_{t-1}(j)$, $j=1..M, k=1..K$ are available. Thus, we can guide the segmentation of the next image $I(t)$ (and thus the extraction of the next attribute $A(t)$) by using the two information sources available:

1. the class predictions $P_i(A(t)), i=1..M$ (1), and
2. our relative confidence in the class predictions, given by the maximum probability of reaching state S_i at time step t, after having observed attributes $A_{1..t-1}$. In the multiple HMM case, this probability can be estimated as:

$$w_t(i) = \max_{k=1..K}\left(\max_{q_1,q_2,\ldots,q_{t-1}} P(q_{1..t-1}, q_t = S_i, A_{1..t-1}|k)\right) = \max_{\substack{j=1..M\\k=1..K}} \delta^k_{t-1}(j)t^k_{ji}. \quad (17)$$

For notation correspondence with the single HMM case, we denote by $\delta_t(A(t), i)$ the prior information offered by class i about the next attribute $A(t)$ and we define it as the product of the two quantities above:

$$\delta_t(A(t), i) = w_t(i)P_i(A(t)) \quad i=1..M = \max_{k=1..K}\delta^k_t(A(t), i). \quad (18)$$

2.2 Image Segmentation and Its Cooperation with Behavior Inference

Motivated by successful prior knowledge-based segmentation approaches (Rousson and Paragios 2002; Cremers et al. 2006c), we introduce a variational framework for segmentation which incorporates the probabilistic behavior class priors $\delta_t(A(t), i)$ via a competition approach. In this way, the segmented object belongs to the class which best accounts for its generation, given the image evidence. Given the outcome of the joint segmentation/behavior inference for the first $t-1$ frames of an image sequence, we segment $I(t)$ by minimizing the following energy functional:

$$E(C, \mathcal{L}, I(t)) = E_{\text{data}}(C, I(t)) + \alpha E_{\text{prior}}(C, \mathcal{L}, I(t)), \quad (19)$$

where C is the segmenting contour, $\mathcal{L} = (L_1, \ldots L_M)$ is the set of labels (defined below) and α is a positive weighing constant. Energy $E_{\text{data}}(C, I(t))$ encodes image-related constraints on the contour C. It can include any boundary- or region-based segmentation terms suitable for the application at hand (e.g. Chan and Vese 2001). Energy $E_{\text{prior}}(C, \mathcal{L}, I(t))$ is:

$$E_{\text{prior}}(C, \mathcal{L}, I(t)) = -\sum_{i=1}^{M} \log\left(\delta_t(A(C, I(t)), i)\right)L_i^2 + \beta\left(1 - \sum_{i=1}^{M} L_i^2\right)^2. \quad (20)$$

The δ function has been defined in (11) for the single HMM case and in (18) for the multiple HMM case. This energy adds up the negative logarithms of the prior probabilities δ, which through energy minimization will lead to the maximization of the respective probabilities. Each prior carries a label factor L_i^2, which controls its contribution to segmentation according to its relative probability with respect to the other priors. The label L_i is a scalar variable that varies continuously between 0 and 1 during energy minimization and converges either to 1 (for the winning prior, whose probability has thus been maximized through segmentation) or to 0 (for the other priors, which have thus been annulled). Competition among priors is enforced by the constraint that the label factors should sum to 1, introduced by the term $\beta(1-\sum_{i=1}^M L_i^2)^2$ in energy (20). Here β is a Lagrange multiplier, updated at each energy minimization step to ensure that $(1-\sum_{i=1}^M L_i^2)^2 \approx 0$. A similar technique has been applied to a different problem in Zhao et al. (1996).

We minimize (19) simultaneously with respect to the segmenting contour C and the labels $\mathcal{L}$ using the calculus of variations and gradient descent. The contour C is driven by image forces (intensity, gradients, etc.) due to $E_{\text{data}}(C)$ and by the M attribute priors due to $E_{\text{prior}}(C, \mathcal{L})$:

$$\frac{\partial C}{\partial \tau} = -\frac{\partial E_{\text{data}}(C, I(t))}{\partial C} - \alpha\frac{\partial E_{\text{prior}}(C, \mathcal{L}, I(t))}{\partial C}. \quad (21)$$

Here τ is the artificial time of variable evolution and $\partial E_{\text{data}}(C, I(t))/\partial C$ can be derived through the calculus of variations for the particular chosen form of $E_{\text{data}}(C, I(t))$. The second term can be written as:

$$\frac{\partial E_{\text{prior}}(C, \mathcal{L}, I(t))}{\partial C} = -\sum_{i=1}^{M}\frac{L_i^2}{\delta_t(A(C, I(t)), i)}\frac{\partial \delta_t(A(C, I(t)), i)}{\partial A}\frac{\partial A(C, I(t))}{\partial C}, \quad (22)$$

where

$$\frac{\partial \delta_t(A(C, I(t)), i)}{\partial A} = w_t(i)\frac{\partial P_i(A(C, I(t)))}{\partial A}. \quad (23)$$

The derivatives $\partial P_i/\partial A$ and $\partial A(C, I(t))/\partial C$ are computed according to the particular likelihood function and attribute employed.

The evolution equations for the labels L_i are:

$$\frac{\partial L_i}{\partial \tau} = L_i\left(\log \delta_t(A(C, I(t)), i) + 2\beta\left(1 - \sum_{i=1}^{M} L_i^2\right)\right), \quad i=1..M. \quad (24)$$

The labels are initialized with equal values, so that $(1-\sum_{i=1}^M L_i^2)^2 \approx 0$. The update equation for the Lagrange

multiplier β is deduced by imposing constancy of the constraint over time: $d(1-\sum_{i=1}^{M} L_i^2)^2/d\tau = 0$. This yields the following update equation:

$$\beta = \frac{\sum_{i=1}^{M} L_i^2 \log \delta_t(A(C, I(t)), i)}{2\sum_{i=1}^{M} L_i^2 \left(\sum_{i=1}^{M} L_i^2 - 1\right)}. \tag{25}$$

From a probabilistic perspective, the minimization of our proposed energy using competing priors can be interpreted as the maximization of the probability $\delta_t(A(C, I(t)), i)$ with respect to both the attribute $A(C, I(t))$ and the class i, subject to image-based constraints imposed through the energy $E_{\text{data}}(C, I(t))$. Then the segmentation of image $I(t)$ can be viewed as the joint estimation of the attribute value $A^*(t)$ and the class i^* as:

$$(A^*(t), i^*) = \arg \max_{A(C,I(t)),i} \delta_t(A(C, I(t)), i), \quad \text{subject to image constraints via } E_{\text{data}}(C, I(t)). \tag{26}$$

Thus, segmentation works concurrently towards the same goal as behavior inference. It maximizes the joint probability of the behavior class and the observation at time t, while remaining consistent with previous observations (due to prior knowledge from the HMM(s)) and integrating new information from image $I(t)$.

2.3 Summary

To sum up, our framework for joint segmentation and behavior inference consists of the following:

– *Training phase:* estimate parameters of the HMM(s) from training attribute sequences, according to Rabiner (1989).
– *Testing phase:* perform joint segmentation and behavior inference on new attribute sequences $A_{1..T}$:

1. Segment first image in the sequence $I(1)$ (manually or using only the data term $E_{\text{data}}(C, I(1))$ in (19).
2. Extract attribute $A(1) = A(C, I(1))$.
3. Initialize δ and ψ functions according to (4), (7) for the single HMM case and (12) for the multiple HMM case.
4. For $t = 2..T$
   - Compute $w_t(i), i = 1..M$ according to (10) (single HMM) and (17) (multiple HMM).
   - Segment image $I(t)$ using energy (19), where the priors $\delta_t(A(C, I(t)), i)$ are given by (11) (single HMM) and (18) (multiple HMM).
   - Extract attribute $A(t) = A(C, I(t))$.
   - Compute $\delta_t(i)$ and $\psi_t(i), i = 1..M$ from (5) and (6) (single HMM) or compute $\delta_t^k(i)$ and $\psi_t^k(i), i = 1..M, k = 1..K$ from (13) (multiple HMM).
5. For the multiple HMM case, estimate winner HMM and thus infer behavior type using (15).
6. Backtrack to infer the behavior class of each attribute instance in $A_{1..T}$ using (8) for the single HMM and (16) for the multiple HMM.

