Medical Image Analysis 17 (2013) 732–745

Contents lists available at SciVerse ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Surgical gesture classification from video and kinematic data

Luca Zappella *[,1], Benjamín Béjar [1], Gregory Hager, René Vidal

Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA

ARTICLE INFO

Article history:
Available online 28 April 2013

Keywords:
Surgical gesture classification
Time series classification
Dynamical system classification
Bag of features
Multiple kernel learning

ABSTRACT

Much of the existing work on automatic classification of gestures and skill in robotic surgery is based on dynamic cues (e.g., time to completion, speed, forces, torque) or kinematic data (e.g., robot trajectories and velocities). While videos could be equally or more discriminative (e.g., videos contain semantic information not present in kinematic data), they are typically not used because of the difficulties associated with automatic video interpretation. In this paper, we propose several methods for automatic surgical gesture classification from video data. We assume that the video of a surgical task (e.g., suturing) has been segmented into video clips corresponding to a single gesture (e.g., grabbing the needle, passing the needle) and propose three methods to classify the gesture of each video clip. In the first one, we model each video clip as the output of a linear dynamical system (LDS) and use metrics in the space of LDSs to classify new video clips. In the second one, we use spatio-temporal features extracted from each video clip to learn a dictionary of spatio-temporal words, and use a bag-of-features (BoF) approach to classify new video clips. In the third one, we use multiple kernel learning (MKL) to combine the LDS and BoF approaches. Since the LDS approach is also applicable to kinematic data, we also use MKL to combine both types of data in order to exploit their complementarity. Our experiments on a typical surgical training setup show that methods based on video data perform equally well, if not better, than state-of-the-art approaches based on kinematic data. In turn, the combination of both kinematic and video data outperforms any other algorithm based on one type of data alone.

1. Introduction

Over 100 years ago, Dr. William Halsted created the first surgical residency training program in the United States. His training paradigm was extremely simple: "see one, do one, teach one". However, recent technological advances have changed the way in which some surgeries are performed. This has opened the opportunity for revisiting Halsted's paradigm in search for improved ways of training surgeons.

One of such technological advances is Robotic Minimally Invasive Surgery (RMIS), which has several advantages over traditional surgery, as shown in Menon and Tewari (2003), Abbou et al. (2001), and Lowrance et al. (2010). For example, Lowrance et al. (2010) compared the post-surgery recovery of patients who underwent RMIS to that of patients who underwent traditional surgery. One of the findings was that the former group experienced a shorter length of stay, and was less likely to receive blood transfusions or develop postoperative respiratory and miscellaneous surgical complications.

However, after a first wave of optimism about RMIS, drawbacks started to arise. In the same study, Lowrance et al. (2010) observed that RMIS was associated with an almost 2-fold increase in the odds of postoperative genitourinary complications. One of the hypotheses for this increment is the steep learning curve for surgeons who want to add RMIS to their armamentarium. In fact, even for expert surgeons, training for RMIS is often considered challenging, as reported in Lenihan et al. (2008). This is exacerbated by the fact that there is a lack of fair, objective, and effective criteria for judging the skills acquired by a trainee with an RMIS system, which could ultimately reduce the benefits of such technology.

These issues have motivated a number of approaches for automatic RMIS skill assessment and gesture classification. One of the most natural approaches is to decompose a surgical task into a series of pre-defined "atomic" gestures or *surgemes*, such as "insert needle", "grab needle", and "position needle". Fig. 1 shows sample frames from three different surgemes taken from the dataset presented in Reiley et al. (2008). The problem then becomes how to segment the task in time, recognize each surgeme, and finally assess the skill level.

Even if RMIS systems are typically equipped with cameras that record the entire procedure, to the best of our knowledge, most of the studies focused mainly on the analysis of kinematic data stored by the robot. This kind of information typically involves the posi-

* Corresponding author.
E-mail address: zappella@cis.jhu.edu (L. Zappella).
[1] These authors are contributed equally to this work.

1361-8415/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.media.2013.04.007

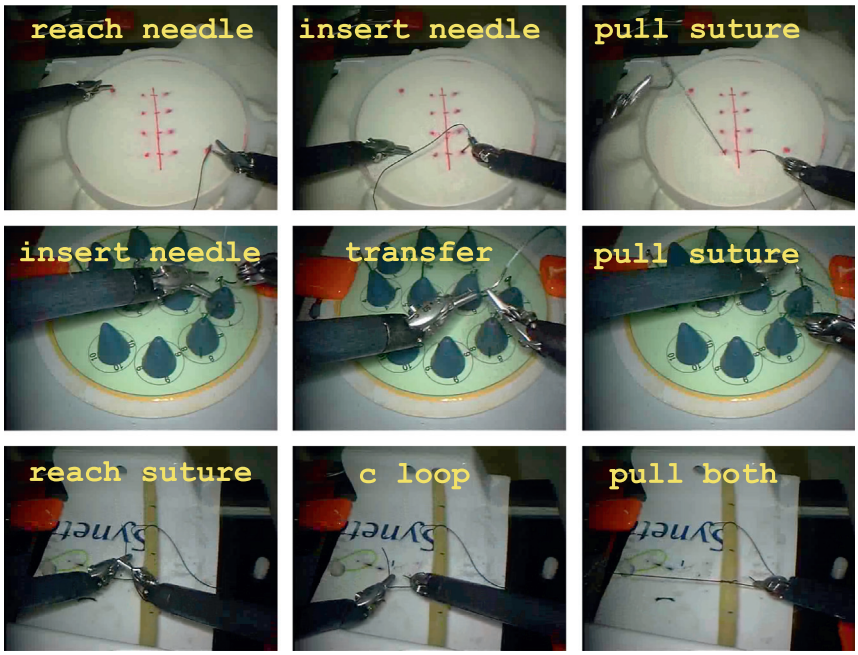

Fig. 1. Examples of different surgemes for the suturing, needle passing, and knot tying tasks.

tion of the robot tools, angles between robot joints, velocity measurements and force/torque signatures. In the medical literature, action recognition techniques from video have been applied to the analysis of the behavior of surgeons and nurses in an operating room (Miyawaki et al., 2005; Padoy et al., 2007; Blum et al., 2008). However, as far as video recognition of surgical gestures, little has been done. For example, Lin (2010) investigated some basic gesture recognition strategies from video and concluded that kinematic-based approaches were generally more accurate. In contrast, recent approaches (Lalys et al., 2011; Béjar et al., 2012; Padoy et al., 2012) show that video data can provide very high performances in automatic recognition of different surgical phases, suggesting that the above conclusion should be revisited.

In fact we argue that both video and kinematic data carry relevant and complementary information. On the one hand, kinematic data is superior to video in measuring actual 3D trajectories and 3D velocities, which is not directly measured in video. On the other hand, video data is superior to kinematic data in providing contextual semantic information such as the presence or absence of a surgical tool. The fundamental challenge is to develop efficient, robust and reliable methods for extracting such semantic information from raw pixel intensities. Indeed, it is very easy for video-based techniques to perform poorly when simple video features are used. For this reason, until recently, researchers focused mainly on kinematic data. Moreover, Automatic extraction of information from video is very challenging due to noise, occlusions and clutter, as well as the variability of tool pose and motions across tasks and surgeon expertise. Therefore, while pure video recognition could be more generally applicable to any surgery that is video recorded, when both video and kinematic data are present their complementarity should be exploited.

The aim of this paper is twofold: we propose a step towards automatic recognition of surgical gestures in video, and we present a framework for the fusion of kinematic and video data, and show that such a combination leads to higher recognition accuracy than when using only one type of data. Rather than aiming to a complete semantic interpretation of a surgical video, which is elusive at this point, we propose to use the statistical properties of features extracted from the video to build models for each gesture and use these models to classify surgical gestures in new videos. More specifically, given a video of a surgical task (e.g., suturing, needle passing, knot tying), we assume that the video has been segmented into video clips corresponding to a single gesture (e.g., position needle, drive needle through tissue, pull suture, etc.) and tackle the problem of recognizing the gesture associated with each video clip. Admittedly, this is a step backwards from existing work on kinematic data, which is able to simultaneously segment and classify gestures. However, given the limited amount of prior work on surgical gesture recognition from video, we believe that assuming known segmentation is a natural first step towards building good models for each gesture from a more complex data source.

