A Probabilistic Temporal Model for Joint Attribute Extraction and Behavior Recognition

Laura Gui 1, Nikos Paragios 2, Jean-Philippe Thiran 1

1 Signal Processing Laboratory (LTS5),
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
2 Laboratoire MAS - Ecole Centrale de Paris,
Chatenay-Malabry, France

Abstract

The focus of this paper is on the recognition of single object behavior from monocular image sequences. The general literature trend is to perform behavior recognition separately after an initial phase of feature/attribute extraction. We propose a framework where behavior recognition is performed jointly with attribute extraction, allowing the two tasks to mutually improve their results. To this end, we express the joint recognition / extraction problem in terms of a probabilistic temporal model, allowing its resolution via a variation of the Viterbi decoding algorithm, adapted to our model. Within the algorithm derivation, we translate probabilistic attribute extraction into a variational segmentation scheme. We demonstrate the viability of the proposed framework through a particular implementation for finger-spelling recognition. The obtained results illustrate the superiority of our collaborative model with respect to the traditional approach, where attribute extraction and behavior recognition are performed sequentially.

1 Introduction

Visual behavior recognition is currently a highly active research area. This is due both to the scientific challenge posed by the complexity of the task, and to the growing interest in its applications, such as automated visual surveillance, human-computer interaction or video indexing/retrieval. Good reviews, covering a large number of different approaches from the literature, can be found in [16, 1, 22]. The general trend followed by these approaches is the separation of the behavior recognition task into two sequential processes. The first one is a feature extraction process, where features considered relevant for the recognition task are extracted from the input image sequence. The second one is the actual recognition process, where the extracted features are classified in terms of predefined behavior classes. A brief outline of a few existing behavior recognition
approaches will help exemplify this two-task separation, while emphasizing our original contribution.

At a coarse level of image analysis, Sato and Aggarwal [28] track persons as moving boxes and then use the boxes’ motion patterns to recognize two-person pedestrian interactions. More precise modeling is used in Wren et al.’s tracking system [30], which yields a human body representation in terms of 2D blobs associated with the head, torso, hands and feet. Their system is used for the gesture control of two virtual reality applications. Hand-gesture recognition applications usually employ more detailed descriptions of the hand. For instance, Lockton and Fitzgibbon [21] extract the hand mask based on skin color and recognize finger-spelling with a nearest-neighbor classification technique.

Other approaches do not explicitly model or track the object of interest, but rather model motion regions within the image sequence, assumed to be provoked exclusively by the object(s) of interest. For example, Bobick and Davis [5] extract motion energy images (MEI) and motion history images (MHI), indicating the presence / absence of motion and the recency of motion, respectively, at a certain pixel. Subsequently, they use the Hu moments of the MEIs and MHIs to classify the image sequence in terms of the shortest Mahalanobis distance to learned models of each action. A sensitive point of such approaches is the ambiguity among different motions, induced by the integration of information throughout the whole image sequence. Another inherent difficulty is posed by the imprecise nature of the motion detection strategy (e.g., inner regions of moving objects may not be detected).

In our general framework, we represent the object targeted for behavior recognition by its contour within the image. This is the finest level of analysis permitted by a 2D representation, where no assumptions are made about the target object (i.e., via a 2D or 3D object model). This allows us to easily extract any object attributes which are functions of the contour and of the image, such as the color, texture properties or position (in 2D).

Regarding the recognition strategy, some existing approaches perform frame-by-frame classification of features extracted from the input image sequence, via methods like maximum likelihood ([3]) or nearest-neighbor template matching ([21]). These methods are limited to cases where the separate recognition of each frame is feasible, without need of context information.

Other approaches use features extracted from the whole image sequence to globally classify it as one of a set of possible actions. In this direction, [5] and [31] use nearest-neighbors methods for classification. Efros et al. [14] compute a set of figure-centric motion features based on the blurred optical flow, which are subsequently matched with a database of learned motions via spatio-temporal cross-correlation. One difficulty faced by this kind of methods is due to the differences in the speed of performing the compared actions. This creates the necessity for temporal alignment between the compared sequences (e.g. [14]), or for adjustment of temporal parameters used in computing global features over the image sequence (e.g. [5]).

Blackburn and Ribeiro [4] deal with the problem of temporal alignment by
using Dynamic Time Warping (DTW). They project human silhouettes obtained by background subtraction to a lower-dimensional space by isometric non-linear manifold mapping. Then, they classify the trajectory in this space by a nearest-neighbor scheme based on the DTW matching score. A general disadvantage of the DTW method is its ignorance of the interaction between nearby subsequences. A remedy in this respect is offered by the Hidden Markov Model (HMM) [26] — a probabilistic temporal model that represents the correlation between adjacent time instances via a Markov process. Robertson and Reid [27] decompose complex behavior into a set of simple actions, whose succession is modeled with HMMs. The HMM input is given by lower level features such as the trajectory, velocity and local action descriptors.

Of particular appeal for behavior recognition are Dynamic Bayesian Networks (DBNs) [17], which generalize HMMs by permitting more complex dependencies between hidden and observed variables. Park and Aggarwal [25] recognize two-person interactions using a hierarchical Bayesian network (BN). Following the tracking and segmentation of multiple body parts from the image sequence, they estimate body-part poses at the low level of the BN, and the overall body pose — at the high level of the BN. Interactions are classified via a DBN modeling body configuration dynamics.

Our framework is also formulated in terms of a DBN. In our case, the DBN permits the joining of the two processes which are considered in separation by the previous approaches: feature extraction and behavior recognition. Our proposed DBN is based on the coupling between an HMM and a probabilistic image segmentation model, used for attribute extraction from the image sequence and influenced by knowledge from the HMM. In contrast, the existing two-phase approaches automatically discard some information in the phase of attribute extraction, without considering higher level knowledge which could be obtained from the existing training data, and is only used in the recognition phase. Also, the retained attributes could be affected by low image quality (noise, occlusions) or poor separation of the target object(s) from the background. Our proposed approach for behavior recognition relies on the collaboration between the low-level attribute extraction process (performed through image segmentation) and the higher level recognition process. This allows the two processes to mutually improve their results through collaboration and sharing of existing knowledge.

Our current paper builds on the initial formulation for cooperative segmentation and behavior inference, that we introduced in [18, 19, 20]. Our major novel contributions with respect to this previous work are as follows:

- The formulation of a strategy for joint attribute extraction and behavior recognition in terms of a DBN. Such a formulation allows the unitary, principled treatment of the two tasks, as well as the explicit statement of the assumptions that we make, regarding the dependencies among the different variables involved.

- The development of a Viterbi algorithm adapted to our proposed DBN.

- The translation of the probabilistic attribute extraction formulation in-
cluded in our model into a variational segmentation formulation.

The remainder of this paper is organized as follows. In Section 2 we present the proposed probabilistic temporal model and its associated Viterbi algorithm. In Section 3 we translate the probabilistic segmentation model into a variational one. Section 4 deals with the training of our model and Section 5 offers a summary of our approach. In Section 6, we validate our general framework through a finger-spelling recognition application. Section 7 concludes our work.

2 Joint Attribute Extraction and Behavior Recognition: a DBN Formulation

2.1 Goal and Motivation

Limiting the scope of our work to single object segmentation and behavior recognition, we define “behavior” as the temporal evolution of the object, observed in the image sequence. We consider object behavior as being composed of a set of basic primitives, that we call actions. At the basis of behavior recognition lies prior knowledge about the possible action classes, their characteristics and the typical ways in which they associate to compose behaviors. The result of behavior recognition is the recognized behavior, as well as its decomposition into action classes, corresponding to each image.