3 Application to Finger-Counting Recognition

In the following, we demonstrate the potential of our general framework by implementing it for a finger-counting recognition application. First we describe the problem that we wish to address. Then, we detail a particular implementation of our general framework, including specific segmentation and probability models. We explain the estimation of HMM parameters from training data and finally we present test results of our implementation on new image sequences.

In our finger-counting application, we identify four gesture classes consisting of a right hand (facing the camera) going through four finger configurations: fist (Class 0), thumb extended (Class 1), thumb and index finger extended (Class 2) and thumb, index, and middle finger extended (Class 3). An example image of each gesture class is shown in Fig. 2.

Our typical gesture image sequences depict finger-counting from 1 to 3 (starting from the fist position) and from 3 to 1 (ending with the fist position), which is, in terms of gesture class successions, 0, 1, 2, 3 and 3, 2, 1, 0. Our aim is to perform joint segmentation and behavior inference of image sequences containing such successions; i.e., for each image, extract the segmenting contour of the hand and determine the gesture/behavior class to which it belongs.

3.1 Solution Using the Proposed Framework

For this application, the object attribute that we employ is the contour segmenting the hand $A(C, I) = C$. We represent the contour using a level set function (LSF) $\phi : \Omega \to \mathbb{R}$,

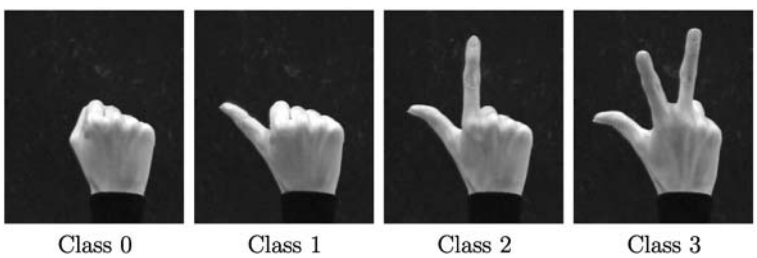


Fig. 2 Samples from the four gesture classes that we use in our finger-counting application

where Ω is the image domain (Osher and Sethian 1988). Function ϕ is chosen to be the signed distance function to the contour, so that $C \equiv \{(x, y) : \phi(x, y) = 0\}$.

Given the reduced number of behavior classes (four) and their relatively good discrimination in terms of the used attribute (the hand contour), we model behavior using a single HMM, as described in Sect. 2.1.1.

As data term in the segmentation energy (19), we use the piecewise constant Mumford-Shah model (Chan and Vese 2001):

$$\begin{aligned} E_{\text{data}}(\phi) &= E_{\text{MS}}(\phi), \\ E_{\text{MS}}(\phi) &= \iint_{\Omega} (I - \mu_+)^2 H(\phi) dxdy \\ &\quad + \iint_{\Omega} (I - \mu_-)^2 (1 - H(\phi)) dxdy \\ &\quad + \nu \iint_{\Omega} |\nabla H(\phi)| dxdy. \end{aligned} \tag{27}$$

Here H is the Heaviside function

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0 \end{cases}$$

and μ_+, μ_- are the mean image intensities corresponding to the positive, respectively negative regions of ϕ. This term aims to separate the two regions (background/hand) by maximizing the distance between their observed mean intensities.

To describe each class i, we use a local Gaussian model of the LSF (Rousson and Paragios 2002):

$$p_i^{(x,y)}(\phi) = \frac{1}{\sqrt{2\pi}\sigma_i((x, y))} e^{-\frac{(\phi(x,y)-\phi_i(x,y))^2}{2\sigma_i^2(x,y)}}. \tag{28}$$

Here $(x, y) \in \Omega$ is an image location, ϕ_i is the average LSF of class i and the variance $\sigma_i(x, y)$ models the local variability of the level set at (x, y). Assuming densities independent across pixels, the likelihood function $P_i(\phi)$ is given by the density product over the image domain:

$$P_i(\phi) = \prod_{(x,y)\in\Omega} p_i^{(x,y)}(\phi). \tag{29}$$

Substituting likelihoods $P_i(\phi)$ and augmenting by similarity transformations $h_{\boldsymbol{\tau}^i}$ (including translation, rotation, and scale) that align each prior i with contour ϕ, the prior energy (20) becomes:

$$\begin{aligned} &E_{\text{prior}}(\phi, \mathcal{L}, \boldsymbol{\tau}^{i=1..M}) \\ &= \sum_{i=1}^{M} \Bigg(-\log w_t(i) \\ &\quad + \iint_{\Omega} \Bigg(\frac{(\phi(x, y) - \phi_i(h_{\boldsymbol{\tau}^i}(x, y))/s^i)^2}{2\sigma_i^2(h_{\boldsymbol{\tau}^i}(x, y))} \\ &\quad + \log \sigma_i(h_{\boldsymbol{\tau}^i}(x, y)) \Bigg) dxdy \Bigg) L_i^2 + \beta \left(1 - \sum_{i=1}^{M} L_i^2\right)^2. \end{aligned} \tag{30}$$

Here $\boldsymbol{\tau} = \{s, \theta, T_x, T_y\}$ are the parameters of a similarity transformation

$$h_{\boldsymbol{\tau}}\left([x \quad y]^T\right) = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}, \tag{31}$$

and the index i in $h_{\boldsymbol{\tau}^i}$ designates the prior which is being aligned.

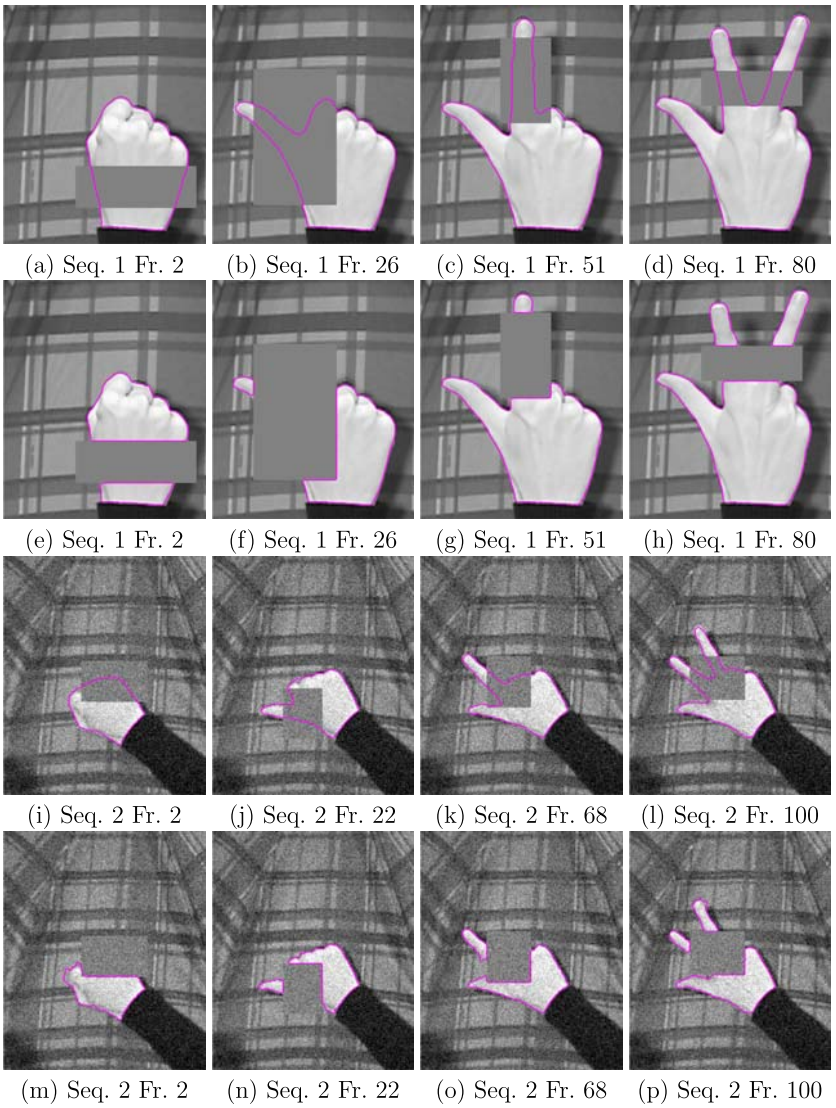
The total energy (19), combining (27) and (30), is minimized via the calculus of variations and gradient descent, resulting in evolution equations for the contour ϕ, the labels $\mathcal{L}$ and the alignment parameters $\boldsymbol{\tau}^{i=1..M}$ (see Appendix).