We propose and evaluate three approaches to surgical gesture classification from video. The first approach uses linear dynamical systems (LDSs) to model the time series of features extracted from each video clip. Metrics between the parameters of the LDSs are then used to train classifiers for each gesture. Different features, such as raw pixel intensities or optical flow, and different metrics in the space of LDSs are evaluated. The second approach is a bag-of-features (BoF) approach in which a dictionary of spatio-temporal words is learned from spatio-temporal features extracted from all video clips. Each video clip is then represented with a histogram of such words and metrics between histograms are used to train classifiers for each gesture. A thorough analysis of all the components of the BoF framework, and their variations, is presented and the best performing combination of components is highlighted and discussed. The third approach combines the LDS and BoF approaches using multiple kernel learning (MKL). Since the method based on LDS is also applicable to kinematic data, we also use MKL to train classifiers that operate jointly in the kinematic and video data.

Our experiments on kinematic data show that methods based on LDSs already outperform state-of-the-art approaches based on Sparse Hidden Markov Models (SHMMs) (Tao et al., 2012). For video data, the BoF approach performs better than the LDS approach, while the MKL approach improves upon each of the individual

methods. Overall, our main conclusion is that methods based on video data perform equally well, if not better, than methods based on kinematic data for a typical surgical training setup. This result should encourage further investigation of video based techniques for surgical gesture classification as videos potentially carry more unexploited information than kinematic data. Moreover, we show that the combination of both kinematic and video data outperforms the accuracy of any algorithm that uses only one type of data.

2. State of the art

Previous work on skill evaluation in RMIS mainly exploited kinematic data recorded by the robot. Many works used global measurements of the task, such as time to completion (Datta et al., 2001; Judkins et al., 2008), speed and number of hand movements (Datta et al., 2001), distance travelled Judkins et al. (2008), and force and torque signatures (Richards et al., 2000; Yamauchi et al., 2002; Judkins et al., 2008). These methods are generally easy to implement. However, they perform a global assessment neglecting the fact that a surgical task is composed of many different gestures. Such global approaches have two main drawbacks. First, they use a single model for a complex task as a whole, while the decomposition of a task into atomic gestures allows for the use of a simpler model for each gesture. Second, they assume that a trainee is either skilled or unskilled at all gestures. In practice, different gestures have different levels of complexity, and one would expect a trainee to learn quickly how to perform simple gestures, and to require more training to perform complex ones.

To address these drawbacks, several works (see e.g., Rosen et al., 2002; McKenzie et al., 2001; Lin et al., 2006; Reiley et al., 2008) have considered the problem of decomposing a surgical task into atomic gestures, usually called surgemes. Such a decomposition not only addresses the drawbacks of global approaches, but also has the advantage of exploiting the set of rules that govern how different surgemes are related to each other. In other words, it allows one to describe a surgical task using a grammar that, for each task, describes which transitions between gestures are allowed. One can leverage this grammar to help the recognition of a surgeme, e.g., by exploiting the fact that the set of surgemes that follows an already labeled surgeme is smaller. One can also use such a grammar as an additional measure of assessment. For instance, each gesture in isolation could be executed perfectly, but the sequence of gestures may not make sense for the given task (e.g., inserting the needle before grabbing the needle). Given the many similarities with the structure of natural languages, this approach to surgical skill assessment is also known as *the language of surgery*. This approach proceeds in three steps: task segmentation, gesture recognition, and assessment of the quality of the execution and the feasibility of the sequence of gestures. Since this paper deals with the recognition phase, we will limit the discussion of previous work to those related to surgical gesture recognition.

Most of the prior work on surgical gesture recognition (see, e.g., Dosis et al., 2005; Reiley and Hager, 2009; Varadarajan, 2011) uses HMMs to analyze kinematic data stored by the robot. All these approaches model each surgeme as one or more states of an HMM. The main difference is in how these approaches model the observations within each surgeme. For example, Reiley and Hager (2009) vector-quantize the observations into discrete symbols, Varadarajan et al. (2009) use a Gaussian model combined with linear discriminant analysis (LDA), Varadarajan (2011) assumes that the observations are generated from a lower-dimensional latent space using Factor Analyzed HMMs (FA-HMMs) and Switched Linear Dynamical Systems (SLDSs), Leong et al. (2006) use a Gaussian mixture model (GMM), and Tao et al. (2012) model the observations as a linear combination of atomic motions with sparse coefficients. All of these models have significantly improved surgical gesture classification over a standard HMM.

Early work on video data analysis, such as Miyawaki et al. (2005), focus on recognizing the (coarse) phases of a surgery by also observing surgeons and nurses in the operating room. In Klank et al. (2008) an automatic feature extraction mechanism from video is proposed based on genetic programming. They use the extracted features to classify the (coarse) phases of a surgery but the average recognition accuracy is around 50%. The work in Blum et al. (2010) and Padoy et al. (2012) propose to recognize the different coarse phases of a surgery (e.g. CO_2 inflation, abdominal suturing, etc.) using laparoscopic videos. For example, the work in Padoy et al. (2012) uses binary signals that indicate the presence or not of a set of tools in the operating room. Using those signatures they use Dynamic Time Warping (DTW) and HMMs in order to classify new sequences. Also in Lalys et al. (2011) an application-dependent framework for the recognition of high-level surgical phases is proposed. The method applies DTW and HMMs on top of a set of SVM classifiers. In a recent contribution, (Lalys et al., 2013), the same authors extend their approach in order to provide additional granularity by further decomposing each of the surgical phases into basic actions. The authors combine the approach in Lalys et al. (2011) together with the detection of tools and organs in order to determine the surgical action being performed. A recognition accuracy around 64% on a frame by frame basis can be achieved with the proposed technique when applied to cataract surgeries. A limitation of the methodology is that it is application dependent and needs to be tuned to target a specific type of intervention. It would be desirable to have a general methodology that can be abstracted from the surgery at hand and that is based on the recognition of elementary actions that can be used to describe almost any surgery.

An attempt to automatic classification of skill and surgical gestures (rather than coarse phases) from video is that of Lin (2010). Lin (2010) uses different types of HMMs where the observation is the histogram of optical flow concatenated with the mean flow computed in spatially separated regions of the image. The conclusion of this study is that kinematic-based approaches are generally more accurate that vision-based methods. We argue that such a conclusion can be revised if other visual features are extracted (for example histogram of gradients) and, more importantly, if visual features are extracted around salient points rather than from the overall image.

3. Classification using linear dynamical systems

In this section, we describe our first approach to surgical gesture classification from video data. We assume we are given a collection of videos from different surgical tasks executed by different subjects with different skill levels. We assume that each video corresponds to a single surgical task and that each video is segmented into *video surgemes*. The label of each video surgeme is assumed to be known during training. In Section 3.1 we show how to model the observations from each video surgeme as the output of a linear dynamical system (LDS). In Section 3.2 we describe metrics in the space of LDSs which can be used to compare how similar two video surgemes are. In Section 3.3 we show how these metrics can be used to classify new video surgemes.

3.1. Linear dynamical systems

We denote the observed signal at time instant t as $\mathbf{z}_t \in \mathbb{R}^p$, e.g., the image intensities of a video frame with p pixels at time t, or the kinematic data (position and velocities of the robot joints) at time t. We assume that $\mathbf{z}_t$ is the output of the LDS:

$$\mathbf{s}_{t+1} = \mathbf{A}\mathbf{s}_t + \mathbf{B}\mathbf{u}_t \tag{1}$$

$$\mathbf{z}_t = \mathbf{C}\mathbf{s}_t + \mathbf{w}_t \tag{2}$$

where $\mathbf{s}_t \in \mathbb{R}^n$ represents an unobserved (hidden) state variable at time instant t of dimension $n \ll p$, and $\mathbf{B} \in \mathbb{R}^{n\times m}$ is a noise-coloring matrix that captures the correlation of the driving noise process $\mathbf{u}_t \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I})$ (Gaussian with zero mean and identity covariance). The state-transition matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ describes the dynamics of the hidden state while the observation matrix $\mathbf{C} \in \mathbb{R}^{p\times n}$ maps the hidden state to the observations. The measurement noise sequence $\mathbf{w}_t$ is also Gaussian with zero mean and covariance $\mathbf{R}$, i.e., $\mathbf{w}_t \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{R})$.