The aim of our framework is to jointly extract the attributes of the target object from an image sequence and to recognize the exhibited behavior. We model this joint extraction/ recognition problem using a Dynamic Bayesian Network (DBN). A DBN is a probabilistic temporal model that represents a sequence of variables. In particular, our proposed DBN is based on coupling a probabilistic attribute extraction model with a Hidden Markov Model (HMM). An HMM is a type of DBN which associates a sequence of discrete states to a sequence of observations (in our case, a sequence of images). Each state is characterized by a probability distribution, which gives the probability of an observation while being in the respective state. The evolution of the states with time is controlled by a transition distribution, which represents the probability of switching to a certain state given the current state. The states are considered as hidden and the only evidence about them is given by the sequence of observations. An approach based on an HMM is particularly appealing in the context of behavior recognition because a discrete state is a natural representation of a behavior component (action). The transition distribution then models the fact that, inside a particular behavior, certain sequences of actions are more likely to be observed than others.

Regarding the attribute extraction task, the generic term “attribute” designates a vector which encapsulates visual properties of an object. We model attribute extraction in terms of image segmentation, which yields the object contour, thus allowing the easy extraction of any image-based object properties relevant for the recognition task. Formally, the attribute vector can be repre-
sented as a functional \( f_A(I, C) \) of the image \( I \) and of the object’s segmenting contour \( C \) (\( f_A \) is assumed to be differentiable with respect to \( C \)). This definition includes many object properties computable with boundary- and region-based functionals, such as position, orientation, average intensity/color or higher order statistics describing texture. Such flexibility in the choice of the extracted attributes makes our framework adaptable to the needs of a wide range of behavior recognition applications.

2.2 The Model

Given an image sequence \( I_{1:T} = \{I_1, I_2, \ldots, I_T\} \), behavior recognition amounts to determining the behavior type \( b \) exhibited in this sequence, which belongs to a finite set of behavior types \( B = \{B_1, B_2, \ldots, B_K\} \). Since behaviors are decomposed into actions, recognition is based on the determination of the action class \( s_t \) which corresponds to each observed image \( I_t \), yielding the action class sequence \( s_{1:T} = \{s_1, s_2, \ldots, s_T\} \). The action classes that compose the behaviors under study belong to a finite set \( S = \{S_1, S_2, \ldots, S_M\} \). The different behaviors (and their component actions) are distinguished in terms of the object attributes \( A_t \), which are extracted from the images \( I_t \) by means of segmentation. Formally, this can be written as \( A_t = f_A(I_t, C_t) \), where \( f_A \) is the function which associates to a given image \( I_t \) and segmentation contour \( C_t \) the corresponding extracted attribute \( A_t \).

In this context, we model the joint attribute extraction and behavior recognition task using the Dynamic Bayesian Network shown in Fig. 1. The central part of this figure represents the model corresponding to two time slices \(- t - 1 \) and \( t \) — the dots implying that the DBN structure and parameters repeat in a similar fashion, starting from the first time slice, up to the one corresponding to the last image in the modeled sequence. Our model is based on coupling an HMM — whose hidden state at time \( t \) is given by the action class \( s_t \) — with a probabilistic generative attribute extraction model, where the image \( I_t \) depends on the contour \( C_t \) and the attribute \( A_t \). The coupling of the two models at each time \( t \) is realized through the attribute \( A_t \). We represent observed variables by shaded nodes (the images \( I_t, t = 1..T \) and hidden variables by clear nodes (the behavior type \( b \), the action classes \( s_t \), the attributes \( A_t \) and the contours \( C_t, t = 1..T \)). Moreover, we depict discrete variables by square nodes (the behavior type \( b \) and the action classes \( s_t, t = 1..T \)) and continuous variables by circular ones (the attributes \( A_t \), the contours \( C_t \) and the images \( I_t, t = 1..T \)).

According to the DBN represented in Fig. 1, our model is characterized by the following joint variable distribution:

\[
P(I_{1:T}, C_{1:T}, A_{1:T}, s_{1:T}, b) = \prod_{t=1}^{T} P(I_t|A_t, C_t) P(C_t) P(A_t|s_t) P(s_t|s_{t-1}, b) P(b),
\]

where \( P(s_1|s_0, b) \equiv P(s_1|b) \) is the initial action class distribution given the behavior type \( b \). In the following, we explain the assumptions underlying our model and we detail each of the probability factors from the right-hand side
Figure 1: The Dynamic Bayesian Network supporting our joint attribute extraction / behavior recognition framework. This model can be regarded as containing an HMM (in the upper half), coupled with a probabilistic attribute extraction model, based on segmentation (in the lower half). For time slice $t$, the hidden state of the HMM is given by the action class $s_t$. Additionally, the action class $s_t$ depends on the particular behavior type $b$ which is being exhibited. Within the attribute extraction model, the image $I_t$ is dependent on the contour $C_t$ and the attribute $A_t$. The observation at time $t$ is given by the image $I_t$. We depict hidden variables by clear nodes and observed variables by shaded nodes. The square nodes designate discrete variables, whereas circular ones designate continuous variables.

product in (1). To this end, let us look at the decomposition of our joint variable distribution:

$$P(I_{1:T}, C_{1:T}, A_{1:T}, s_{1:T}, b) = P(I_{1:T}, C_{1:T}, A_{1:T}|s_{1:T}, b) P(s_{1:T}, b)$$

$$= \prod_{t=1}^{T} P(I_t, A_t, C_t|s_t) P(s_t|s_{t-1}, b) P(b).$$

The last equality implies two assumptions. One is a first order Markov assumption, namely that the action class at time $t$ only depends on the action class at time $t - 1$ (and on the behavior type $b$), being independent with respect to the action classes previous to time $t - 1$. The second one is that, given the action class at time $t$, the image $I_t$, attribute $A_t$ and contour $C_t$ are independent with respect to all the other variables.

The prior probability of the behavior type $P(b)$ represents a free parameter.
of our framework, whose model can be chosen depending on the application, in order to reflect the fact that some behavior types may be more probable than others. In the absence of such information, a uniform prior $P(b)$ can be chosen.

Within the last decomposition of Eq. 2, let us now look at the joint variable distribution for time slice $t$ given the action class $P(I_t, A_t, C_t | s_t)$. Directly working with such a joint distribution is in general too complicated. The model can often be made more tractable by considering a simpler factorized distribution, where some of the dependencies between the variables are removed. We propose to use a joint distribution of the form

$$P(I_t, C_t, A_t | s_t) = P(I_t | A_t, C_t) P(C_t) P(A_t | s_t),$$

(3)

which, substituted in Eq. 2, leads to the decomposition in Eq. 1, illustrated in Fig. 1.

Let us now explain the significance of Eq. 3. The attributes $A_t$ represent the essential characteristics of the object captured in image $I_t$, which are relevant for the recognition task. The prior knowledge we have about these attributes, associated to a particular action class, is given by $P(A_t | s_t)$, which represents the probability of the attributes $A_t$ given the action class $s_t$. The most suitable model for this probability depends on the application to be solved and on the type of attributes that were chosen. Thus, we let the modeling of this probability constitute one of the degrees of freedom of our framework, to be performed according to the application at hand. In order to allow attribute extraction by variational image segmentation, we require that this probability be modeled by a function $P(A_t | s_t = S_i)$ which is differentiable with respect to $A_t$. A modeling example for this probability will be offered in Section 6, where we present an implementation of our framework for a finger-spelling application.