3.2 Training the Model

In the training phase, we start by segmenting and class-labeling sequences of gestures (0, 1, 2, 3 and 3, 2, 1, 0) performed on a simple contrasting background, as in Fig. 2. Then, we align the resulted contours for each class with respect to similarity transformations (scale, rotation and translation) using genetic algorithms (Davis 1991). Next, we use aligned contours for each class to estimate the parameters of the HMM. Namely, for the Gaussian likelihoods, we use the method described in Rousson and Paragios (2002) to obtain smooth estimates of the mean ϕ_i and variance σ_i for each class i. We learn the state initial and transition probabilities by counting the occurrences of starting classes and of transitions between classes from the training sequences. Alternatively, one can estimate the HMM parameters using an expectation-maximization (EM) approach, via the Baum-Welch algorithm (see Rabiner 1989).

3.3 Results

We tested this implementation of our framework on new image sequences of a hand performing the succession of gestures 0, 1, 2, 3, 2, 1, 0, in front of a complex background and degraded by occlusions. The segmentation contour for the first image of each sequence has been determined by a manual initialization in the proximity of the hand, followed by segmentation using only the data term (27). The parameters for the variational segmentation were $\alpha = 5000$ and



(a) Seq. 1 Fr. 2 (b) Seq. 1 Fr. 26 (c) Seq. 1 Fr. 51 (d) Seq. 1 Fr. 80

(e) Seq. 1 Fr. 2 (f) Seq. 1 Fr. 26 (g) Seq. 1 Fr. 51 (h) Seq. 1 Fr. 80

(i) Seq. 2 Fr. 2 (j) Seq. 2 Fr. 22 (k) Seq. 2 Fr. 68 (l) Seq. 2 Fr. 100

(m) Seq. 2 Fr. 2 (n) Seq. 2 Fr. 22 (o) Seq. 2 Fr. 68 (p) Seq. 2 Fr. 100

Fig. 3 (**a**)–(**d**), (**i**)–(**l**) Segmentation with the proposed framework (Gaussian likelihood implementation) of two image sequence in the presence of occlusion, background complexity and noise (second sequence). (**e**)–(**h**), (**m**)–(**p**) Conventional segmentation of the same image sequences

$\nu = 4000$. The average execution time using un-optimized code (Matlab and C) was 3–4 minutes per frame.

Our framework brings considerable improvements to the segmentation/behavior inference task, even in the case of employing the unsophisticated Gaussian likelihood model. By virtue of the prior information supplied by the inference process, segmentation is able to cope with severe occlusions, as can be seen in Fig. 3(a)–(d), (i)–(l). Figure 3(e)–(h), (m)–(p) shows that the results obtained on the same sequence with conventional segmentation are clearly inferior, since the desired shape of the object cannot be recovered because of the occlusions.

Figure 4 shows the inference results for the first test sequence, which correctly follow the test gesture sequence and our understanding of it in terms of the executed gestures. Moreover, the frame classification obtained by backtracking from the inference process corresponds to the partial classification results obtained throughout the sequence, which have been used to guide segmentation. This concordance can be seen in Fig. 4, which exhibits, as functions of time (frame), (a) the final classification, (b) the delta functions of each class, and (c) the prior confidence of each class (the w function) used as input to the segmentation. The w values have been scaled with respect to their maximum value for every frame.

4 Application to Finger-Spelling Recognition

In sign languages, information is mostly conveyed through a word-level sign vocabulary. Finger-spelling is the part of sign language which connects it with the surrounding (spoken) languages. It consists of manual representations of alphabet letters (Padden and Gunsauls 2003), used to spell words without sign equivalent (e.g. proper nouns or foreign words).

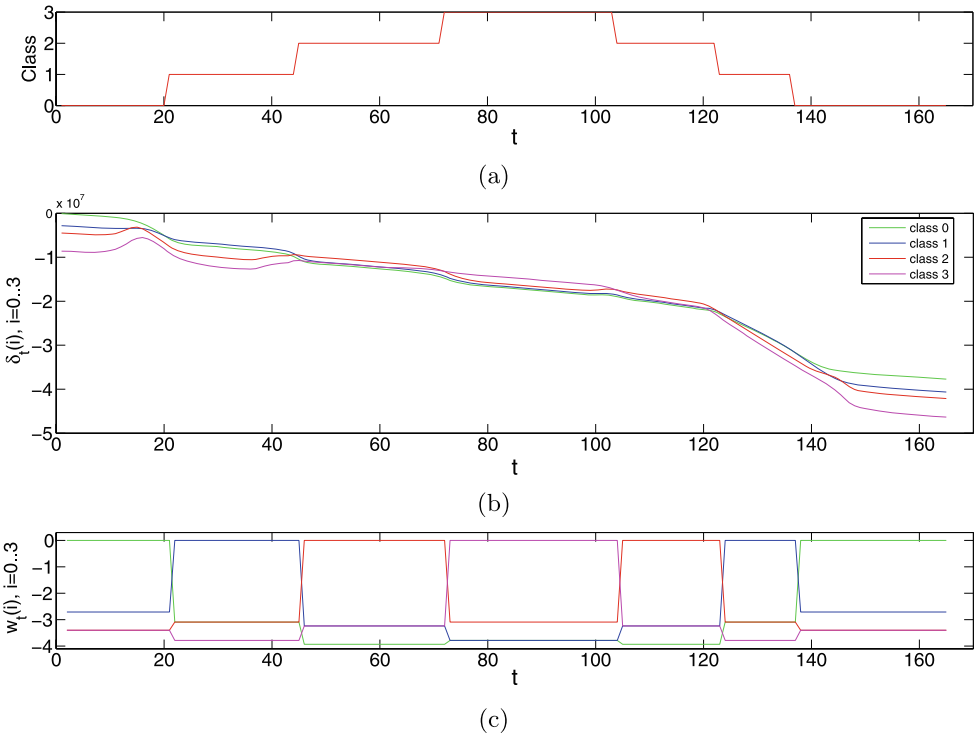


Fig. 4 Behavior inference results plotted per frame. (**a**) Final frame classification. (**b**) Delta functions of each class. (**c**) Prior confidence of each class used as input to the segmentation

Fig. 5 Manual alphabet of the French-speaking part of Switzerland. Reproduced from FSS (2007)

Table 1 Vocabulary of our finger-spelling application

ALBANIA	ALGERIA	ARMENIA	AUSTRIA	BELARUS	BELGIUM
BURUNDI	CROATIA	DENMARK	ECUADOR	ERITREA	ESTONIA
FINLAND	GEORGIA	GERMANY	HUNGARY	ICELAND	LEBANON
LESOTHO	LIBERIA	MOLDOVA	NAMIBIA	NIGERIA	ROMANIA
SENEGAL	SOMALIA	TUNISIA	UKRAINE	URUGUAY	VIETNAM

The second application that we used to test our framework focuses on finger-spelling recognition. It is more challenging than our first application, since it involves a larger number of classes and poorer discrimination among them. We use the manual alphabet of the French-speaking part of Switzerland (FSS 2007), depicted in Fig. 5. Our goal is to perform finger-spelling recognition on a 30 word-vocabulary containing country names, as presented in Table 1.

With the support of the Swiss Federation for the Hearing-Impaired (FSS 2007), we have acquired a data base containing image sequences of a hearing-impaired person finger-spelling the above mentioned words. Acquisition has been performed both in ideal conditions (contrasting background, low speed gesturing), for training purposes, and realistic ones (cluttered background, normal speed gesturing), for testing purposes.

4.1 Implementation Using the Proposed Framework

For this application, we maintain the same object attribute as for our finger-counting application. Namely, we use the hand contour $A(C, I) = C$, represented via the LSF $\phi : \Omega \to \mathbb{R}$, which is the signed distance function to the contour.

As can be seen in Fig. 5, the gestures corresponding to different letters are not easy to differentiate, with letters pairs such as (A, S), (G, H), (M, N) or (R, U) easily confoundable. In these conditions, we make use of the multiple

HMM variant of our framework (Sect. 2.1.2), allowing us to introduce prior knowledge regarding the allowed words (belonging to the vocabulary in Table 1).