An LDS model $\mathscr{M}$ is then represented by the tuple $\mathscr{M} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{R})$. Notice, however, that this representation is not unique. This is because an equivalent representation can be found by a change of coordinates of the state variable. More specifically, if we define $\mathbf{s}_t = \mathbf{T}\mathbf{r}_t$, where $\mathbf{T} \in \mathbb{R}^{n\times n}$ is non-singular, then the two representations $\mathscr{M} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{R})$ and $\tilde{\mathscr{M}} = (\mathbf{T}^{-1}\mathbf{A}\mathbf{T}, \mathbf{T}^{-1}\mathbf{B}, \mathbf{C}\mathbf{T}, \mathbf{R})$ are equivalent (i.e., both represent the same process $\mathbf{z}_t$). This consideration will be important when comparing two video surgemes, because we cannot directly compare the parameters.

Nonetheless, given a sequence of observations $\{\mathbf{z}_t\}$ corresponding to a single video surgeme, we can identify a representation $\mathscr{M} = (\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{R})$. Since the number of pixels p can be large, we use a sub-optimal, but computationally efficient, method based on Principal Component Analysis proposed in Doretto et al. (2003). Our set of N training examples is hence represented by a set of N LDS parameters $\{\mathscr{M}_i\}_{i=1}^N$.

3.2. Metrics for comparing LDSs

Once we have represented each video surgeme with an LDS model, we need a dissimilarity metric or distance to assess how close two given models are. Several metrics can be found in the literature, such as distances based on the Binet-Cauchy kernels (Vishwanathan et al., 2007), probabilistic metrics based on the KL-divergence (Chan and Vasconcelos, 2005), or metrics such as the Martin distance (e.g., see Cock and Moor, 2002; Martin, 2000) based on the subspace angles between the observability subspaces of the dynamical models. Recently, Afsari et al. (2012) defined a pseudo-distance between LDSs based on the equivalence of representations between models. In what follows, we describe some of these metrics in more detail.

3.2.1. Metrics based on the subspace angles

Martin (2000) proposed a distance between Single-Input Single-Output (SISO) Auto-Regressive Moving Average (ARMA) processes based on comparing their cepstrum. Cock and Moor (2002) extended this distance to the case of Multiple-Input Multiple-Output (MIMO) ARMA models by using the principal angles between the observability subspaces of the models, which depend only on the parameters $\mathbf{A}$ and $\mathbf{C}$ of the LDSs.

More specifically, let $\mathscr{M}_i = (\mathbf{A}_i, \mathbf{C}_i)$ for $i = 1, 2$ be the parameters of two LDS models of order n. Let $\theta_1, \ldots, \theta_{2n}$ be the subspace angles between the range spaces of their infinite observability matrices O_1 and O_2, which are defined as

$$O_i = [\mathbf{C}_i^\top, (\mathbf{C}_i\mathbf{A}_i)^\top, (\mathbf{C}_i\mathbf{A}_i^2)^\top, \ldots], \quad i = 1, 2. \tag{3}$$

If the systems are stable, i.e., $\|\mathbf{A}_i\|_2 < 1$, the subspace angles θ_i can be computed as the roots of $\theta_i = \cos^{-1}(\sqrt{\lambda_i})$, where λ_i is the i-th eigenvalue of $\mathbf{P}_{11}^{-1}\mathbf{P}_{12}\mathbf{P}_{22}^{-1}\mathbf{P}_{21}$ and $\mathbf{P}_{ij}$ is the solution to the Sylvester's equation

$$\mathbf{P}_{ij} = \mathbf{A}_i^\top \mathbf{P}_{ij}\mathbf{A}_j + \mathbf{C}_i^\top \mathbf{C}_j \quad i, j = 1, 2. \tag{4}$$

One can show that the subspace angles are invariant with respect to a change of basis in the state space. Thus, as described in Cock and Moor (2002), one can define many distances based on the subspace angles. For example, the (squared) Martin and Frobenius distances between the models $\mathscr{M}_1$ and $\mathscr{M}_2$ are, respectively, given by:

$$d_M^2(\mathscr{M}_1, \mathscr{M}_2) = -\log \prod_{i=1}^{2n} \cos^2(\theta_i), \tag{5}$$

$$d_F^2(\mathscr{M}_1, \mathscr{M}_2) = 2\sum_{i=1}^{2n} \sin^2(\theta_i). \tag{6}$$

3.2.2. Determinant kernel

Some classification algorithms, such as support vector machines (Vapnik, 1998), rely not only on distances but also on kernels. For this purpose, Vishwanathan et al. (2007) introduced a family of kernels for LDSs called the *Binet-Cauchy* (BC) kernels. Such kernels depend on not only on the parameters $(\mathbf{A}, \mathbf{C})$, but also the initial condition $\mathbf{s}_0$. Since in our particular application the initial state should not affect the classification of the different surgemes, we will use one special case of BC kernel, called the (normalized) determinant kernel, which was proposed by Chaudhry and Vidal (2009). This kernel is independent of the initial conditions and also invariant with respect to basis transformations. The normalized determinant kernel is given by

$$\kappa_D(\mathscr{M}_1, \mathscr{M}_2) = \frac{(\det(\mathbf{P}_{12}))^2}{\det(\mathbf{P}_{11})\det(\mathbf{P}_{22})}, \tag{7}$$

where $\mathbf{P}_{ij}$ is the solution to the Sylvester's equation

$$\mathbf{P}_{ij} = \rho\mathbf{A}_i\mathbf{P}_{ij}\mathbf{A}_j + \mathbf{C}_i^\top\mathbf{C}_j \tag{8}$$

with $0 < \rho < 1$ being a parameter. A distance can now be computed as

$$d_D^2(\mathscr{M}_1, \mathscr{M}_2) = \kappa_D(\mathscr{M}_1, \mathscr{M}_1) + \kappa_D(\mathscr{M}_2, \mathscr{M}_2) - 2\kappa_D(\mathscr{M}_1, \mathscr{M}_2). \tag{9}$$

3.2.3. Action-induced distances

Afsari et al. (2012) proposed an alternative approach to comparing LDSs based on finding the "closest" representation between two models through a basis transformation. Instead of allowing for any arbitrary non-singular matrix transformation $\mathbf{T}$, the authors in Afsari et al. (2012) restrict themselves to the orthogonal group $O(n)$ (i.e., the set of matrices $\mathbf{T} \in \mathbb{R}^{n\times n}$ such that $\mathbf{T}^\top = \mathbf{T}^{-1}$). This allows a more tractable computation of the metric due to the compactness of $O(n)$. The distance between two models is quantified using the Frobenius norm between the model's parameters. In particular, let $\mathbf{Q} \in O(n)$ be an orthogonal matrix, then the (squared) Align metric between two models $\mathscr{M}_i$ and $\mathscr{M}_j$ is defined as

$$d_A^2(\mathscr{M}_1, \mathscr{M}_2) = \min_{\mathbf{Q}\in O(n)} \Big\{ \lambda_A \|\mathbf{Q}^\top\mathbf{A}_1\mathbf{Q} - \mathbf{A}_2\|^2 + \lambda_C \|\mathbf{C}_1\mathbf{Q} - \mathbf{C}_2\|^2 + \lambda_B \|\mathbf{Q}^\top\mathbf{B}_1 - \mathbf{B}_2\|^2 \Big\}, \tag{10}$$

where $\lambda_A \geqslant 0, \lambda_B \geqslant 0$, and $\lambda_C \geqslant 0$ are parameters that weight the contribution of each of the terms in (10).

3.3. Classification of LDSs

Once a metric is selected, a common approach to classify novel sequences is to use a k-Nearest Neighbors (k-NNs) classifier or a kernel Support Vector Machine (SVM) (Schölkopf and Smola, 2002). In this work, we will use a Radial Basis Function (RBF) kernel between dynamical models. That is

$$\kappa_{RBF}(\mathscr{M}_i, \mathscr{M}_j) = e^{-\gamma d_X^2(\mathscr{M}_i, \mathscr{M}_j)}, \tag{11}$$

where d_X is a particular metric (e.g., Martin, Frobenius, Align or Determinant) and $\gamma > 0$ is a parameter.