The probabilities $P(C_t)$ and $P(I_t | A_t, C_t)$ in (1) and (3) constitute a probabilistic attribute extraction model based on segmentation. The prior contour probability $P(C_t)$ is a free parameter of our framework, which gives us the possibility to include (application-dependent) prior knowledge about the target object contour, which is independent of the action class. A common choice for this probability favors a short length $|C_t|$ of the segmenting contour, creating a smoothing effect over the contour during segmentation:

$$P(C_t) \propto e^{-\nu |C_t|}, \quad \nu > 0.$$

(4)

Moreover, $P(I_t | A_t, C_t)$ corresponds to a generative image formation model. This model states that, given a set of prior attributes $A_t$ and a prior contour $C_t$, an image $I_t$ can be obtained by sampling from the distribution $P(I_t | A_t, C_t)$. In other words, this means that we focus on the attributes and object contour only, and consider all the other properties of the image as resulting from random variations. The distribution $P(I_t | A_t, C_t)$ represents the probability of observing image $I_t$, given that $C_t$ is the boundary of the object of interest and $A_t = f_A(I_t, C_t)$ are the attributes extracted from the image via the function $f_A$. Since $f_A$ is a deterministic function of $I_t$ and $C_t$, we need to give it a probabilistic interpretation in order to be able to incorporate it into our model.
A simple approach is to consider that the probability of observing an image $I_t$ whose extracted contour is $C_t$ and whose extracted attributes $A_t$ are different from $f_A(I_t, C_t)$, is zero. Formally, this can be achieved by defining

$$P(I_t|A_t, C_t) \propto \delta(A_t - f_A(I_t, C_t)) e^{-E_{\text{image}}(I_t, C_t)},$$

(5)

where $\delta$ represents a Dirac distribution, which selects the images with the right attributes. Moreover, $E_{\text{image}}$ is a free parameter of our framework, given by a functional which expresses image-based constraints on the segmentation contour. Based on the variational image segmentation paradigm [23], such a functional is designed so that its minimum is attained when the desired constraints are fulfilled. For instance, this functional can indicate that the target object is distinguishable from the background based on its edges (a boundary-based functional) or based on its different average intensity with respect to the background (a region-based functional). In general, the functional can be made up of any boundary- or region-based terms suitable for the application at hand (such as the ones adopted in [8] or [24]). Denoting by $\Omega \subset \mathbb{R}^2$ the image domain and by $\omega \subset \Omega$ — the region inside $C_t$, a typical example for $E_{\text{image}}$ is given by assuming the values of the image feature values $I(x, y)$ (which can be scalar or vectorial) to be independent and identically distributed samples of two independent random processes, corresponding to the object and background region, respectively:

$$E_{\text{image}}(I_t, C_t) = \int \int_{\omega} - \log P(I_t(x, y)|(x, y) \in \omega) \, dx \, dy$$

$$+ \int \int_{\Omega\setminus\omega} - \log P(I_t(x, y)|(x, y) \in \Omega \setminus \omega) \, dx \, dy.$$  

(6)

A common modeling choice for the region probabilities $P(I_t(x, y)|(x, y) \in \omega)$ and $P(I_t(x, y)|(x, y) \in \Omega \setminus \omega)$ is the Gaussian distribution. Concrete modeling examples for the application-dependent parameters of our framework, i.e., $P(A_t|s_t), P(C_t)$ and $E_{\text{image}}(I_t, C_t)$, will be offered in Section 6.

### 2.3 The Algorithm for Joint Attribute Extraction and Behavior Recognition

#### 2.3.1 Main Derivation

Since we perform attribute extraction by image segmentation, the joint attribute extraction / behavior recognition task becomes a task of joint segmentation and behavior recognition. The latter can be formulated as the task of finding the contours $C_{1:T}$, the action classes $s_{1:T}$ and the behavior type $b$ whose probability given the observed images $I_{1:T}$ is maximum:

$$(C_{1:T}^*, s_{1:T}^*, b^*) = \arg \max_{C_{1:T}, s_{1:T}, b} P(C_{1:T}, s_{1:T}, b|I_{1:T}).$$

(7)
This can be equivalently written as:

\[
(b^*, s_{1:T}^1, C_{1:T}^*) = \arg \max_{C_{1:T}^* s_{1:T}^1} P(I_{1:T}, C_{1:T}, s_{1:T}, b),
\]

where \( P(I_{1:T}, C_{1:T}, s_{1:T}, b) \) is obtained by integrating the joint distribution given by Eq. 1 over the attributes \( A_{1:T} \), i.e.,

\[
P(I_{1:T}, C_{1:T}, s_{1:T}, b) = \int_{A_{1:T}} P(I_{1:T}, C_{1:T}, A_{1:T}, s_{1:T}, b) \, dt.
\]

Some insight on how to solve Eq. 8 can be gained by first considering the problem of finding the likelihood of the most likely configuration \( (C_{1:T}^*, s_{1:T}^1, b^*) \):

\[
P(I_{1:T}, C_{1:T}^*, s_{1:T}^1, b^*) = \max_{C_{1:T}^* s_{1:T}^1} P(I_{1:T}, C_{1:T}, s_{1:T}, b)
\]

\[
= \max_b \max_{C_{1:T}^* s_{1:T}^1} P(I_{1:T}, C_{1:T}, s_{1:T}, b).
\]

The last equality suggests the use of an adapted Viterbi decoding strategy in order to compute the inner maximization for each behavior type \( b \in B \), followed by the maximization over the behavior type of the resulted quantities.

The structure of the DBN in Fig. 1 suggests that, considering a time moment \( t \in \{1, \ldots, T - 1\} \), the inner maximization from the last line of Eq. 10 can be written as:

\[
\max_{C_{1:T}^* s_{1:T}^1} P(I_{1:T}, C_{1:T}, s_{1:T}, b)
\]

\[
= \max_{C_{1:T}^* s_{1:T}^1} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T}|I_{1:t}, C_{1:t}, s_{t+1}, b) P(I_{1:t}, C_{1:t}, s_{t+1}, b)
\]

\[
= \max_{C_{1:T}^* s_{1:T}^1} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T}|s_{t+1}, b) P(I_{1:t}, C_{1:t}, s_{t+1}, b)
\]

\[
= \max_{C_{1:T}^* s_{1:T}^1} \max_{s_{t+1}|I_{1:t}, C_{1:t}, s_{t+1}, b} P(s_{t+1}|I_{1:t}, C_{1:t}, s_{t+1}, b) P(I_{1:t}, C_{1:t}, s_{t+1}, b)
\]

\[
= \max_{C_{1:T}^* s_{1:T}^1} P(I_{t+1:T}, C_{t+1:T}, s_{t+2:T}|s_{t+1}, b) \max_{b_{t+1}} P(s_{t+1}|b_{t+1}) \max_{b_{t+1}} P(I_{1:t}, C_{1:t}, s_{t+1}, b).
\]

For the second equality, we used the fact that, according to the DBN of Fig. 1, the future observations \( I_{t+1:T} \), contours \( C_{t+1:T} \) and actions classes \( s_{t+2:T} \) are independent of any past quantity once \( s_{t+1} \) is known. Similarly, for the fourth equality, we used the fact that \( s_{t+1} \) is independent of the past images, contours and action classes once \( s_t \) is known. The probability \( P(I_{1:t}, C_{1:t}, s_{1:t}, b) \) from
Eq. 11 can be written as:

\[ P(I_{1:t}, C_{1:t}, s_{1:t}, b) = P(I_t, C_t | I_{1:t-1}, C_{1:t-1}, s_{1:t-1}, b) P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1}, b) P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1}, b) P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1}, b) \]

The optimal action class sequence for each behavior type can be retrieved as:

\[ P(I_t, C_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1}, b) P(I_{1:t-1}, C_{1:t-1}, s_{1:t-1}, b). \]

This formulation prompts us to the definition of the quantity \( \delta_t(s_t, b) \) as:

\[ \delta_t(s_t, b) = \max_{s_{t-1}} P(I_{1:t}, C_{1:t}, s_{1:t}, b). \]

According to Eq. 13, \( \delta_t(s_t, b) \) can be computed with the recursive formula:

\[ \delta_t(s_t, b) = \max_{C_t} P(I_t, C_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1}, b) \delta_{t-1}(s_{t-1}, b), \]

which is initialized by setting \( \delta_0(s_0, b) = 1 \). Therefore, we can obtain the likelihood of the most likely configuration \( \delta_T(s_1^T, b^*) \), defined by (10), by recursively estimating \( \delta_t(s_t, b) \) for each time step \( t \in \{1, \ldots, T\} \) and each action class \( s_t \in S \), and then maximizing \( \delta_T(s_T, b) \) over the action class \( s_T \) and behavior type \( b \):

\[ P(I_{1:T}, C_{1:T}, s_1^T, b^*) = \max_{s_T, b} \delta_T(s_T, b). \]

We notice therefore that the optimal behavior type \( b^* \) can be retrieved as

\[ b^* = \arg \max_b \delta_T(s_T, b). \]

The optimal action class sequence for each behavior type \( b \) can be retrieved by storing, at each time step \( t \), for each action class \( s_t \) and behavior type \( b \), the action class \( s_{t-1} \) which maximizes the right-hand side of Eq. 15. Denoting by \( \psi_t(s_t, b) \) this latter quantity, we have

\[ \psi_t(s_t, b) = \max_{s_{t-1}} P(s_t | s_{t-1}, b) \delta_{t-1}(s_{t-1}, b). \]

Then, the optimal action class sequence \( s_1^T, b^* \) corresponding to the optimal behavior type \( b^* \) is obtained by applying iteratively, and backward in time, the
\[ s^*_T = \arg \max_{s_T} \delta_T(s_T, b^*), \]
\[ s^*_t = \psi_{t+1}(s^*_{t+1}, b^*), \quad t = T - 1, T - 2, \ldots, 1. \]

Equations 15 and 19 form a Viterbi decoding algorithm \[29\] adapted to our model. A difference between our formulation and the one generally encountered in the HMM literature \[26\], is the presence of the additional maximizations over the hidden variable \( C_t \) in Eq. 15 and over the behavior type \( b \) in Eq. 17, whose result is used in Eq. 19.