The words in our vocabulary constitute our behavior types and each of them is modeled by an individual HMM. Letters are the common basic components of all words and are modeled as shared states (behavior classes) of our HMMs.

Regarding the state probability models, a limitation of the Gaussian likelihood model that we used in our first application is the fact that the mean and variance of the prior corresponding to each class are fixed throughout the image sequence, and thus cannot adapt to varying shapes of the same class. This makes it difficult to obtain accurate segmentations for images where the winning class prior doesn't offer a close match to the image, even after the similarity transformation. For this application, we obtain an improvement with respect to this limitation by using PCA-based likelihood models, which adapt dynamically to the content of new images, as we describe in the following.

The likelihood model $P_i(\phi)$ for each class i relies on a shape distance function between the segmenting contour and a prior contour corresponding to that class (Bresson et al. 2006). The prior contours are computed via principal components analysis (PCA) from specific training data for each class. They evolve during segmentation so as best to match image information, within class constraints imposed by the PCA. We improve the distance function proposed in Bresson et al. (2006) by making it symmetric, resulting into likelihood models which are suitable for classification. Symmetry in the construction of shape priors for level set functions is advocated in Cremers and Soatto (2003).

The purpose of PCA is to reduce redundant information and summarize the main variations of a training set. Based on the training LSFs for a class, we approximate a new LSF $\hat{\boldsymbol{\phi}}$ from that class via PCA as:

$$\hat{\boldsymbol{\phi}} = \overline{\boldsymbol{\phi}} + \mathbf{E}\mathbf{c}. \tag{32}$$

Here $\overline{\boldsymbol{\phi}}$ is the mean of the training LSFs, $\mathbf{E}$ is a matrix whose columns are the reduced set of p PCA eigenvectors, obtained from the covariance matrix of the training data and corresponding to the p largest eigenvalues, and $\mathbf{c}$ is the p-dimensional vector of eigencoefficients.

Our shape distance function between the current segmenting contour ϕ and the prior contour $\hat{\phi}$ is given by:

$$d(\phi, \mathbf{c}, \boldsymbol{\tau}) = \iint_\Omega \left(\hat{\phi}^2 |\nabla\phi|\delta(\phi) + \phi^2 |\nabla\hat{\phi}|\delta(\hat{\phi})\right) dxdy. \tag{33}$$

Here, δ is the Dirac function and $\hat{\phi}$ is the continuously interpolated LSF of the prior contour, obtained from (32):

$$\hat{\phi}(\mathbf{c}, \boldsymbol{\tau})\,|_{(x,y)} = \frac{1}{s}\left(\overline{\boldsymbol{\phi}}(h_{\boldsymbol{\tau}}(x, y)) + \mathbf{E}(h_{\boldsymbol{\tau}}(x, y))\mathbf{c}\right). \tag{34}$$

Here $\boldsymbol{\tau} = \{s, \theta, T_x, T_y\}$ are the parameters of a similarity transformation $h_{\boldsymbol{\tau}}$ (31) which aligns the prior with contour ϕ. Since $\iint_\Omega |\nabla\phi|\delta(\phi)dxdy$ represents the length of the zero level set of ϕ and the LSFs are represented as signed distance functions, we readily observe that the first term of (33) approximates the minimal Euclidean distance to the prior contour, integrated along the segmenting contour. The second term of (33) exchanges the roles of ϕ and $\hat{\phi}$ relative to the first term, making the distance function symmetric and thus suitable for use in classification. Based on this distance function, we define the likelihood of the segmenting contour represented by ϕ, for time t (image $I(t)$) and class i as

$$P_i(\phi(t)) = e^{-d(\phi(t), \mathbf{c}^i(t), \boldsymbol{\tau}^i(t))}. \tag{35}$$

We use the piecewise constant Mumford-Shah model (Chan and Vese 2001) to guide the evolution of the main contour ϕ and prior contours $\hat{\phi}_i(\mathbf{c}^i, \boldsymbol{\tau}^i)$, in terms of their parameters $\mathbf{c}^i$ and $\boldsymbol{\tau}^i$:

$$\begin{aligned}
&E_{\text{data}}(\phi, \mathbf{c}^{i=1..M}, \boldsymbol{\tau}^{i=1..M}) \\
&= E_{\text{MS}}(\phi) + \sum_{i=1}^{M} E_{\text{MS}}(\hat{\phi}_i) \\
&= \iint_\Omega (I - \mu_{\phi+})^2 H(\phi) + (I - \mu_{\phi-})^2 H(-\phi) dxdy \\
&\quad + \sum_{i=1}^{M} \iint_\Omega (I - \mu_{\hat{\phi}_i+})^2 H(\hat{\phi}_i) \\
&\quad + (I - \mu_{\hat{\phi}_i-})^2 (H(-\hat{\phi}_i)) dxdy \\
&\quad + \nu \iint_\Omega |\nabla H(\phi)| dxdy.
\end{aligned} \tag{36}$$

Here H is the Heaviside function, $\mu_{\phi+}$, $\mu_{\hat{\phi}_i+}$ and $\mu_{\phi-}$, $\mu_{\hat{\phi}_i-}$ are the mean image intensities over the positive, respectively negative, regions of the LSFs ϕ and $\hat{\phi}_i$. Function $\hat{\phi}_i = \hat{\phi}_i(\mathbf{c}^i, \boldsymbol{\tau}^i)$ is the continuously interpolated LSF of the prior contour (34), and the last term of (36) imposes smoothness of contour ϕ.

The prior term of the energy, based on models $\delta_t(\phi, i)$ in the multiple HMM framework (18), is obtained from (20) by substituting likelihoods $P_i(\phi)$ (35):

$$\begin{aligned}
&E_{\text{prior}}(\phi, \mathcal{L}, \mathbf{c}^{i=1..M}, \boldsymbol{\tau}^{i=1..M}) \\
&= \sum_{i=1}^{M} \left(-\log w_t(i) + d(\phi(t), \mathbf{c}^i(t), \boldsymbol{\tau}^i(t))\right) L_i^2 \\
&\quad + \beta \left(1 - \sum_{i=1}^{M} L_i^2\right)^2.
\end{aligned} \tag{37}$$

Fig. 6 Sample images (and corresponding letter/behavior classes) from training sequences used in our application

Towards computational efficiency, we adopt a pruning strategy, using only the top 4 most probable priors (out of the 20 available priors) to guide the segmentation of each image. These top 4 prior letters are chosen using the maximum prior letter probabilities, computed with (17). This pruning strategy does not affect recognition performance, while diminishing segmentation time and improving convergence towards the optimal prior.

The total energy (19), summing (36) and (37), is minimized via the calculus of variations and gradient descent. The evolution equations for the LSF ϕ, the labels $\mathcal{L}$, the PCA and alignment parameters $\mathbf{c}^i$ and $\boldsymbol{\tau}^i$ are presented in Appendix.

4.2 Training the Model

We have trained our model using image sequences of each vocabulary word from the acquired database. For training, the gesturing person was filmed on a dark, contrasting background and the gestures were performed at slow speed. Figure 6 presents images from the training sequences.

First, the gesturing hand has been segmented in each training sequence and the resulted contours have been assigned to their respective letter classes and aligned with respect to similarity transformations (scale, rotation and translation) using genetic algorithms (Davis 1991). Subsequently, a separate HMM was trained for each vocabulary word (Rabiner 1989), as follows. The observation probabilities for the shared HMM states have been learned by PCA ($p = 20$) separately from the training contours of each letter class. This resulted in a corresponding mean $\overline{\boldsymbol{\phi}}_i$ and eigenvectors $\mathbf{E}_i$ for each letter/behavior class i. The state initial and transition probabilities have been learned by counting the occurrences of starting classes and of transitions between classes from the training sequences. As mentioned in Sect. 3.2, one could alternatively estimate the HMM parameters through an expectation-maximization (EM) approach, via the Baum-Welch algorithm (see Rabiner 1989).

4.3 Results

We tested the resulting implementation of our framework on image sequences of the same person finger-spelling words from the vocabulary. For testing, we have considered realistic conditions, involving a cluttered background, normal gesturing speed and changed lighting conditions with respect to the training image sequences. Despite the complexity of the task, the results are accurate in terms of the recognized words, due to the infusion of knowledge about the dynamics of vocabulary words via our collaborative framework.