4. Classification using bag of spatio-temporal features

Our second approach to surgical gesture classification is based on the Bag of Features (BoF) approach to object recognition (Csurka et al., 2004; Sivic and Zisserman, 2003). This approach is composed of several steps: (i) extracting some salient features from images of different objects, e.g., SIFT features (Lowe, 1999); (ii) clustering these features to form a bag of visual words (also known as dictionary or codebook); (iii) encoding the set of features extracted from an image (typically a histogram of quantized features); and (iv) training a classifier to recognize different objects based on their encoded description. The BoF approach can also be applied to action recognition in videos. The most direct way to do so is to build a histogram (encoding step) for each video, where features are extracted from groups of frames rather than from a single image (see, e.g., Laptev, 2005; Willems et al., 2008; Chaudhry et al., 2009).

Each of the four steps of the BoF framework can be implemented using a variety of techniques. However, a priori it is not possible to identify which is the best combination of techniques: for different problems the best combination might change. In Chatfield et al. (2011) a detailed explanation of the most popular choices used to build a BoF framework for the task of object recognition was presented. In the following we present a similar analysis for the task of surgical action recognition in which each of the four steps of the BoF approach is analyzed in detail.

4.1. Features

In the case of surgical gesture recognition, we extract features from multiple cuboids inside each video surgeme, where each cuboid is centered at a Space–Time Interest Point (STIP) (Laptev, 2005). A STIP is a point (x, y, t) where the video has significant variations (i.e., large gradients) both in space and in time (as opposed to uniform regions). Such salient points can be detected for a fixed set of multiple spatio-temporal scales. Moreover, STIP are always detected in correspondence of motion, as shown in Fig. 2, thus most of the information contained in the static background is automatically discarded.

A 3D cuboid is then centered around each of the detected points at the space–time scale where the point was found. The local information contained in the cuboid is used to build a 72-bin histogram of oriented gradients (HOG) and a 90-bin histogram of optical flow (HOF), as described in Wang et al. (2009). In the experimental section we will test the HOG and HOF features separately, and we will show the benefit of combining them either by concatenation, producing thereby a unique feature vector of size 162, or by a multi-channel approach, similar to the one used in Zhang et al. (2007).

4.2. Clustering

Once the features have been extracted a codebook needs to be built. The codebook is built such that similar features will be identified as the same visual word. This clustering stage has two purposes. On the one hand, it reduces the dimensionality of the problem, since a video can now be described as a set of words (which is typically smaller than the set of features). On the other hand, some robustness is achieved with respect to small variations of the features. This step is usually performed by k-means. However, recently a method based on sparse dictionary learning (SDL) optimization has been proposed by Yang et al. (2009).

4.2.1. k-Means

Let $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_G]^\top \in \mathbb{R}^{G \times D}$ be a matrix whose rows are G feature descriptors of dimension D. The k-means clustering consists of finding a set of centroids (also called words) $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]^\top$ such that:

$$\underset{\mathbf{V}}{\operatorname{argmin}} \sum_{g=1}^{G} \min_{k=1,\ldots,K} \|\mathbf{f}_g - \mathbf{v}_k\|_2^2, \tag{12}$$

where $\|\cdot\|_2$ is the ℓ_2 norm. Once the codebook is built, the distance between a feature and each word can be easily computed. Such a distance will be used in the encoding step.

4.2.2. Sparse dictionary learning

A different approach was proposed in Yang et al. (2009), where the codebook is learnt so that each feature can be reconstructed as a sparse linear combination of the words in the codebook. Hence, the problem is re-formulated as:

$$\underset{\mathbf{V},\mathbf{Y}}{\arg\min} \quad \sum_{g=1}^{G} \|\mathbf{f}_g - \mathbf{y}_g \mathbf{V}\|_2^2 + \lambda \|\mathbf{y}_g\|_1 \tag{13}$$

$$\text{subject to} \quad \|\mathbf{v}_k\| \leqslant 1, \quad \forall\ k = 1, 2, \ldots, K \tag{14}$$

where $\lambda \in \mathbb{R}^+$ weights the sparseness term, $\|\cdot\|_1$ is the ℓ_1 norm, and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_G]^T$ is a matrix, where each $\mathbf{y}_g$ contains the cluster membership of feature $\mathbf{f}_g$ with respect to all the words of the codebook $\mathbf{V}$. This problem can be solved by alternating between two steps: first fix the dictionary $\mathbf{V}$ and solve a LASSO problem, as defined in Tibshirani (1994), to find $\mathbf{Y}$, then fix $\mathbf{Y}$ and solve the dictionary update problem. When the dictionary $\mathbf{V}$ is finally built, and a new feature $\mathbf{f}$ has to be encoded, the problem of Eq. (13) reduces to the LASSO problem.

4.3. Encoding

Once the codebook is obtained, each video clip (containing a specific action) can be represented in terms of such a codebook. Typically a histogram representation is used, where each bin of the histogram corresponds to a word of the codebook. At this stage two choices can be made regarding thresholding and pooling.

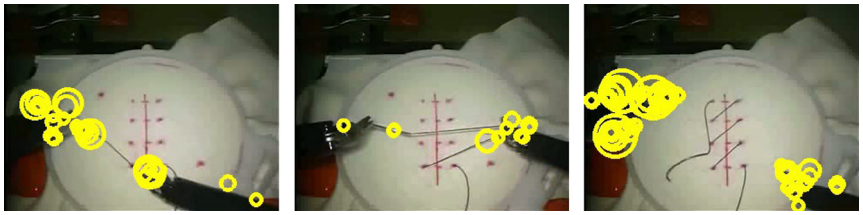

Fig. 2. Examples of detected STIPs during a suturing task.

4.3.1. Hard, soft, and hybrid thresholding

The thresholding decision can be seen as the way in which each feature "votes" for the words of the codebook. If hard-thresholding is used the vote is binary (i.e., 1 or 0) and each feature is associated only with one word. Hence, each feature will cast its vote for the closest word (in the case of k-means classification) or for the word with highest membership coefficient (in the case of sparse dictionary learning):

$$\mathscr{V}(\mathbf{f}_g, \mathbf{v}_k) = \begin{cases} 1 & \text{if } k = \arg\ \min_j ||\mathbf{v}_j - \mathbf{f}_g||_2^2 \\ & (K-\text{means}) \text{ or} \\ & \text{if } k = \arg\ \max_j |\mathbf{y}_g(j)| \\ & (\text{SDL}) \\ 0 & \text{otherwise} \end{cases} \qquad \forall j = 1, \ldots, K \tag{15}$$

In the case of soft-thresholding the vote of each feature is spread among all K words. The distance between a feature $\mathbf{f}_g$ and a word $\mathbf{v}_k$ can be converted to a score/vote as follows:

$$\mathscr{V}(\mathbf{f}_g, \mathbf{v}_k) = \frac{\exp(-\beta||\mathbf{f}_g - \mathbf{v}_k||_2^2)}{\sum_{j=1}^{K} \exp(-\beta||\mathbf{f}_g - \mathbf{v}_j||_2^2)} \tag{16}$$

where $\beta \in \mathbb{R}^+$ weights how hard the thresholding will be (larger β corresponds to harder thresholding). If the codebook is computed by SDL then the membership coefficients (normalized to sum to one for a given feature) can be directly used as a soft vote: $\mathscr{V}(\mathbf{f}_g, \mathbf{v}_k) = |\mathbf{y}_g(k)|/||\mathbf{y}_g||_1$.

We also propose a third thresholding strategy that can be seen as a hybrid between hard and soft thresholding: as in hard thresholding, each feature casts a vote for the "closest" word only. However, such a vote corresponds to its soft score:

$$\mathscr{V}(\mathbf{f}_g, \mathbf{v}_k) = \begin{cases} \frac{\exp(-\beta||\mathbf{f}_g - \mathbf{v}_k||_2^2)}{\sum_{j=1}^{K} \exp(-\beta||\mathbf{f}_g - \mathbf{v}_j||_2^2)} & \\ \text{if } k = \arg\ \min_j ||\mathbf{v}_j - \mathbf{f}_g||_2^2 & \\ (K-\text{means}) & \\ |\mathbf{y}_g(k)|/||\mathbf{y}_g||_1 & \forall j = 1, \ldots, K \\ \text{if } k = \arg\ \max_j |\mathbf{y}_g(j)| & \\ (\text{SDL}), & \\ 0 \text{ otherwise}, & \end{cases} \tag{17}$$

4.3.2. Sum vs Max pooling

At this point each feature has casted its vote and one histogram $\mathbf{h} \in \mathbb{R}^K$ can be built to represent the entire video surgeme. There are two ways to build the histogram: by sum-pooling or by max-pooling. In the case of sum-pooling:

$$\mathbf{h}(k) = \sum_{g=1}^{G} \mathscr{V}(\mathbf{f}_g, \mathbf{v}_k) \quad \forall k = 1, \ldots, K. \tag{18}$$

In the case of max-pooling:

$$\mathbf{h}(k) = \max\{\mathscr{V}(\mathbf{x}_1, k), \mathscr{V}(\mathbf{x}_2, k), \ldots, \mathscr{V}(\mathbf{x}_M, k)\}, \tag{19}$$

for all $k = 1, \ldots, K$.