According to Eq. 15, once \( s^*_1:T \) has been obtained, the most likely contour sequence \( C^*_1:T \), defined by Eq. 8, is given by
\[ C^*_t = \arg \max_{C_t} P(I_t, C_t | s^*_t). \]

Using Eq. 5, \( P(I_t, C_t | s_t) \) can be written as:
\[ P(I_t, C_t | s_t) = \int_{A_t} P(I_t, A_t, C_t | s_t) \]
\[ = \int_{A_t} P(I_t | A_t, C_t) P(C_t) P(A_t | s_t) \]
\[ \propto \int_{A_t} \delta(A_t - f_A(I_t, C_t)) e^{-E_{\text{image}}(I_t, C_t)} P(C_t) P(A_t | s_t) \]
\[ \propto e^{-E_{\text{image}}(I_t, C_t)} P(C_t) P(A_t = f_A(I_t, C_t) | s_t). \]

Using a Dirac distribution centered on the attributes \( A_t \) in \( P(I_t | A_t, C_t) \) proves to be particularly handy here because it allows us to easily integrate over \( A_t \).

2.3.2 An Approximation Towards Computational Efficiency

The maximization over \( C_t \) in Eq. 15 requires the computation of the locally most likely contour \( C^*(s_t) \) for each action class \( s_t \):
\[ C^*(s_t) = \arg \max_{C_t} P(I_t, C_t | s_t). \]

However, since the estimation of \( C^*(s_t) \) needs to be performed by image segmentation, the time costs of repeating the segmentation procedure for each action class \( s_t \) can be prohibitive. We therefore prefer to choose an alternative solution, where the segmentation of the image \( I_t \) is performed only once. Such a solution is more desirable if we wish our framework to scale well with an increasing number of action classes. A possible approach is to approximate \( \delta_t(s_t, b) \), given by Eq. 15, by
\[ \tilde{\delta}_t(s_t, b) = P(I_t, C^*_t | s_t) \max_{s_{t-1}} P(s_t | s_{t-1}, b) \tilde{\delta}_{t-1}(s_{t-1}, b). \]
or equivalently
\[\tilde{\delta}_t(s_t, b) = P(I_t, \tilde{C}_t^* | s_t) w_t(s_t, b),\] (24)

where we define
\[w_t(s_t, b) = \max_{s_{t-1}} P(s_t | s_{t-1}, b) \tilde{\delta}_{t-1}(s_{t-1}, b).\] (25)

In Eq. 23 and 24, \(\tilde{C}_t^*\) is an approximation of the most likely contour \(C_t^*\) (Eq. 20), obtained from a single segmentation of the image \(I_t\), and is given by
\[\tilde{C}_t^* = \arg \max_{C_t} \left( \max_{s_t} P(I_t, C_t | s_t) w_t(s_t, b) \right),\] (26)

or equivalently
\[\tilde{C}_t^* = \arg \max_{C_t} \left( \max_{s_t} P(I_t, C_t | s_t) \tilde{w}_t(s_t) \right),\] (27)

if we define
\[\tilde{w}_t(s_t) = \max_b w_t(s_t, b).\] (28)

Equation 27 shows that we make an approximation of the true most likely contour \(C_t^*\) for image \(I_t\), based on the currently most likely action class \(s_t\), in the light of past evidence accumulated in the \(\delta\) quantities and of the new image information given by \(I_t\). This constitutes a “greedy” technique, making a final and (most-likely) locally optimum solution based on the current existing information. The details of our segmentation method implementing (27) are presented in the next section.

The first time step of our recursive formulation (23) reads
\[\tilde{\delta}_1(s_1, b) = P(I_1, \tilde{C}_1^* | s_1) P(s_1 | b).\] (29)

Here \(\tilde{C}_1^*\) is obtained by the segmentation of the first image \(I_1\) of the sequence \(I_{1:T}\), for which no classification information regarding the current sequence is available yet \((w_1(s_1, b) = P(s_1 | b))\).

Given the fact that our segmentation method is quite sensitive to its initial conditions (as is the case with variational segmentation methods) and also the fact that we use the final segmentation contour of one image as the initial contour for the next image, it is desirable to obtain a good segmentation of the first image in the sequence. Therefore, we leave the particular segmentation method employed for the first image of a sequence as a free parameter of our framework, to be chosen depending on the application. Along the lines of our original formulation, one option is to perform this segmentation automatically, using (26), with \(w_1(s_1, b) = P(s_1 | b)\), and the variational segmentation scheme that we propose in the following section. Alternatively, one can perform the segmentation once for each possible value of \(s_1\), as in (22) and then choose the most likely contour for the first image as the one corresponding to the value of \(s_1\) which maximizes \(\max_b \delta_1(s_1, b)\) given by (15). The segmentation in this case can
also be performed by a simplification of the variational scheme presented in the next section. The most reliable method, but also the most time-consuming for the human operator, is the manual segmentation of the first image. Irrespective of the particular method that is chosen, we consider for the moment that a satisfactory segmentation $C^*_1$ of $I_1$ is available.

Similarly to our initial formulation of the Viterbi decoding algorithm, in order to be able to retrieve the optimal action class sequence $s^*_{1:T}$ corresponding to the optimal behavior type $b^*$ by backtracking, we need to store the argument fulfilling the maximization from the computation of $\hat{\delta}_t(s, b)$ (23), using:

$$\psi_t(s_t, b) = \arg \max_{s_{t-1}} P(s_t|s_{t-1}, b) \hat{\delta}_{t-1}(s_{t-1}, b), \quad s_t \in S, b \in B. \quad (30)$$

Then, the optimal action class sequence $s^*_{1:T}$ can be obtained by backtracking from $\psi_t(s_t, b^*)$:

$$s^*_T = \arg \max_{s_T} \hat{\delta}_T(s_T, b^*),$$

$$s^*_t = \psi_{t+1}(s^*_{t+1}, b^*), \quad t = T - 1, T - 2, \ldots, 1, \quad (31)$$

where

$$b^* = \arg \max_{b} \max_{s_T} \hat{\delta}_T(s_T, b). \quad (32)$$

### 3 Translation of Probabilistic Attribute Extraction into a Variational Segmentation Model

Variational segmentation [23] is a principled, mathematically sound way of performing image segmentation, which can flexibly integrate different image-based segmentation criteria (edges, intensity, color, texture) and also higher level prior knowledge about the target object(s) (e.g. shape information, expected trajectory, etc). Statistical interpretations of variational segmentation methods were offered, among others, in [32, 24, 9, 7, 10]. In the same spirit, we translate our probabilistic formulation for attribute and contour estimation (27) into a variational segmentation formulation.