Figure 7 presents examples of cooperative segmentation and behavior inference on three image sequences, which have been correctly classified as the words "ALBANIA", "BELARUS" and "BURUNDI" respectively. The classification framework has helped orient segmentation towards the correct behavior classes at each time instance. Moreover, the dynamical PCA-based class prior models have adapted to significant shape variations within behavior classes, allowing the segmentation of the hand in difficult cases of cluttered background. The frame-wise behavior inference results for these sequences, yielded by the winner HMMs, are presented in rows 2, 4 and 6 of Fig. 7 and correspond to our understanding of the sequences in terms of the executed gestures. In contrast, using the traditional (sequential) approach for recognition, i.e. first segmenting the image sequences (with the same variational approach, without prior models) and then performing inference (with the same HMMs), produces completely erroneous results. Figure 8 shows such a result for the "ALBANIA" sequence, where the segmentation has been side-tracked by the cluttered background, and as a result the sequence has been miss-classified as "ICELAND".

The variational segmentation parameters for the presented test sequences were $\alpha = 4000$ and $\nu = 4000$. The average execution time using un-optimized code (Matlab and C) was 6–7 minutes per frame. The segmenting contour of the first image of each sequence has been automatically determined by the succession of the following steps:

1. initialization with regularly distributed small circles,
2. variational segmentation with the piecewise-constant Mumford-Shah model (27),
3. elimination of small regions by morphological operations,
4. alignment of the mean LSFs $\overline{\phi}_i$ for each letter prior i with respect to the current contour, with genetic algorithms (Davis 1991),

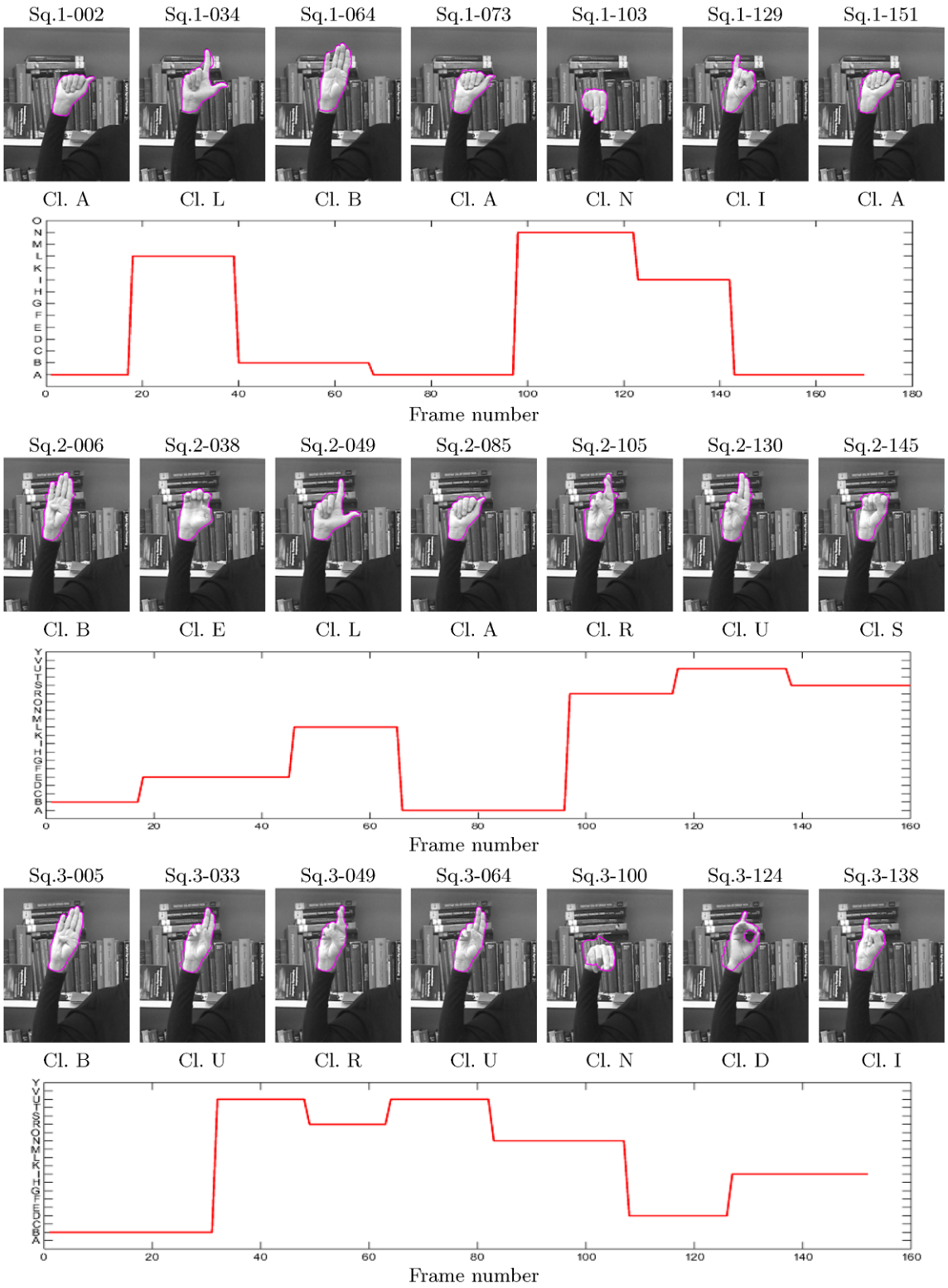


Fig. 7 Correct segmentation and behavior inference using our framework, demonstrated on three test sequences representing the words "Albania" (*rows* 1–2), "Belarus" (*rows* 3–4) and "Burundi" (*rows* 5–6)

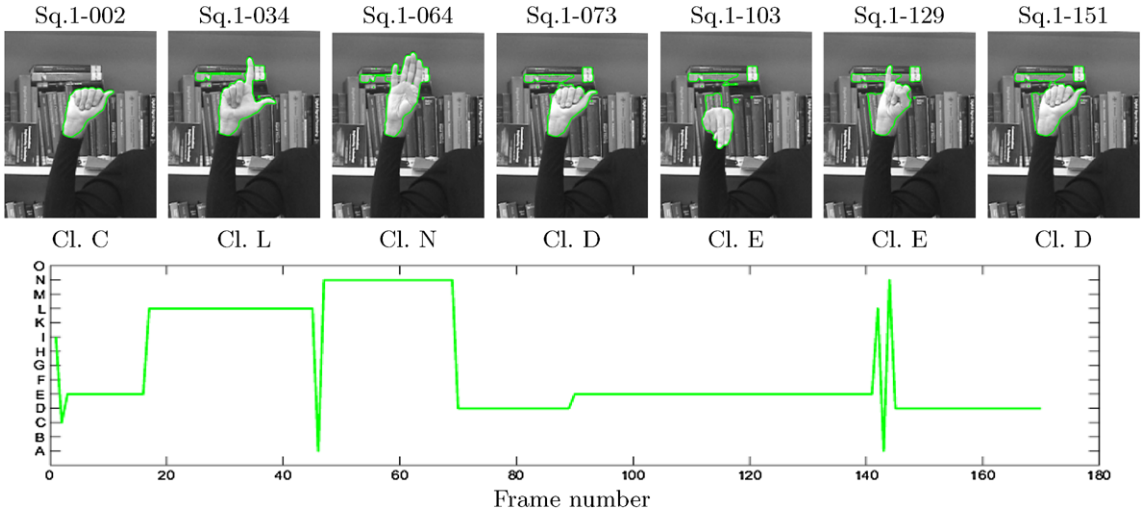


Fig. 8 Erroneous results using *sequential* segmentation and behavior inference on the "Albania" sequence, miss-classified as "Iceland"