Once the histogram has been built the last step consists of normalization of the histograms so that $\sum_{k=1}^{K} \mathbf{h}(k) = 1$. Although this last step is not a requirement it is a commonly accepted practice and it usually leads to slightly better results in terms of classification rate.

4.4. Classifier

Each video clip i is now represented by its histogram $\mathbf{h}_i$. These histograms can be used to train a one-vs-one multi-class SVM classifier. For additional details about SVM refer to Section 5. SVM can be used by feeding directly the histograms obtained in the previous step (linear SVM) or by adopting non-linear kernels. Linear kernels have a faster training phase, however, non-linear kernels tend to lead to better results. In the BoF framework the kernels typically used are the intersection and the χ^2 kernel. Given two videos i and j the intersection kernel is defined as $\kappa_I(\mathbf{h}_i, \mathbf{h}_j) = \min(\mathbf{h}_i, \mathbf{h}_j)$, where the min operator is applied for each bin of the histograms. While the χ^2 kernel is defined as $\kappa_\chi(\mathbf{h}_i, \mathbf{h}_j) = \sum_k 2\frac{\mathbf{h}_i(k)\mathbf{h}_j(k)}{\mathbf{h}_i(k)+\mathbf{h}_j(k)}$.

When Q different kind of features are extracted (in our case we have HOG and HOF descriptors, hence $Q = 2$) one can concatenate them, and therefore have one histogram for each video clip, or keep them separately and build a different dictionary for each kind of descriptor. In this last case each video clip i has Q histogram representations $\mathbf{h}_i^q$ (with $q = 1, \ldots, Q$). Each representation can be seen as a *channel*, and the histograms can be combined in order to produce a unique kernel by using a multi-channel approach similar to the one used in Zhang et al. (2007):

$$\kappa_{RBF}(\mathbf{h}_i, \mathbf{h}_j) = \exp\left(-\gamma \sum_{q=1}^{Q} \frac{1}{\mu_q} d(\mathbf{h}_i^q, \mathbf{h}_j^q)\right), \tag{20}$$

where $\gamma \in \mathbb{R}^+$ is a parameter, $d(\mathbf{h}_i^q, \mathbf{h}_j^q)$ is the χ^2 distance between the histograms $\mathbf{h}_i^q$ and $\mathbf{h}_j^q$, and μ_q is the mean distance between all pairs of training histograms for channel q.

5. Support vector machine and multiple kernel learning

In this section the principles behind SVM, kernel SVM and MKL will be revised. Recall that in a binary (two-class) SVM classification problem, we want to find an hyperplane that maximally separates the two classes (Schölkopf and Smola, 2002). Therefore, given N training samples $\mathbf{x}_i, \quad i = 1, \ldots, N$, with associated labels $y_i \in \{-1, +1\}$, the goal is to find a linear classification function of the form $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, where $\mathbf{w}$ is the separating hyperplane and b is some offset. The label assigned to sample x_i is then computed as $\hat{y}_i = sign(f(\mathbf{x}_i))$. However, for some problems it is not possible to linearly separate the two classes. In that case, one can use a kernel SVM in order to find a separating hyperplane in a different (of higher dimension) subspace. The idea behind a kernel SVM is to map the data points into a high-dimensional subspace where the data is (hopefully) linearly separable. If we denote such mapping as $\phi_k(\cdot)$, then the kernel SVM classifier can be found by solving the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}, \{\xi_i\}, b}{minimize} \quad & \tfrac{1}{2}\mathbf{w}^\top \mathbf{w} + C\sum_{i=1}^{N} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^\top \phi_k(\mathbf{x}_i) + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0, \ i = 1, \ldots, N \end{aligned} \tag{21}$$

where ξ_i are slack variables that penalize the misclassification of sample i, and the parameter C weights the penalization. Note that now, the decision function is of the form $f(\mathbf{x}) = \mathbf{w}^\top \phi_k(\mathbf{x}) + b$. It can be shown (Schölkopf and Smola, 2002) that in order to compute the optimal decision function $f(\cdot)$, it is not necessary to explicitly know the mapping $\phi_k(\cdot)$ as long as we know its associated kernel function. That is, a function of the form

$$\kappa_k(\mathbf{x}, \mathbf{y}) = \phi_k(\mathbf{x})^\top \phi_k(\mathbf{y}) \tag{22}$$

for all possible combination of points $\mathbf{x}$ and $\mathbf{y}$ in the original space.

There are different choices of kernels that can be used and some of them have already been introduced (e.g., RBF or χ^2). Different kernels may perform differently depending on their parameter values and the application at hand. An appropriate selection of the kernel and its parameters is not a trivial task and may drastically affect

the final classification performance. Imagine that we have a number of kernels $\kappa_k(\cdot,\cdot),\ k=1,\dots,N_k$. Since a nonnegative weighted combination of kernel functions is also a valid kernel (Schölkopf and Smola, 2002), a reasonable approach would be to build a new kernel that is a linear combination of the given kernels as

$$\kappa(\cdot,\cdot)=\sum_{k=1}^{N_k} d_k \kappa_k(\cdot,\cdot) \tag{23}$$

where $d_k \geqslant 0$. This is precisely the principle behind MKL where the goal is to simultaneously find the most discriminative classifier and the kernel weights. Thanks to this model we can avoid tedious parameter tuning, instead we can use many kernels with many different parameter values and let the algorithm choose the weights of each kernel that are most discriminative for the given application.

Note, that now we also have a machinery to combine kernels built with different metrics (for example different LDS distances) and possibly with different data sources (for example features from kinematic and features from video data). In fact, while LDS based approaches (either computed from kinematic or video data) capture the dynamics of the scene, the BoF approach is based on sparse (due to feature detection) local structures of the frame (captured by HOG) and very small and sparse motion (captured by HOF). Since both techniques use kernels, we can now use MKL to combine them and exploit their complementarity.

More formally, if we define

$$\phi(\mathbf{x})=\left[\sqrt{d_1}\phi_1(\mathbf{x})^{\mathsf{T}},\dots,\sqrt{d_{N_k}}\phi_{N_k}(\mathbf{x})\right]^{\mathsf{T}} \tag{24}$$

then the MKL problem can be written as

$$\begin{aligned} \underset{\mathbf{w},\{\xi_i\},b,\{d_k\}}{\text{minimize}} \quad & \tfrac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}+C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_i)+b)\geqslant 1-\xi_i \\ & \xi_i\geqslant 0,\ i=1,\dots,n \\ & d_k\geqslant 0,\ k=1,\dots,N_k \\ & \sum_k d_k=1, \end{aligned} \tag{25}$$

where the last constraint is needed in order to avoid trivial solutions. It is easy to see that $\kappa(\mathbf{x}_i,\mathbf{x}_j)=\phi(\mathbf{x}_i)^{\mathsf{T}}\phi(\mathbf{x}_j)=\sum_k d_k\phi_k(\mathbf{x}_i)^{\mathsf{T}}\phi_k(\mathbf{x}_j)$. There exist several formulations and variants of the MKL problem, see Gönen and Alpaydin (2011) for a complete review. One of such alternative approaches is the one presented in Varma and Babu (2009) where a regularization term is added in order to penalize large weights. In this paper we follow this approach, hence, the MKL objective can be written as

$$\begin{aligned} \underset{\mathbf{w},\{\xi_i\},b,\{d_k\}}{\text{minimize}} \quad & \tfrac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}+C\sum_{i=1}^{N}\xi_i+r(\mathbf{d}) \\ \text{subject to} \quad & y_i(\mathbf{w}^{\mathsf{T}}\phi(\mathbf{x}_i)+b)\geqslant 1-\xi_i \\ & \xi_i\geqslant 0,\ i=1,\dots,n \\ & d_k\geqslant 0,\ k=1,\dots,N_k \end{aligned} \tag{26}$$

where $\mathbf{d}=[d_1,\dots,d_{N_k}]^{\mathsf{T}}$ and $r(\cdot)$ is a regularizer (ℓ_1 or ℓ_2 norm) on the kernel weights. Note that if we fix the kernel weights d_k, then the problem reduces to a standard (kernel) SVM problem as in (21). Therefore, in order to solve for problem (26), the authors in Varma and Babu (2009) alternatively solve the SVM problem with fixed weights, and the optimization of the weights given the current classifier $\mathbf{w}$. This iterative procedure is repeated until convergence or until a maximum number of iterations has been reached, see Varma and Babu (2009) for further details.