Combining Eq. 27 and 21, we obtain:

$$\tilde{C}^*_t = \arg \max_{C_t} \left( \max_{s_t} e^{-E_{\text{image}}(I_t, C_t)} P(C_t) P(A_t = f_A(I_t, C_t)|s_t) \tilde{w}_t(s_t) \right)$$

$$= \arg \max_{C_t} \left( e^{-E_{\text{image}}(I_t, C_t)} P(C_t) \max_{s_t} P(A_t = f_A(I_t, C_t)|s_t) \tilde{w}_t(s_t) \right) \quad (33)$$

Towards a variational segmentation formulation, we equate the maximization with respect to the contour $C_t$ in (33) with the minimization with respect to $C_t$. 

13
of the negative logarithm of the right-hand side quantity:

\[ \tilde{C}_t^* = \arg\min_{C_t} \left( E_{\text{image}}(I_t, C_t) - \log P(C_t) \right. \]
\[ \left. - \min_{s_t} \log \left( P(A_t = f_A(I_t, C_t)|s_t) \tilde{w}_t(s_t) \right) \right). \]  

(34)

By identifying the first term of the right-hand side with an image-dependent energy term, the second one with a contour-dependent energy term, and the third one with an energy term embodying prior information offered by the recognition process, we can formulate our total segmentation energy as the sum of three energies:

\[ E(C_t, \mathcal{L}, I_t) = E_{\text{image}}(I_t, C_t) + \nu E_{\text{contour}}(C_t) + \alpha E_{\text{prior}}(C_t, \mathcal{L}, I_t). \]  

(35)

Here \( \nu \) and \( \alpha \) are positive constants which balance the contributions of the three terms to segmentation and \( \mathcal{L} = (L_1, \ldots, L_M) \) is a set of labels, which serves in the implementation of the minimization with respect to the class \( s_t \) from (34), as will be shown in the following.

As explained in Section 2, the image-dependent energy term \( E_{\text{image}}(I_t, C_t) \) can contain any region- or boundary-based energy term which suits the application to be solved. The contour dependent term \( E_{\text{contour}}(C_t) \) expresses a priori knowledge regarding the contour, generally including smoothness constraints on the contour. An example is the term limiting contour length, obtained by choosing \( P(C_t) \) as in (4), that is:

\[ E_{\text{contour}}(C_t) = |C_t|. \]  

(36)

The third term of the right-hand side of (34) is the one which incorporates prior information, provided by the recognition process. We include the minimization implied by this term within the variational segmentation formulation by means of a competition approach, motivated by [11]. To this end, we consider the following prior energy term:

\[ E_{\text{prior}}(C_t, \mathcal{L}, I_t) = -M \sum_{i=1}^M \log \left( P(A_t|s_t = S_i) \tilde{w}_t(S_i) \right) L_i^2 + \beta \left( 1 - \sum_{i=1}^M L_i^2 \right)^2, \]  

(37)

where \( A_t = f_A(I_t, C_t) \). The set of labels \( \mathcal{L} = (L_1, \ldots, L_M) \) controls the contribution to segmentation of the attribute prior information corresponding to each action class \( S_i \), according to its respective probability \( P(A_t|s_t = S_i) \tilde{w}_t(S_i) \). The label \( L_i \) is a scalar variable that varies continuously between 0 and 1 during energy minimization, according to the corresponding gradient descent evolution equation. The evolution of a label converges either to 1 (for the winning prior class \( S_i \), corresponding to the probability \( P(A_t|s_t = S_i) \tilde{w}_t(S_i) \) that has been maximized through segmentation, since it has been present in the energy (37)) or to 0 (for the other priors, whose contribution has 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 (37). Here $\beta$ is a Lagrange multiplier, updated at each energy minimization step to ensure that $(1 - \sum_{i=1}^{M} L_i^2)^2 \approx 0$, as will be explained in the following. The competition between the attribute priors of the different action classes during energy minimization means that the final estimated segmenting contour $C_t$ will be obtained by the influence of the most likely action class, in light of image evidence. Therefore, the minimization of our proposed total energy (35) with respect to the labels $L$, can be considered as the equivalent of the maximization with respect to the class $s_t$ from (27).

We minimize the total energy (35) simultaneously with respect to the segmenting contour $C_t$ and the labels $L$ using the calculus of variations and gradient descent. The contour $C_t$ is driven by image forces (region homogeneity, gradients, etc.) due to $E_{\text{image}}(I_t, C_t)$, smoothing forces due to $E_{\text{contour}}(C_t)$ and by the competing attribute priors of each action class, due to $E_{\text{prior}}(C_t, L, I_t)$:

$$\frac{\partial C_t}{\partial \tau} = \frac{\partial E_{\text{image}}(I_t, C_t)}{\partial C_t} - \nu \frac{\partial E_{\text{contour}}(C_t)}{\partial C_t} - \alpha \frac{\partial E_{\text{prior}}(C_t, L, I_t)}{\partial C_t}.$$  \hspace{1cm} (38)

Here $\tau$ is the artificial time of variable evolution. The first variations of the energies $\frac{\partial E_{\text{image}}(I_t, C_t)}{\partial C_t}$ and $\frac{\partial E_{\text{contour}}(C_t)}{\partial C_t}$ can be derived through the calculus of variations for the particular chosen forms of $E_{\text{image}}(I_t, C_t)$ and $E_{\text{contour}}(C_t)$, respectively. The third term of (38) can be written as:

$$\frac{\partial E_{\text{prior}}(C_t, L, I_t)}{\partial C_t} = -\sum_{i=1}^{M} \frac{L_i^2}{P(A_t|s_t = S_i)} \frac{\partial P(A_t|s_t = S_i)}{\partial A_t} \frac{\partial f_A(I_t, C_t)}{\partial C_t},$$  \hspace{1cm} (39)

where $A_t = f_A(I_t, C_t)$ and the derivatives $\partial P(A_t|s_t = S_i)/\partial A_t$ and $\partial f_A(I_t, C_t)/\partial C_t$ are computed according to the particular probability model and attribute employed.

Through gradient descent derivation, we obtain the following evolution equations for the labels $L_i$:

$$\frac{\partial L_i}{\partial \tau} = L_i \left( \log \left( P(A_t|s_t = S_i) \tilde{w}_t(S_i) \right) + 2\beta \left( 1 - \sum_{i=1}^{M} L_i^2 \right) \right), \hspace{0.5cm} i = 1..M.$$  \hspace{1cm} (40)

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

$$L_i = 1/\sqrt{M} = \epsilon_L, \hspace{0.5cm} \epsilon_L = 10^{-5}.$$  \hspace{1cm} (41)

The update equation for the Lagrange multiplier $\beta$ is deduced by imposing constancy of the constraint over time: $d(1 - \sum_{i=1}^{M} L_i^2)^2/d\tau = 0$, yielding

$$\beta = \frac{\sum_{i=1}^{M} L_i^2 \log \left( P(A_t|s_t = S_i) \tilde{w}_t(S_i) \right)}{2 \sum_{i=1}^{M} L_i^2 (\sum_{i=1}^{M} L_i^2 - 1)}.$$  \hspace{1cm} (42)

Thus, the segmentation of an image $I_t, t > 0$ comprises the following steps:
1. Initialize contour $C_t$ with the final estimated contour of the previous image: $C_t = \hat{C}_{t-1}^\ast$.

2. Initialize labels $L_i$ using (41).

3. while (not converged($C_t$))
   (a) Perform one contour evolution step given by (38).
   (b) Update the Lagrange multiplier $\beta$, using (42).
   (c) Perform one evolution step for each label $L_i$, $i = 1..M$ using (40).

4. end

5. $\hat{C}_{t}^\ast = C_t$.

The convergence with respect to the contour $C_t$ can be tested by verifying whether the contour rate of change falls below a predefined threshold.