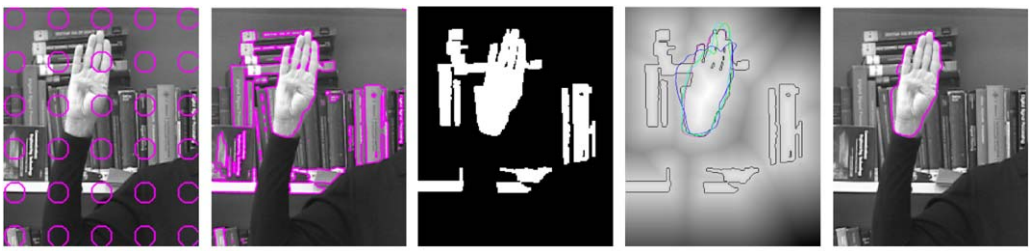


Fig. 9 Initialization process for the first frame of the "Belarus" sequence. (**a**) Initialization with small circles (step 1), (**b**) variational segmentation with the piecewise-constant Mumford-Shah model (step 2), (**c**) elimination of small regions by morphological operations (step 3): resulting binary mask, (**d**) alignment of the mean LSFs for the 4 top fitting priors (steps 4 and 5): image of the current LSF and its zero level set in black, together with the aligned means of the best fitting 4 priors (B, R, U, A) in color, (**e**) variational segmentation result using the piecewise-constant Mumford-Shah model and the top 4 best fitting priors in a competition approach (step 6)

5. choice of the best fitting priors in terms of the distance $d(\phi, \overline{\phi}_i) = \iint_\Omega \phi^2 |\nabla \overline{\phi}_i| \delta(\overline{\phi}_i) dxdy / \iint_\Omega |\nabla \overline{\phi}_i| \delta(\overline{\phi}_i) dxdy$,
6. variational segmentation using data term (36) and the top 4 best fitting priors obtained at step 5, in a competition approach (prior term (20) with $\delta_t(\phi, i) = P_i(\phi)$, given by (35)). This process is illustrated in Fig. 9 for the first image of the "BELARUS" sequence.

One of the advantages of performing behavior inference (via the Viterbi algorithm) in parallel with image segmentation is the fact that it offers us, at each instance t, the optimal classification of the sequence up to time t, which is used to guide further segmentation. This allows the correction of potential cases of miss-classification of previous frames, thus adding robustness to our approach. An example of miss-classification which is corrected in later frames is presented in Fig. 10, which shows partial classification results for the "Belarus" sequence. The partial classification result at frame 19 yields erroneous results (letter U instead of either B or E) for frames 17–19, which are transition frames between two letters (see Fig. 10, first row). This result is corrected at frame 20, where letter E is clearly perceived and the Viterbi algorithm corrects the classification of the previous frames (Fig. 10, second row).

As can be seen from our experimental results, this implementation of our general framework can cope with difficult cluttered background. In this respect, it was shown to perform better than sequential segmentation and classification. However, its performance is still bounded due to the simplicity of the segmentation model. For instance, a challenging case for our method would be one where the average intensity level of the background is similar to that of the hand. In this case, our method would be incapable of discriminating the hand from the background, despite prior knowledge regarding the most likely behavior classes, offered by the inference process. The solution lies in choosing more complicated segmentation models, which would augment the computational costs of the method. Other challenges for our method would be poor resolution images (since it would increase class ambiguity in terms of hand contour), very noisy

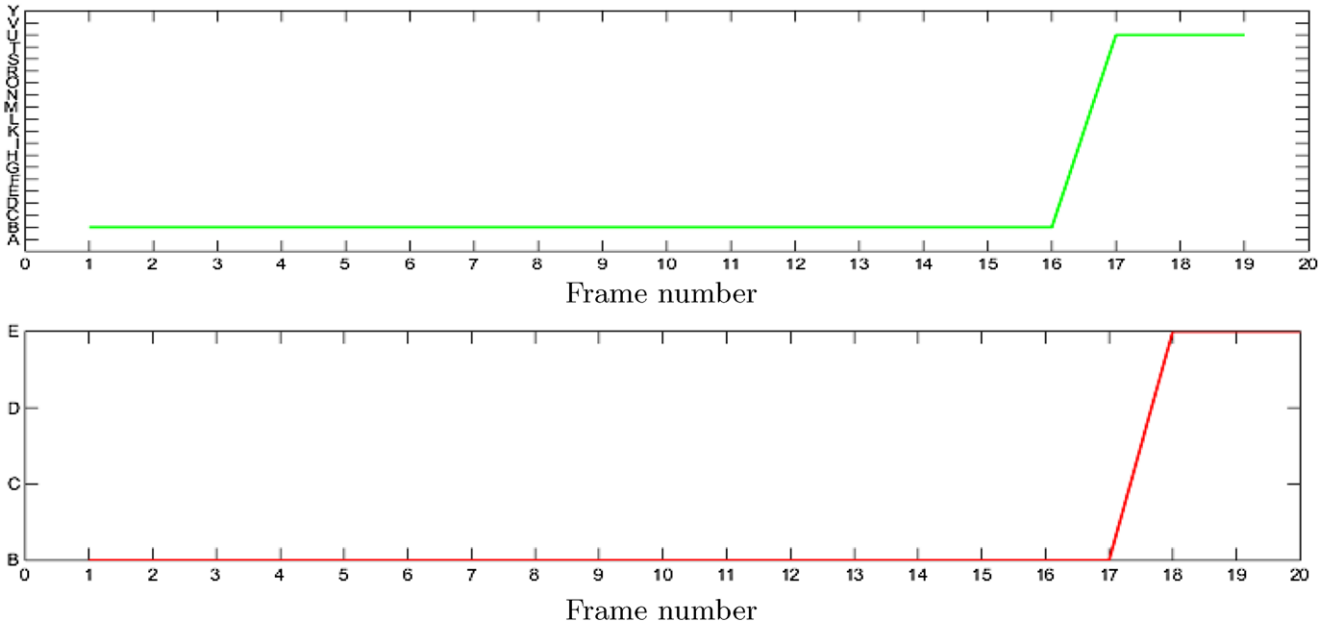


Fig. 10 Partial classification results for the "Belarus" sequence: at frame 19 (*first row*) and at frame 20 (*second row*). Mislabeling of 3 frames starting at frame 17 (*first row*), corrected in subsequent frames starting with 20 (*second row*)

images (leading segmentation into unwanted local minima that match the wrong class prior information) or an important number of missing frames from the video sequences (misleading for the inference process).

5 Conclusion

We have introduced and developed a novel and general framework that enables cooperative segmentation and object behavior inference in image sequences. The proposed collaboration between the segmentation and inference processes facilitates a mutual exchange of information, which is beneficial to their joint success. In particular, we employed an inference strategy based on generative models that provides dynamic probabilistic attribute priors to guide image segmentation. These priors enable the segmentation process to work towards the same goal as the inference process, by outlining the object that best accounts both for the image data and for the prior knowledge encapsulated in the generative model.

Appendix A: Image Segmentation Using the Gaussian Prior Model

A.1 Evolution Equations

Let us denote $\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)) = \phi_i(h_{\boldsymbol{\tau}^i}(x,y))/s^i$. The evolution equation for the segmenting contour ϕ is:

$$\frac{\partial \phi(x,y)}{\partial t} = \delta_\varepsilon(\phi(x,y))\Bigg((I(x,y)-\mu_-)^2 - (I(x,y)-\mu_+)^2 + \nu \mathrm{div}\left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right)\Bigg) + \alpha\sum_{i=1}^{M}\frac{\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)) - \phi(x,y)}{\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))}L_i^2, \tag{38}$$

where δ_ε is a regularized version of the Dirac function: $\delta_\varepsilon(x) = \frac{\varepsilon}{\pi(x^2+\varepsilon^2)}$. The similarity transformation parameters of each prior evolve according to:

$$\begin{aligned}\frac{\partial \tau^i}{\partial t} = &-\iint_\Omega \frac{1}{\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))} \times \left(\nabla\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))\cdot\frac{\partial}{\partial\tau^i}(h_{\boldsymbol{\tau}^i}(x,y))\right)dxdy \\ &+\iint_\Omega \frac{\phi(x,y)-\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))}{\sigma_i^2(h_{\boldsymbol{\tau}^i}(x,y))} \times \frac{\partial}{\partial\tau^i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))dxdy \\ &+\iint_\Omega \frac{(\phi(x,y)-\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))^2}{\sigma_i^3(h_{\boldsymbol{\tau}^i}(x,y))} \times \left(\nabla\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))\cdot\frac{\partial}{\partial\tau^i}(h_{\boldsymbol{\tau}^i}(x,y))\right)dxdy.\end{aligned} \tag{39}$$

where $\boldsymbol{\tau}^i$ stands for each of s^i, θ^i, and T^i and

$$\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))) = \nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))\cdot\frac{\partial}{\partial\tau_i}(h_{\boldsymbol{\tau}^i}(x,y)), \tag{40}$$

if $\tau^i = \theta^i, T^i$ and

$$\frac{\partial}{\partial \tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))) = \nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)) \cdot \frac{\partial}{\partial \tau_i}(h_{\boldsymbol{\tau}^i}(x,y)) - \frac{1}{s^i}\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)), \tag{41}$$

if $\tau^i = s^i$. The derivatives $\partial(h_{\boldsymbol{\tau}^i}(x,y))/\partial\tau^i$ are computed as follows:

$$\frac{\partial}{\partial s}(h_{\boldsymbol{\tau}}(x,y)) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{bmatrix} x \\ y \end{bmatrix}, \tag{42}$$

$$\frac{\partial}{\partial \theta}(h_{\boldsymbol{\tau}}(x,y)) = s\begin{pmatrix} -\sin\theta & \cos\theta \\ -\cos\theta & -\sin\theta \end{pmatrix}\begin{bmatrix} x \\ y \end{bmatrix}, \tag{43}$$

$$\frac{\partial}{\partial T_x}(h_{\boldsymbol{\tau}}(x,y)) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{\partial}{\partial T_y}(h_{\boldsymbol{\tau}}(x,y)) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{44}$$