Until now we assumed a binary classification problem. Since our problem is a multi-class classification one, we adopt the one-versus-one strategy with majority voting for classifying the surgemes. In the experimental part we will use MKL to combine kernels coming from video data only, kinematic data only, as well as hybrid combinations of kernels coming from both video and kinematic.

6. Experiments

6.1. Surgical data

For our tests we used the dataset presented in Reiley et al. (2008). The data was collected under IRB at Intuitive Surgical and consists of three different tasks: suturing (SU, 39 trials), needle passing (NP, 26 trials) and knot tying (KT, 36 trials). Each task is performed by eight subjects with different skill levels (expert, intermediate and novice). Typically each user performed around 3–5 trials for each task. Each trial lasts, on average, 2 min and both kinematic and video data are recorded at a rate of 30 frames per second. Kinematic data consists of 78 motion variables (positions, rotation angles, and velocities of the master/patient side manipulators), whereas video data consists of JPEG images of size 320×240.

The data was manually segmented based on the surgeme's definition of Reiley et al. (2008). Specifically, the vocabulary of possible atomic actions consisted of 15 surgemes: (1) reaching for needle with right hand, (2) positioning needle, (3) pushing needle through tissue, (4) transferring needle from left to right, (5) moving to center with needle in grip, (6) pulling suture with left hand, (7) pulling suture with right hand, (8) orienting needle, (9) using right hand to help tighten suture, (10) loosening more suture, (11) dropping suture at end and moving to end points, (12) reaching for needle with left hand, (13) making 'C' loop around right hand, (14) right hand reaches for suture and (15) both hands pull. Note that, although there are a total of 15 surgemes, not all of them appear in a given task. For example, suturing typically involves 10 of these 15 surgemes, needle passing involves nine surgemes, and knot tying involves six surgemes. A typical suturing trial is a collection of about 20 video clips, while a needle passing has an average of 13 video clips, and knot tying is composed of about nine video clips.

In order to compare the accuracy of the surgeme recognition task, we created two different test setups. The first setup is the *leave-one-super-trial-out* (LOSO), where we leave one trial for each one of the users out for testing. The second setup is the *leave-one-user-out* (LOUO), where we leave all the trials from one user out for testing. For each task we performed a training and a test phase using only the surgemes that appeared in that task.

6.2. Results of the linear dynamical systems approach

In this section we present the results for surgeme classification when using LDSs for modeling the different gestures. We use the manually segmented surgemes to fit an LDS model to the data and use the metrics described in Section 3 to classify novel sequences using either 1-NN or SVMs with RBF kernels (one-versus-one multi-class classification). For system identification, we employ the PCA-based approach of Doretto et al. (2003). We apply this technique to both video and kinematic data. In the case of video data, we use the raw pixel intensities as well as optical-flow extracted from the videos with a downsampling factor of 2.

From the identified dynamical models, we compute all the pairwise distances between the models and scale them using a sigmoidal function. Since different metrics may have different scales and ranges, the scaling procedure serves as a normalization step so that all the distances lie within the interval [0 1].

It is important to mention that in our dataset different surgemes have a different number of occurrences. In order to avoid over-fitting in favor of the most represented surgemes in the surgical trials, we randomly sample a small subset of them so that

the training phase is more balanced. In particular, we randomly sample no more than 40 surgemes per class and average the results over five repetitions. The SVM penalty parameter C as well as the γ parameter of the RBF kernel are chosen using 3-fold cross-validation over the training set. That is, for each random sample of the training set we randomly split it into three disjoint subsets and use two of them for training and the remaining one for testing.

The performance is measured as the percentage of correctly identified surgemes averaged over all tests and repetitions for each setup.

6.2.1. Effect of the order of the dynamical model

In order to see the effect of the order of the dynamical models on classification performance, we have varied it from $n = 5$ up to $n = 21$ with an increase-step of two in our simulations. Since we have observed the same trend in all tasks, we only report here the results for the suturing task, but the general conclusions also apply to the other tasks. In Fig. 3 we have depicted the average classification rate obtained for the considered metrics when varying the order of the dynamical models. The plots on each column correspond to a different type of data, starting with optical-flow (left), followed by pixel intensities (middle), and kinematic data (right). The first two rows of plots correspond to the LOSO setup while the last two represent the LOUO setup. The first and third row correspond to SVM classification while the remaining two correspond to the average classification rate when using a 1-NN classifier.

It can be observed from Fig. 3 that the Determinant metric generally degrades when increasing the order of the system for the case of SVM classification. When using 1-NN classifier it does well over the considered range of orders and for all the data types with the exception of pixel intensities with order above 17. The rest of the metrics appear to be more or less constant for orders above nine or slightly increasing in some cases. This last observation is particularly true for the Align metric over the pixel intensities where a small increase in performance follows with an increase of the order. For kinematic data, however the Align metric exhibits a negative slope with the increasing order of the LDS. Different metrics exhibit a different behavior with different input data. We cannot extract a single best order from the plots in Fig. 3. Nevertheless, if we were to choose one order, it seems to be reasonable to select one that lies between, say 10 and 17, where almost all metrics perform well in all scenarios.

6.2.2. Effect of the LDS metrics and feature types

From the experiments we can also see that different metrics work differently over different input data (see Fig. 3). We have displayed in Table 1 the best average classification results for different data types (optical-flow, pixel intensities or kinematics) and classification strategies (SVM and 1-NN). We can observe that, in general, metrics based on subspace angles (i.e., Frobenius and Martin) provide the best performance either using 1-NN or SVM for classification. However, all metrics perform similarly, with the only exception being the Determinant distance in the SVM case, which showed a significantly lower performance that the other metrics.

We can also observe from Table 1 that when using pixel intensities to fit the LDSs, the Align metric seems to work really well. It provides the best performance in the LOSO setting with 1-NN classifier and it is also close to the best one in the LOUO setup. In the SVM case, it does consistently better than all the previous metrics providing in some cases, an improvement of around 8% in accuracy.

In contrast, when dealing with kinematic data, the best performing metric is the Frobenius distance, which provides the highest accuracy over all setups and classification methods, and in some cases it provides an increase of around 7% in classification accuracy.

Given the results presented in Table 1 there is no metric that is a clear winner over the other metrics for all kind of input data. However, the Frobenius metric worked quite well in all the setups and for all data types. In contrast, the Determinant distance provided good results using 1-NN classifier while its performance was significantly lower than that of the other metrics in the SVM case. We also observe that the Align distance works pretty well on images while the Frobenius distance also does so on kinematic data. Overall, SVM classification provides an improvement over 1-NN.

Regarding the evaluation of which data type is most discriminative, it can be seen from Table 1 that optical-flow data does worse than pixel intensities or kinematic data, with the only exception being the suturing task with a 1-NN classifier and the LOSO setup, where it does better than the other data types by a fraction of a percentage. In all other situations that use the 1-NN classifier, kinematic data gives the best performance, with the improvement being particularly significant for the needle passing task. When using SVM classifiers, we get very close performance when using either kinematic or pixel intensities data with the exception of the needle passing task, where LDSs based on kinematic data clearly outperform the use of video data. In the knot tying task, pixel intensities do better while in the suturing task there is no clear winner.