4 Learning the Parameters of Our Model

Prior to testing our framework for the recognition of behavior in new image sequences, the parameters of the proposed model need to be estimated from training data. More precisely, these parameters characterize the probability distributions $P(b), P(s_1|b), P(s_{t-1}|b), P(A_t|s_t), P(I_t|A_t, C_t)$ and $P(C_t)$ from the joint distribution (1). Supposing that we have at our disposal a training set of $N$ image sequences $\{I_{1:T_1}, \ldots, I_{N:T_N}\}$ — where $T_n$ is the length of the $n$-th sequence — the training of our model consists in finding the parameter setting which maximizes the total log-likelihood of the training data, i.e.,

$$\Psi^\ast = \arg \max_\Psi \sum_{n=1}^{N} \log P(I_{1:T_n}|\Psi).$$  \hfill (43)

Here $\Psi$ denotes the set of model parameters and

$$P(I_{1:T}|\Psi) = \sum_b \sum_{s_{1:T}} \int_{A_{1:T}} \int_{C_{1:T}} P(I_{1:T}, A_{1:T}, C_{1:T}, s_{1:T}, b|\Psi).$$  \hfill (44)

Note that here we write explicitly the dependency on $\Psi$ of the joint distribution defined by (1). The summation and integration make the direct optimization difficult because they couple all the factors together.

To simplify the problem, we propose to decompose it in two parts: one corresponding to the behavior part, HMM-based, and the other one corresponding to the attribute extraction model, based on segmentation. To this end, we suppose that we can directly observe the attributes $A_{1:T}$ of the training images. This can be realized by the segmentation of the training image sequences. To favor automatic segmentation, the training sequences should contain the object of interest evolving on a simple background, while displaying similar behavior
content as the images targeted for recognition in the testing phase. Once the object attributes have been extracted from the training sequences, our problem is reduced to the training of the behavior part of the model, which can be performed through classical HMM training. To this end, we consider that the behavior type of our training sequences can also be observed. In this case, the set of parameters is reduced to the ones characterizing the HMM core of our model, i.e., the parameters of the action class initial and transition distribution \( P(s_1|b) \) and \( P(s_t|s_{t-1}, b) \) corresponding to each behavior type \( b \in B \), as well as the parameters of the attribute probability model \( P(A_t|s_t) \).

HMM training can be performed in supervised or unsupervised fashion. In the unsupervised case, the action classes corresponding to the observed attributes are considered as hidden. Dividing the training set into image sequences corresponding to each behavior type \( b \in B \), resulting into the set of \( N_k \) extracted attribute sequences \( \{A_1^{n_1}, ..., A_N^{N_k} \} \) for a behavior type \( b = B_k, k = 1..K \), the parameter estimation for \( B_k \) can be expressed as:

\[
\Psi_{Hk}^* = \arg\max_{\Psi_{Hk}} \sum_{n=1}^{N_k} \log P(A_{1:T}^n|b = B_k, \Psi_{Hk}), \tag{45}
\]

where \( \Psi_{Hk} \) denotes the set of HMM parameters for behavior \( B_k \) and

\[
P(A_{1:T}|b = B_k, \Psi_{Hk}) = \sum_{s_1:T} P(A_{1:T}, s_{1:T}|b = B_k, \Psi_{Hk}), \tag{46}
\]

with

\[
P(A_{1:T}, s_{1:T}|b = B_k, \Psi_{Hk}) = \prod_{t=1}^{T} P(s_t|s_{t-1}, b = B_k, \Psi_{Hk}) P(A_t|s_t, \Psi_{Hk}), \tag{47}
\]

\[
P(s_1|s_0, b = B_k, \Psi_{Hk}) = P(s_1|b = B_k, \Psi_{Hk}).
\]

This problem can be solved by the Expectation Maximization (EM) algorithm [13] or the Baum-Welch algorithm [2, 26]. Such an estimation yields different parameter values for \( P(A_t|s_t, \Psi_{Hk}) \) for each behavior type. Since action classes are shared among behavior types, we unify the obtained models by gathering the attributes allocated by the estimation to each action class \( s_t \in S \), followed by maximum-likelihood estimation of a unique set of parameters for \( P(A_t|s_t) \) from the gathered attributes. Fixing these parameters, the parameters of \( P(s_1|b) \) and \( P(s_t|s_{t-1}, b) \) can be re-estimated via the EM or Baum-Welch algorithm.

The alternative is supervised training, where the action classes corresponding to the attribute sequences are also considered as visible (observed). To this end, a manual classification of attribute sequences into action classes is necessary. This makes possible the individual estimation of the parameters for each of the probabilities involved \( P(s_1|b), P(s_t|s_{t-1}, b) \) and \( P(A_t|s_t) \)) by maximum likelihood. This simplification is due to the fact that by observing the action classes, we can re-write the estimation problem (45) as:

\[
\Psi^{*}_{Hk} = \arg\max_{\Psi_{Hk}} \sum_{n=1}^{N_k} \log P(A_{1:T}^n, s_{1:T}^n | \Psi_{Hk}). \tag{48}
\]
Substituting the expression of the HMM joint variable distribution (47), we obtain:

\[
\Psi^*_{H_k} = \arg \max_{\Psi_{H_k}} \left( \sum_{n=1}^{N_k} \log P(s^n_1|b = B_k, \Psi_{H_k}) + \sum_{n=1}^{N_k} \prod_{t=2}^{T_n} \log P(s^n_t|s^n_{t-1}, b = B_k, \Psi_{H_k}) + \sum_{n=1}^{N_k} \prod_{t=1}^{T_n} \log P(A^n_t|s^n_t, \Psi_{H_k}) \right),
\]

which leads to the maximum likelihood estimation, separately for the sets of parameters corresponding to each of the probabilities \(P(s_1|b), P(s_t|s_{t-1}, b)\) and \(P(A_t|s_t)\). In particular, for the initial action class distribution \(P(s_1|b)\), this estimation yields, for each action class \(s_1 \in S\), its relative frequency of occurrence at the first frame of the sequences from the training set corresponding to behavior \(b\). Likewise, for the transition probability distribution \(P(s_t|s_{t-1}, b)\), the estimation yields, for each action class pair \((s_t, s_{t-1})\), its relative frequency of occurrence among consecutive frames of the sequences from the training set corresponding to behavior \(b\). Similarly to the unsupervised case, the parameter estimation for \(P(A_t|s_t)\) is performed by gathering the attributes corresponding each action class. The supervised training method of the HMM is potentially more reliable than the unsupervised one — which is based on an automatic optimization algorithm susceptible to local minima — but also more time consuming for the human operator, due to the necessary manual labeling of the attribute sequences.

Let us now look at the training of the segmentation model parameters, i.e., the parameters of \(P(C_t)\) and \(P(I_t|A_t, C_t)\), the latter being actually the parameters of \(E_{\text{image}}(I_t, C_t)\), due to (5). A parameter example for the image-dependent segmentation energy \(E_{\text{image}}(I_t, C_t)\) is given by the intensity means corresponding to the object and background region, respectively. Such parameters can be learned from training data by maximum likelihood estimation, given appropriate segmentations of training image sequences. The learning of the parameter values for \(E_{\text{image}}(I_t, C_t)\) and \(P(C_t)\) imposes some degree of similarity (in terms of these parameters) among the images of a test sequence — since the model is fixed throughout the test sequence — and also between the images of the testing set and the ones of the training set. Some relief from this constraint would be brought by learning these parameters from the first frame of a test sequence, assuming that they remain relatively constant throughout the test sequence. The least engaging option, that we also chose in our implementation in Section 6, is to deduce and update these parameters dynamically at testing time, during the segmentation of each image. In this case, there is no need for image similarity between the training and the testing set.
5 Framework Summary

In the following, we present a schematic description of the steps involved in the use of our framework for joint attribute extraction and behavior recognition:

- **Training phase**: estimate parameters of the model from training attribute sequences, as explained in Section 4.

- **Testing phase**: perform joint attribute extraction and behavior recognition on new image sequences $I_{1:T}$:
  1. Segment first image in the sequence $I_1$, according to the options given in Section 2, resulting in contour $\tilde{C}^*_1$.
  2. Initialize $\tilde{\delta}$ variables according to (29).
  3. for $t = 2$ to $T$
     - Compute $\tilde{w}_t(s_t)$, $s_t \in S$ according to (28) and (25).
     - Estimate contour $\tilde{C}^*_t$ by segmenting image $I_t$ using energy (35).
     - The initial segmentation contour is given by $\tilde{C}^*_{t-1}$.
     - Compute $\tilde{\delta}_t(s_t, b)$ and $\psi_t(s_t, b)$, $s_t \in S$, $b \in B$, using (23) and (30).
  4. Estimate optimal behavior type using (32).
  5. Backtrack to infer the action class of each image $s^*_{1:T}$ using (31).