The labels $L_i, i = 1..M$ evolve according to:

$$\frac{\partial L_i}{\partial t} = L_i\left(\log w_t(i) - \iint_\Omega \left(\frac{(\phi(x,y) - \hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))^2}{2\sigma_i^2(h_{\boldsymbol{\tau}^i}(x,y))} + \log\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))\right)dxdy + 2\beta\left(1 - \sum_{i=1}^{M} L_i^2\right)\right). \tag{45}$$

The update equation for the Lagrange multiplier β is as follows:

$$\beta = \frac{\sum_{i=1}^{M} L_i^2 \log\delta_t(\phi, i)}{2\sum_{i=1}^{M} L_i^2\left(\sum_{i=1}^{M} L_i^2 - 1\right)}, \tag{46}$$

with $\delta_t(\phi, i) = w_t(i)P_i(\phi)$ and $P_i(\phi)$ given by (29).

A.2 Numerical Approach

To minimize energy (19), with E_{data} given by (27) and E_{prior} given by (30), we use the evolution equations (38), (39) and (45). We solve these equations numerically by iterating the following steps until convergence is reached:

1. Computation of the mean intensities μ_+ and μ_- over image I regions corresponding to the positive, respectively negative regions of the LSF ϕ.
2. Computation of the class prior information $\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))$ and $\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))$ from the average LSF $\phi_i(x,y)$ and the variance $\sigma_i(x,y)$, by applying the similarity transformations $h_{\boldsymbol{\tau}^i}$ (31) via the B-splines interpolation method (Unser 1999).
3. Computation of the curvature $\text{div}(\nabla\phi(x,y)/|\nabla\phi(x,y)|)$ and of the gradients $\nabla\sigma_i(h_{\boldsymbol{\tau}^i}(x,y))$ and $\nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))$ using a central difference scheme.
4. Calculation of the temporal derivatives in (38), (39) and (45) using a forward difference approximation.
5. Re-distancing of the LSF ϕ with the fast marching method of Adalsteinsson and Sethian (1995).
6. Update of the Lagrange multiplier β according to (46).

Appendix B: Image Segmentation Using the PCA-Based Prior Model

B.1 Evolution Equations

Main contour evolution is governed by the equation:

$$\begin{aligned}\frac{\partial\phi(x,y)}{\partial t} &= \delta_\varepsilon(\phi(x,y))\Big((I(x,y)-\mu_-)^2 - (I(x,y)-\mu_+)^2\Big) \\ &\quad + \nu\text{div}\left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right) \\ &\quad + \alpha\sum_{i=1}^{M}\left(\hat{\phi}_i^2(h_{\boldsymbol{\tau}^i}(x,y))\text{div}\left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right)\delta_\varepsilon(\phi(x,y))\right. \\ &\quad + \left(\nabla\hat{\phi}_i^2(h_{\boldsymbol{\tau}^i}(x,y)) \cdot \left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right)\right)\delta_\varepsilon(\phi(x,y)) \\ &\quad \left. - 2\phi(x,y)|\nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))|\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))\right)L_i^2.\end{aligned} \tag{47}$$

The similarity transformation parameters of each prior evolve according to:

$$\begin{aligned}\frac{\partial\tau^i}{\partial t} &= \iint_\Omega (I(x,y) - \mu_{\hat{\phi}_i-})^2\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))) \\ &\quad \times \frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))dxdy \\ &\quad - \iint_\Omega (I(x,y) - \mu_{\hat{\phi}_i+})^2\delta_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))) \\ &\quad \times \frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))dxdy \\ &\quad - 2\iint_\Omega \hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))\frac{\partial}{\partial\tau_i}(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y))|\nabla\phi(x,y)| \\ &\quad \times \delta_\varepsilon(\phi(x,y)))dxdy \\ &\quad - \iint_\Omega \phi^2(x,y)\Big(|\nabla\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)|\delta'_\varepsilon(\hat{\phi}_i(h_{\boldsymbol{\tau}^i}(x,y)))\end{aligned}$$

$$\times \frac{\partial}{\partial \tau_i}(\hat{\phi}_i(h_{\tau^i}(x,y)) + \delta_\varepsilon(\hat{\phi}_i(h_{\tau^i}(x,y)))\frac{1}{|\nabla\hat{\phi}_i(h_{\tau^i}(x,y)|} \times \left((\hat{\phi}_i)_x(h_{\tau^i}(x,y))\frac{\partial}{\partial\tau_i}((\hat{\phi}_i)_x(h_{\tau^i}(x,y))) + (\hat{\phi}_i)_y(h_{\tau^i}(x,y)\frac{\partial}{\partial\tau_i}((\hat{\phi}_i)_y(h_{\tau^i}(x,y))\right)dxdy, \tag{48}$$

where τ^i stands for each of s^i, θ^i, and T^i, and $(\hat{\phi}_i)_x$, $(\hat{\phi}_i)_y$ are the x and y derivatives of $\hat{\phi}_i$. The derivatives $\partial\hat{\phi}_i(h_{\tau^i}(x,y))/\partial\tau_i$, $\partial(\hat{\phi}_i)_x(h_{\tau^i}(x,y))/\partial\tau_i$ and $\partial(\hat{\phi}_i)_y(h_{\tau^i}(x,y))/\partial\tau_i$ are computed as in (40), (41). The evolution equation for the jth PCA coefficient of prior class i is:

$$\begin{aligned}\frac{\partial \mathbf{c}_j^i}{\partial t} = {} & \frac{1}{s^i}\iint_\Omega (I(x,y)-\mu_{\hat{\phi}_i-})^2 \times \delta_\varepsilon(\hat{\phi}_i(h_{\tau^i}(x,y)))E_{ij}(h_{\tau^i}(x,y))dxdy \\ & - \frac{1}{s^i}\iint_\Omega (I(x,y)-\mu_{\hat{\phi}_i+})^2 \times \delta_\varepsilon(\hat{\phi}_i(h_{\tau^i}(x,y)))E_{ij}(h_{\tau^i}(x,y))dxdy \\ & - \frac{2}{s^i}\iint_\Omega \hat{\phi}_i(h_{\tau^i}(x,y))E_{ij}(h_{\tau^i}(x,y)) \times |\nabla\phi(x,y)|\delta_\varepsilon(\phi(x,y))dxdy \\ & - \frac{1}{s^i}\iint_\Omega \phi^2(x,y) \times \left(|\nabla\hat{\phi}_i(h_{\tau^i}(x,y))|\delta'_\varepsilon(\hat{\phi}_i(h_{\tau^i}(x,y)))E_{ij}(h_{\tau^i}(x,y)) + \delta_\varepsilon(\hat{\phi}_i(h_{\tau^i}(x,y)))\frac{1}{|\nabla\hat{\phi}_i(h_{\tau^i}(x,y))|} \times \left((\hat{\phi}_i)_x(h_{\tau^i}(x,y))(E_{ij})_x(h_{\tau^i}(x,y)) + (\hat{\phi}_i)_y(h_{\tau^i}(x,y))(E_{ij})_y(h_{\tau^i}(x,y))\right)\right)dxdy, \end{aligned} \tag{49}$$

where E_{ij} is the jth eigenvector of class i, arranged as the columns of an image-sized matrix (continuously interpolated) and $(E_{ij})_x$ and $(E_{ij})_y$ are its x and y derivatives, respectively. The labels $L_i, i = 1..M$, evolve according to:

$$\begin{aligned}\frac{\partial L_i}{\partial t} = {} & L_i\left(\log w_t(i) - \iint_\Omega \hat{\phi}_i{}^2(h_{\tau^i}(x,y)) \times |\nabla\phi(x,y)|\delta_\varepsilon(\phi(x,y))dxdy \right. \\ & \left. - \iint_\Omega \phi^2((x,y))|\nabla\hat{\phi}_i(h_{\tau^i}(x,y))| \times \delta_\varepsilon(\hat{\phi}_i(h_{\tau^i}(x,y)))dxdy + 2\beta\left(1-\sum_{i=1}^M L_i^2\right)\right). \end{aligned} \tag{50}$$