6.3. Results of the bag of features approach

In Section 4 we presented the most common variations of the BoF framework. For a complete evaluation one should provide results of every possible combination and for each task and test setup. Such an approach, however, would require a particularly long list of figures with the risk of hiding important conclusions among complicated discussions. Instead, we prefer to present the results where at each step we evaluate some aspects of the framework while fixing the others. Hence, we adopt the most promising configurations found in the previous step and we test other aspects. Initially we will focus on one task and one test setup, once we have identified the two most promising configurations, we will present results for all tasks and all setups.

In all experiments we use the SVM classifier. As in the experiments with the LDS approach, the penalty parameter C and the γ parameter for the RBF kernel were estimated using 3-fold cross validation over the training set. In order to avoid over-fitting in favor of the most frequent surgemes, we use no more than 1000 randomly sampled features per each class, during the construction of the dictionary, and no more than 40 randomly sampled surgemes per class for the training of the SVM. We repeat the sampling five times.

6.3.1. Effect of the feature type and dictionary size

We start by evaluating the features used and the dictionary size. Specifically, we focus on the suturing task using a LOSO setup, and we build a BoF framework using k-means for clustering, hard thresholding and sum-pooling for building the histograms (that are then normalized), and we use the χ^2 kernel for the SVM classifier. We compare four different descriptors. First we use only the HOG feature, and only the HOF feature. Then we build a hybrid distance where the distances computed in the previous two cases are mixed by a multi-channel approach (20). Finally, we create a new feature where the HOG and HOF descriptors are concatenated. For each of these four cases we test different dictionary sizes: 300, 500, 1000, 2000, and 4000 words. The results of this first step are presented in Table 2.

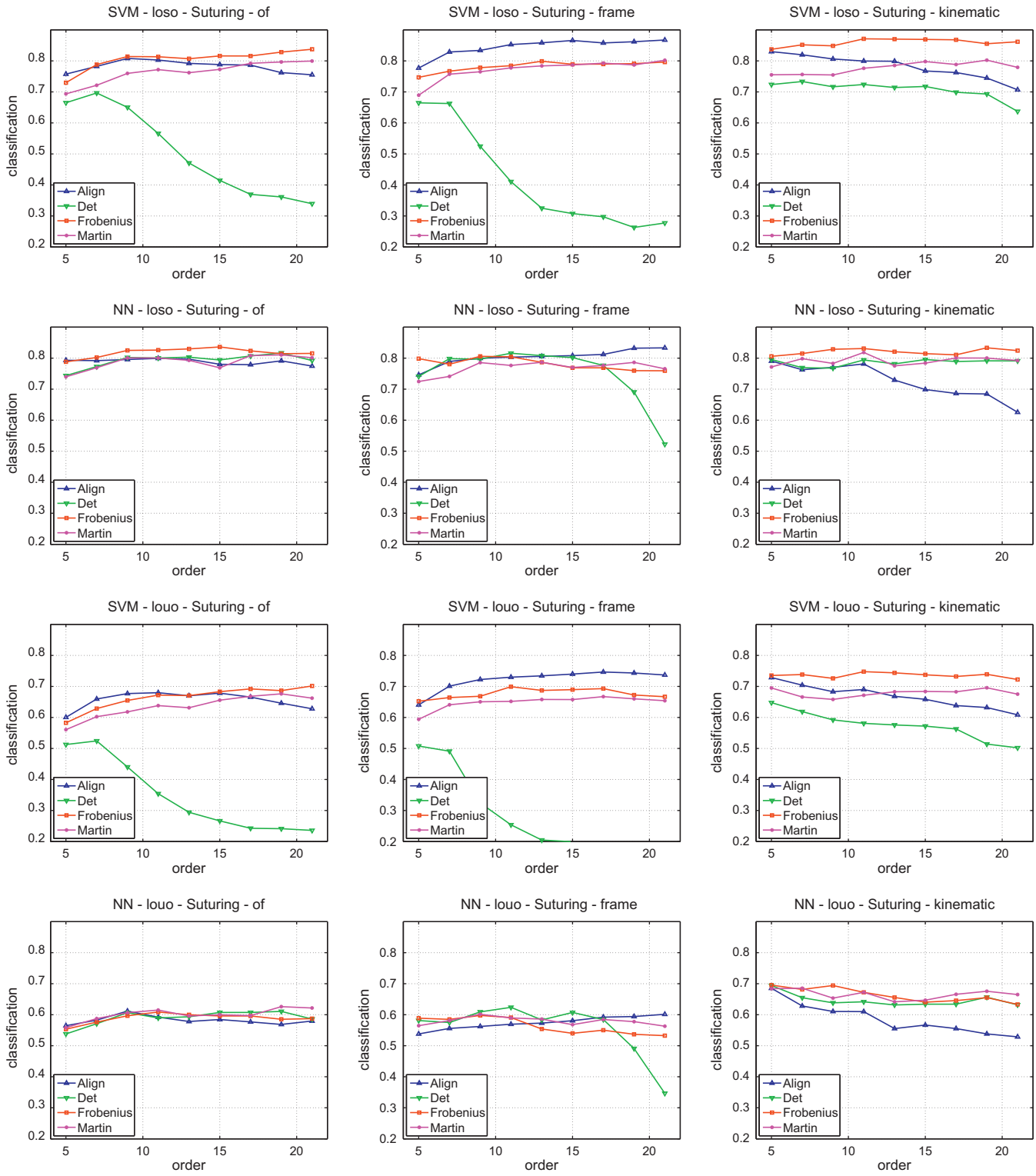

Fig. 3. Effect of the order of the LDS on the average classification rate for the suturing task and for the two setups (LOSO and LOUO). Both SVM and 1-NN performances are shown in the plots for the considered metrics. The first column corresponds to optical-flow data while the second and third correspond, respectively to pixel intensities and kinematic data.

It is interesting to notice that the HOG descriptors tend to be more discriminative than HOF. Note that this ability of HOG to discriminate among different actions is largely due to the fact that HOG are extracted around spatio-temporal interest points, as shown in Fig. 2. In fact, if a HOG descriptor is extracted from the whole frame, then the resulting histogram would be very similar in every action since the background is fairly constant. Instead, STIP points tend to fire around the moving arms and hands of the robot. Hence, the gradient information captures the different poses of the robot while performing different actions, which turns out to be highly discriminative. Qualitatively, this comparison is shown in Fig. 4 where the HOG descriptors extracted around STIP

Table 1
Best average classification rates for the LDS approach using different data types and classification strategies. Corresponding metrics to the displayed results are indicated in brackets and refer to one of the considered metrics, i.e. Align, Frobenius, Martin or Determinant. The best performance for a particular task and setup is highlighted in bold.

	SVM			1-NN		
Task	OF	Img	Kin	OF	Img	Kin
LOSO						
Suturing	85.15 (4.45) [Fro]	87.28 (5.15) [Ali]	**87.45** (4.30) [Fro]	83.61 (8.07) [Fro]	83.31 (7.51) [Ali]	83.33 (7.43) [Fro]
Needle passing	65.22 (4.41) [Mar]	69.39 (2.70) [Ali]	**78.77** (5.90) [Fro]	64.00 (6.64) [Fro]	60.92 (7.15) [Ali]	74.00 (6.67) [Fro]
Knot tying	82.35 (3.54) [Fro]	**87.66** (3.52) [Ali]	85.50 (3.72) [Fro]	80.00 (8.17) [Fro]	77.72 (7.43) [Ali]	85.37 (6.76) [Fro]
LOUO						
Suturing	70.33 (9.68) [Fro]	74.81 (9.93) [Ali]	**75.65** (11.32) [Fro]	62.60 (12.64) [Mar]	62.39 (10.12) [Det]	69.54 (9.87) [Fro]
Needle passing	56.97 (10.59) [Mar]	59.20 (8.97) [Ali]	**69.90** (7.83) [Fro]	50.00 (15.91) [Mar]	45.89 (9.38) [Mar]	64.40 (10.17) [Fro]
Knot tying	73.32 (8.18) [Fro]	**79.36** (10.12) [Ali]	78.89 (8.07) [Fro]	63.78 (11.09) [Det]	65.68 (9.87) [Det]	69.00 (9.51) [Fro]

Table 2
Suturing test, LOSO setup, dictionaries learned by k-means, hard thresholding with sum-pooling, χ^2 kernel. Percentages of correctly identified surgemes for different feature descriptors and dictionary size. Results are averaged across five different random samples for each of the five LOSO test sets. Standard deviation are in parenthesis. The configuration that achieves the highest classification accuracy is highlighted in bold.