6 Application to Finger-spelling Recognition

6.1 Goal Description

Finger-spelling is a component of sign language which consists of manual representations of alphabet letters. For our application, we use the manual alphabet of the French-speaking part of Switzerland (Suisse Romande) [15], depicted in Fig. 2. The gestures corresponding to different letters are not easy to differentiate, with letter pairs such as (A, S), (M, N) or (R, U) easily confoundable. In this context, our goal is to perform finger-spelling recognition on a 15-word vocabulary containing country names, presented in Table 1.

With the support of the Swiss Federation for the Hearing-Impaired (Fédération Suisse des Sourds) [15], 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.

6.2 Solution Based on the Proposed Framework

The attribute we use to discriminate different gestures is the hand contour, represented via the level set function $\phi$: $f_A(I_t, C_t) = \phi(I_t, C_t)$, with $\phi: \Omega \rightarrow \mathbb{R}$
given by the signed distance function to the hand contour $C_t$. We employ the framework proposed in Section 2 to introduce constraints regarding the allowed behavior types, which correspond to the words of the given vocabulary. Each word can be decomposed into its basic components — the letters — which are shared among all words and constitute the action classes of our model. In the following, the probabilistic and segmentation models corresponding to time slice $t$ will be expressed in terms of the function $\phi$, which encapsulates the contour $C_t$.

The probability model $P(\phi | s_t = S_i)$ is based on a shape distance function between the segmenting contour and a prior contour corresponding to class $S_i$, motivated by [6]. Class-specific prior contours are computed through principal components analysis (PCA) from appropriate training data. During segmentation, these contours evolve in PCA-eigenspace in order to match image information.

Given a training set of level set functions, discretized on a rectangular grid and arranged in vector format $\{\phi_1, \ldots, \phi_n\} \subset \mathbb{R}^m$, its principal directions of variation are captured by the eigenvectors $\{e_1, \ldots, e_m\} \subset \mathbb{R}^m$ of the covariance matrix $\Sigma = \frac{1}{n-1} M M^\top$. The column vectors of the matrix $M$ are the $n$ mean-centered training level set functions, obtained by subtracting the mean $\overline{\phi} = \frac{1}{n} \sum_{k=1}^n \phi_k$ from each training sample $\phi_k$. An approximate representation of the training data can then be obtained in the reduced space of the $p < m$ eigenvectors $\{e_1, \ldots, e_p\}$ corresponding to the $p$ largest eigenvalues from the eigen-decomposition of $\Sigma$. A new level set function $\hat{\phi}$ can be approximated
with respect to the PCA eigenvectors as

$$\hat{\phi} = \overline{\phi} + E\, c,$$  \hspace{1cm} (50)

with \(E = [e_1, \ldots, e_p]\) and \(c\) being the \(p\)-dimensional vector of eigen-coefficients. This enables us to obtain the level set function of the prior contour \(\hat{\phi}\) as the continuous interpolation throughout the image domain of the discrete level set function \(\hat{\phi}\), computed with (50). Moreover, we introduce the alignment of a prior contour with respect to the current segmenting contour, in terms of similarity transformations acting on the image domain

$$h_\tau([x\ y]^\top) = s \left( \begin{array}{cc} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{array} \right) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix},$$  \hspace{1cm} (51)

where \(\tau = \{s, \theta, T_x, T_y\}\) and \(s\) represents scale, \(\theta\) is the rotation angle and \(T_x, T_y\) are the \(x\)- and \(y\)-axis translations, respectively. Thus, we obtain the level set function of the prior contour \(\hat{\phi}(c, \tau)\) from its class-specific PCA and alignment parameters \(c\) and \(\tau\), as the interpolation of

$$\hat{\phi}(c, \tau) = \frac{1}{s} \left( \overline{\phi}(h_\tau(x, y) + E(h_\tau(x, y))\, c) \right).$$  \hspace{1cm} (52)

We define a shape distance function between the current segmenting contour \(\phi\) and the prior contour \(\hat{\phi}\) (the latter parameterized by \(c\) and \(\tau\)), as

$$d(\phi, c, \tau) = \iint_\Omega \left( \hat{\phi}^2(c, \tau) |\nabla \hat{\phi}| \delta(\hat{\phi}) + \phi^2 |\nabla \phi(c, \tau)| \delta(\phi(c, \tau)) \right) \, dx \, dy,$$  \hspace{1cm} (53)

where \(\delta\) is the Dirac function. Since \(\iint_\Omega |\nabla \phi| \delta(\phi) \, dx \, dy\) represents the length of the zero level set of \(\phi\) and the level set functions are represented as signed distance functions, the first term of (53) approximates the minimal Euclidian distance to the prior contour, integrated along the segmenting contour. The second term of (53) exchanges the roles of \(\phi\) 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 probability of the segmenting contour represented by \(\phi\), corresponding to class \(S_i\), as

$$P(\phi|\,s_t = S_i) \propto e^{-d(\phi, c^i, \tau^i)}.$$  \hspace{1cm} (54)

As image- and contour-dependent terms, guiding the evolution of the main contour \(\phi\) and prior contours \(\hat{\phi}_i(c^i, \tau^i)\) (in terms of their parameters \(c^i\) and \(\tau^i\)), we use the piecewise constant Chan-Vese model [8], adapted to color images
given by the red, green and blue components \( I(x,y) = (I_R(x,y), I_G(x,y), I_B(x,y)) \):

\[
E_{\text{image}}(I_t, \phi) + \nu E_{\text{contour}}(\phi) =
\sum_{k \in \{R,G,B\}} \lambda_k \int_\Omega (I^k_t - \mu^k_\phi^+)^2 H(\phi) + (I^k_t - \mu^k_\phi^-)^2 H(-\phi) \, d x \, d y
\]

\[
+ \sum_{k \in \{R,G,B\}} \lambda_k \sum_{i=1}^M \int_\Omega (I^k_i - \mu^k_{\phi_i}^+)^2 H(\phi_i) + (I^k_i - \mu^k_{\phi_i}^-)^2 H(-\phi_i) \, d x \, d y
\]

\[
+ \nu \int_\Omega |\nabla H(\phi)| \, d x \, d y.
\]

(55)

Here \( H \) is the Heaviside function, \( \mu^k_\phi^+, \mu^k_\phi^- \) and \( \mu^k_{\phi_i}^+, \mu^k_{\phi_i}^- \) are the mean values of the \( k \)-th component of the image vector \( k \in \{R,G,B\} \) over the positive, respectively negative, regions of the level set functions \( \phi \) and \( \phi_i \). The ratio between the RGB components is given by the weights \( \lambda_k \geq 0, k \in \{R,G,B\} \).

Function \( \hat{\phi}_i = \hat{\phi}_i(c^i, \tau^i) \) is the continuously interpolated level set function of the prior contour (52), and the last term of (55) imposes smoothness of contour \( \phi \).

The prior term of the energy is obtained from (37) by substituting the probabilities \( P(A_t|s_t = S_i) \) with \( P(\phi|s_t = S_i) \) (54), yielding

\[
E_{\text{prior}}(\phi, L, c^{i=1..M}, \tau^{i=1..M}) = \sum_{i=1}^M \left( -\log \tilde{w}_i(S_i) + d(\phi, c^i, \tau^i) \right) L_i^2 + \beta \left( 1 - \frac{M}{\sum_{i=1}^M L_i^2} \right)^2.
\]

(56)

A gain in computational time and improved convergence towards the optimal prior is obtained by employing only the top most probable 4 priors to guide the segmentation of each image (instead the available \( M = 18 \) priors). These priors are selected using the maximum prior class probabilities, computed with (28).