The update equation for the Lagrange multiplier β is as follows:

$$\beta = \frac{\sum_{i=1}^M L_i^2 \log\delta_t(\phi,i)}{2\sum_{i=1}^M L_i^2\left(\sum_{i=1}^M L_i^2 - 1\right)}, \tag{51}$$

with $\delta_t(\phi,i) = w_t(i)P_i(\phi)$ and $P_i(\phi)$ given by (35).

B.2 Numerical Approach

To minimize energy (19), with E_{data} given by (36) and E_{prior} given by (37), we use the evolution equations (47), (48), (49) and (50). We solve these equations numerically by iterating the following steps until convergence is reached:

1. Computation of the mean intensities μ_+ and μ_- over image I regions corresponding to the positive, respectively negative regions of the LSF ϕ.
2. Computation of the mean intensities $\mu_{\hat{\phi}_i+}$ and $\mu_{\hat{\phi}_i-}$ over image I regions corresponding to the positive, respectively negative regions of the LSFs $\hat{\phi}_i$.
3. Computation of the class prior information $\hat{\phi}_i(h_{\tau^i}(x,y))$ and $E_{ij}(h_{\tau^i}(x,y))$ from the average LSF $\phi_i(x,y)$ and the eigenvectors $E_{ij}(x,y)$, by using (34) and applying the similarity transformations h_{τ^i} (31) via the B-splines interpolation method (Unser 1999).
4. Computation of the curvature $\text{div}(\nabla\phi(x,y)/|\nabla\phi(x,y)|)$, derivatives $(\hat{\phi}_i)_x$, $(\hat{\phi}_i)_y$, $(E_{ij})_x$ and $(E_{ij})_y$ and gradients $\nabla\phi(x,y)$ and $\nabla\hat{\phi}_i(h_{\tau^i}(x,y))$ using a central difference scheme.
5. Calculation of the temporal derivatives in (47), (48), (49) and (50) using a forward difference approximation.
6. Re-distancing of the LSF ϕ with the fast marching method of Adalsteinsson and Sethian (1995).
7. Update of the Lagrange multiplier β according to (51).

References

Adalsteinsson, D., & Sethian, J. (1995). A fast level set method for propagating interfaces. *Journal of Computational Physics*, *118*, 269–277.

Bresson, X., Vandergheynst, P., & Thiran, J.-P. (2006). A variational model for object segmentation using boundary information and shape prior driven by the Mumford-Shah functional. *International Journal of Computer Vision*, *28*(2), 145–162.

Caselles, V., Kimmel, R., & Sapiro, G. (1995). Geodesic active contours. In *Proc. IEEE intl. conf. on comp. vis.* (pp. 694–699). Boston, USA.

Chan, T., & Vese, L. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, *10*(2), 266–277.

Chen, Y., Tagare, H., Thiruvenkadam, S., Huang, F., Wilson, D., Gopinath, K., Briggs, R., & Geiser, E. (2002). Using prior shapes in geometric active contours in a variational framework. *International Journal of Computer Vision*, *50*(3), 315–328.

Cootes, T., Beeston, C., Edwards, G., & Taylor, C. (1999). Unified framework for atlas matching using active appearance models. In *Int'l conf. inf. proc. in med. imaging* (pp. 322–333).

Cremers, D. (2006). Dynamical statistical shape priors for level set based tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(8), 1262–1273.

Cremers, D., & Soatto, S. (2003). A pseudo-distance for shape priors in level set segmentation. In N. Paragios (Ed.), *IEEE 2nd int. workshop on variational, geometric and level set methods*. (pp. 169–176). Nice.

Cremers, D., Guetter, C., & Xu, C. (2006a). Nonparametric priors on the space of joint intensity distributions for non-rigid multi-modal image registration. In *IEEE conference on computer vision and pattern recognition (CVPR)* (Vol. 2, pp. 1777–1783).

Cremers, D., Osher, S. J., & Soatto, S. (2006b). Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, *69*(3), 335–351.

Cremers, D., Sochen, N., & Schnörr, C. (2006c). A multiphase dynamic labeling model for variational recognition-driven image segmentation. *International Journal of Computer Vision*, *66*(1), 67–81.

Davis, L. (1991). *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold.

Ferrari, V., Tuytelaars, T., & Gool, L. V. (2004). Simultaneous object recognition and segmentation by image exploration. In *ECCV*.

FSS (2007). Fédération Suisse des Sourds. http://www.sgb-fss.ch/.

Gui, L., Thiran, J.-P., & Paragios, N. (2007a). Joint object segmentation and behavior classification in image sequences. In *Proc. IEEE conference on computer vision and pattern recognition (CVPR 2007)*. Minneapolis, MN, USA.

Gui, L., Thiran, J.-P., & Paragios, N. (2007b). A variational framework for the simultaneous segmentation and object behavior classification of image sequences. In *Proc. scale space and variational methods in computer vision. Lecture notes in computer science* (pp. 652–664). Berlin: Springer.

Kass, M., Witkin, A., & Terzopoulos, D. (1987). Snakes: active contour models. *International Journal of Computer Vision*, *1*, 321–331.

Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., & Yezzi, A. (1995). Gradient flows and geometric active contour models. In *Proc. IEEE intl. conf. on comp. vis.* (pp. 810–815).

Kokkinos, I., & Maragos, P. (2005). An expectation maximization approach to the synergy between image segmentation and object categorization. In *ICCV* (pp. 617–624).

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on SLCV*.

Leventon, M., Grimson, W., & Faugeras, O. (2000). Statistical shape influence in geodesic active approach. In *IEEE int. conf. on computer vision and pattern recognition* (pp. 316–323).

Malladi, R., Sethian, J., & Vemuri, B. (1995). Shape modeling with front propagation: a level set approach. *IEEE PAMI*, *17*, 158–175.

Mumford, D., & Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications in Pure and Applied Mathematics*, *42*, 577–685.

Osher, S., & Sethian, J. (1988). Fronts propagating with curvature-dependent speed: algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, *79*, 12–49.

Padden, C., & Gunsauls, D. C. (2003). How the alphabet came to be used in a sign language. *Sign Language Studies*, *4*(1), 10–33.

Paragios, N., & Deriche, R. (2002). Geodesic active regions and level set methods for supervised texture segmentation. *International Journal of Computer Vision*, *46*(3), 223–247.

Paragios, N., & Deriche, R. (2005). Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, *97*, 259–282.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2).

Rathi, Y., Vaswani, N., Tannenbaum, A., & Yezzi, A. (2007). Tracking deforming objects using particle filtering for geometric active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(8), 1470–1475.

Rousson, M., & Paragios, N. (2002). Shape priors for level set representations. In *European conference in computer vision* (Vol. 2, pp. 78–92).

Terzopoulos, D., & Szeliski, R. (1992). In *Tracking with Kalman snakes* (pp. 3–20). Cambridge: MIT Press.

Tu, Z., Chen, X., Yuille, A., & Zhu, S. (2003). Image parsing: segmentation, detection, and recognition. In *ICCV* (pp. 18–25).

Unser, M. (1999). Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, *16*(6), 22–38.

Vese, L., & Chan, T. (2002). A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, *50*(3), 271–293.

Zhao, H.-K., Chan, T., Merriman, B., & Osher, S. (1996). A variational level set approach to multiphase motion. *Journal of Computational Physics*, *127*, 179–195.