The total energy (35), summing (55) and (56), is minimized via the calculus of variations and gradient descent, yielding the corresponding evolution equations for the level set function \( \phi \), the labels \( L \), the PCA and alignment parameters \( c^i \) and \( \tau^i \).

6.3 Database and Training of the Model

We 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 3 presents images from the training sequences.

First, the gesturing hand was segmented in each training sequence and the resulting contours were assigned to their respective letter classes and aligned with respect to similarity transformations (scale, rotation and translation) using genetic algorithms [12]. Subsequently, the parameters of the observation
probability model $P(A_t|s_t = S_i) = P(\phi|s_t = S_i)$ (54) for each letter class $S_i$ were learned by PCA ($p = 7$) separately from class-specific training contours. This resulted in a mean $\Phi_i$ and eigenvectors $E_i$ for each letter class $S_i$.

Afterwards, the action class initial and transition distributions $P(s_1|b)$ and
$P(s_t|s_{t-1}, b)$ were learned separately for each behavior type (word) $b$, from specific training sequences. These probability distributions were estimated as being the occurrence frequencies of starting classes and of transitions between classes from the training sequences.

6.4 Experimental Results

The obtained trained model was tested using 10 repetitions of each vocabulary word, finger-spelt by the same person, by courtesy of the Swiss Federation for the Hearing Impaired. This time, we have considered realistic conditions, involving a cluttered background, normal gesturing speed and changed lighting conditions with respect to the training image sequences. Our collaborative setting has enabled us to obtain accurate results for both segmentation and recognition, in spite of the task complexity.

In Figures 4, 5, 6, 7 and 8, rows 1 — 3, we present examples of collaborative segmentation and behavior recognition on five image sequences, which are correctly recognized by our framework as the words “Albania”, “Belarus”, “Denmark”, “Ecuador” and “Estonia” respectively. The recognition framework helped orient segmentation towards the correct action classes at each time instance. Moreover, the dynamical PCA-based class prior models adapted to significant shape variations within behavior classes, allowing the segmentation of the hand in difficult cases of cluttered background. The frame-wise behavior recognition results for these sequences, yielded by backtracking for the winner behavior type, are presented in row 3 of each of these figures 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 recognition using the extracted contours (with the same Viterbi decoding scheme), produces completely erroneous results. Such results are presented for each of the above sequences, in Figs 4, 5, 6, 7 and 8, rows 4 — 6. In all these cases, the segmentation was mislead by the cluttered background, and as a result the sequences were mis-classified (as “Algeria”, “Belgium”, “Burundi”, “Finland” and “Ecuador”, respectively).

The variational segmentation parameters for the presented test sequences were $\alpha = 4000$, $\nu = 4000$, $\lambda_R = 1$, $\lambda_G = 0$ and $\lambda_B = 0$. 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 was determined by a rough manual initialization of the contour, followed by segmentation using only the image- and contour-based terms given by the piecewise-constant Chan-Vese
Figure 4: Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word “Albania”. Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is “Algeria”.
Figure 5: Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word “Belarus”. Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is “Belgium”.

26
Figure 6: Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word “Denmark”. Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is “Burundi”.

27
Figure 7: Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word “Ecuador”. Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is “Finland”.

28
Figure 8: Rows 1 — 3: correct segmentation and behavior recognition using our framework, demonstrated on a test sequence representing the word “Estonia”. Rows 4 — 6: erroneous segmentation and behavior recognition of the same sequence, using the traditional sequential approach. The recognized word is “Ecuador”. 
model, adapted to color images:

\[
E_{\text{image}}(I_1, \phi) + \nu E_{\text{contour}}(\phi)
= \sum_{k \in \{R,G,B\}} \lambda_k \int \int_{\Omega} (I^k_1 - \mu^k_{\phi=+} )^2 H(\phi) + (I^k_1 - \mu^k_{\phi=-} )^2 H(-\phi) \, dx \, dy
+ \nu \int \int_{\Omega} |\nabla H(\phi)| \, dx \, dy.
\]

(57)

In order to show some limitations of the chosen implementation of our framework, in Fig. 9 we present two examples of mis-classification using our method. Rows 1 and 2 present the segmentation and recognition results of an image sequence representing the word “Belgium”. This word is wrongfully classified as “Belgium”. Examining the reasons for this decision, we note the similarity of the two words in terms of the contained letters — they have 4 common letters (B, E, L, U) in identical positions within the word — and also in terms of the outlines of the rest of the letters (pairs (G, A) and (M, S)). Indeed, the first part of the word was correctly recognized as containing letters B, E, L. Further along, G was correctly segmented, but recognized as A, due to the contour similarity between the two letters. Letter I was not correctly segmented due to the strong influence of the prior information, which was inclining towards the word “Belarus”, due to the first letters recognized as B, E, L, A. Letter U was correctly segmented and recognized, being common to the two words and finally letter M, though correctly segmented, was recognized as S. Segmentation and recognition results for the second sequence, representing the word “Eritrea”, are illustrated in rows 3 — 4 of Fig. 9. This sequence has been mis-classified as “Estonia”, for reasons which are similar to the case of the previous sequence.

To interpret these results, we note that we used the same parameters for the segmentation of all images in all the test sequences. However, our experience has shown that improved results can be obtained by tuning these parameters to different test sequences. We did not consider such an approach, since it would render our method impractical to use. In the case of the above presented sequences, an important factor for the failure of our method (beside the inherent similarity of the confounded words) is the misleading of the segmentation due to the too powerful influence of prior recognition information. The remedy for this problem would consist in slightly diminishing the weight \( \alpha \) of the prior term in our segmentation energy. This would allow segmentation to better capture new letter characteristics, while receiving more moderate guidance from the recognition. For the reasons mentioned above, we did not consider such sequence-dependent parameter modifications.

To finish off the presentation of our experimental results, in Table 2 we illustrate the confusion matrix between the words in our vocabulary and the statistic recognition results per word and for the whole vocabulary. As can be seen, for only 3 words out of 15 the results are quite poor (\( \leq 50 \% \)), mainly due to problems of parameter tuning, such as the ones exemplified above. However,
Figure 9: Examples of erroneous classification using our method. Rows 1 — 2: segmentation and recognition of a sequence representing the word “Belgium”, classified as “Belarus”. Rows 3 — 4: segmentation and recognition of a sequence representing the word “Eritrea”, classified as “Estonia”.

31
Table 2: Confusion matrix. Each row corresponds to the test sequences of one of the countries in our vocabulary (represented on the left of the row). The row entries for each column contain the percentage of these test sequences which were classified as belonging to the country associated with that column (represented on top of each column). The last column of the table gives the percentage of correctly classified test sequences for each country. The figure at the end of the last row represents the total percentage of correct classification over the ensemble of the test sequences.

<table>
<thead>
<tr>
<th>Country</th>
<th>Classification (%)</th>
<th>Correct (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Albania</td>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>Algeria</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Armenia</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Belarus</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Belgium</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Burundi</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Croatia</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Denmark</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ecuador</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Estonia</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Finland</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Georgia</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Germany</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>85.3</td>
<td></td>
</tr>
</tbody>
</table>

for most words (12 out of 15), we obtain excellent recognition results (more than 80%), with a total recognition rate of 85.3%.

## 7 Conclusion

In this paper we proposed a probabilistic temporal model for performing joint attribute extraction and behavior recognition from image sequences. Our joint approach enables the sharing of all existing information resources between the two tasks, which leads to their mutual improvement. Our model was developed by formulating the double extraction / recognition problem in terms of a Dynamic Bayesian Network, which incorporates a Hidden Markov Model and a probabilistic attribute extraction model, based on segmentation. The solution to the problem was elaborated as a modified Viterbi decoding scheme, which blends recognition with segmentation along the image sequence. Guidelines and examples were provided regarding the choice of the free parameters of our framework, consisting mainly of modeling choices for the included probabilities.
Moreover, alternative learning methods for the parameters of the probability models were described. Finally, the proposed model was validated via an implementation for a finger-spelling recognition application. In this context, a comparison with the traditional approach, where the attribute extraction and recognition phases are performed separately, has shown the better performance of our joint approach.

References