BoF with k-means				
# Words	HOG	HOF	HOG + HOF	HOGHOF
300	82.44 (5.35)	76.79 (4.81)	87.62 (5.73)	87.75[a] (5.01)
500	84.41 (5.40)	78.74 (5.26)	88.46 (5.33)	88.89 (5.33)
1000	86.34 (5.24)	81.07 (5.57)	89.23 (5.32)	89.68 (4.83)
2000	87.49 (5.11)	82.06 (5.77)	89.68 (5.43)	90.68 (5.01)
4000	88.36 (5.29)	82.76 (6.48)	89.86 (5.02)	**90.70** (5.08)

[a] This result corresponds to the one reported in Béjar et al. (2012).

points, Fig. 4a, and from the whole frame, Fig. 4b are shown. Specifically, all the HOG descriptors of each video clip have been accumulated and normalized; the surgemes shown are an example of surgemes 2, 3 and 6 (columns 1, 2 and 3, respectively) of one suturing task trial. It is possible to note a considerable difference among the three histograms in Fig. 4a, while the histograms in Fig. 4b look much more alike. Although, this is just a qualitative example it should help to comprehend the importance of extracting features from meaningful patches.

The difference between multi-channel and concatenation is very small. The combination of the HOG and HOF descriptors, in both cases, always improves over the use of any of the descriptors alone. These results are not surprising since it is expected that gradients and motion capture different kind of information. For the next experiments we use the concatenation since it requires computation of a single dictionary rather than one for each channel.

As far as the dictionary size is concerned, with any kind of feature it is possible to observe an improvement of the classification rates when the size is increased. Even if we did not find a point, where performance started to degrade, we stopped at the size of 4000 words due to efficiency reasons. We found that a dictionary size of 2000 words yields a good trade-off between accuracy and computational time requirement (note that by doubling the size to 4000 words very minor improvement is achieved).

6.3.2. Encoding

We now evaluate the encoding step of the BoF framework. We use the concatenated HOGHOF feature, k-means for clustering and a dictionary size of 2000 words. Meanwhile, we try different combinations of thresholding (hard, soft, and hybrid, as shown in (15)–(17) respectively, and pooling (sum and max, as described in (18) and (19). Mean and standard deviation for these tests are shown in Table 3. Overall, hard thresholding with sum-pooling provides the best classification rate (90.68%). Soft thresholding seems to perform better when associated with max rather than sum-pooling. Hybrid thresholding is comparable to hard thresholding both in terms of performance and pooling (i.e., sum-pooling tends to be better than max-pooling). Note that when using the soft or hybrid thresholding an appropriate tuning of the β parameter is crucial. In our experiments we have only tried three different values and the results seem to indicate a preference for higher β values especially when max-pooling is used. Finally, it not surprising that the results of soft thresholding and hybrid thresholding with max-pooling coincide since these two specific combinations lead to the exact same histogram. The performance of the three different encoding techniques is very similar. The only clear conclusion that one may draw is that hybrid is more robust than soft-thresholding with respect to the choice of β. The difference between hybrid and hard-thresholding are minor, hence, it is not possible to state that one encoding is better than the other. We choose to continue our experiments with hard-thresholding since it is faster to compute and it does not require any parameter tuning.

6.3.3. Effect of the sparsity weight for dictionary learning method

Another choice in the BoF framework is the method for constructing the dictionary of visual words. So far we have been using k-means. However, as explained in Section 4.2, one can also use the SDL approach. In our specific implementation we have used the functions mexTrainDL and mexLasso of the SPAMS toolbox[2] for the dictionary learning and coefficient computation phases respectively. As shown in (13), the problem involves the choice of the parameter λ, which weights the sparsity term. Hence, we performed a set of experiments with different λ values, from 0.1 to 0.5, while keeping all the other choices frozen as follows: HOGHOF feature, 2000 words, hard thresholding, sum-pooling, SVM with χ^2 kernel. The dictionary learning algorithm proved to be quite stable with respect to the choice of λ. The percentage of correctly classified surgemes remains very stable, going from 89.93% with $\lambda = 0.1$ to 90.92% with $\lambda = 0.5$ (with a standard deviation always around 5%). For the remaining set of experiments that involved SDL we report results with $\lambda = 0.5$.

6.3.4. Effect of the kernels

The last option left to evaluate is which type of kernel should be used for the SVM classifier. Up until this point we have always used the χ^2 kernel. However, there is a long list of possible kernels. We compare the results of the χ^2 kernel with a linear kernel, and with

[2] Code available at http://spams-devel.gforge.inria.fr/.

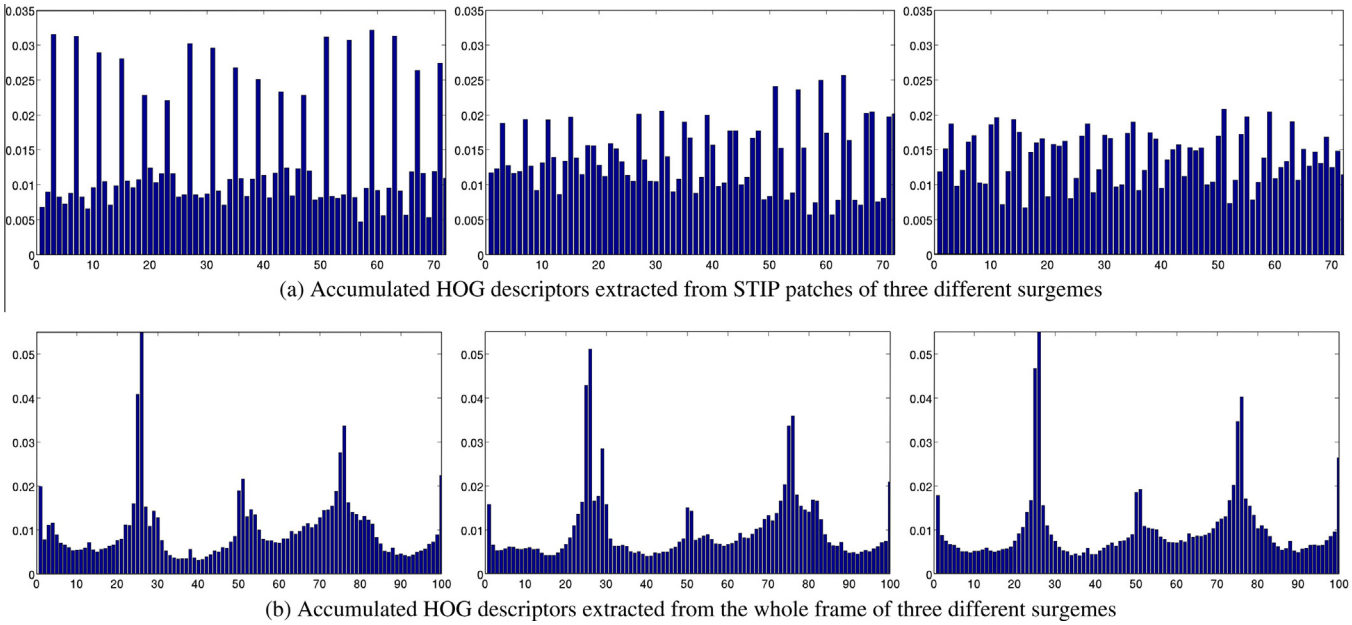

Fig. 4. Comparison between HOG descriptors extracted from patches around STIP and from the whole frame of the same video clips. The surgemes shown are an example of surgemes 2, 3 and 6 (columns 1, 2 and 3, respectively) of one suturing task trial.

Table 3
Suturing test, LOSO setup, HOGHOF features, 2000 words learned by k-means, χ^2 kernel. Percentages of correctly identified surgemes for different thresholding (hard, soft, and hybrid) and pooling (sum and max) techniques. Results are averaged across five different random samples for each of the five LOSO test sets. Standard deviations are in parenthesis. Bold numbers indicate the best performance over task and setup.

BoF with k-means						
Hard	Soft			Hybrid		
	$\beta = 0.9$	$\beta = 0.5$	$\beta = 0.1$	$\beta = 0.9$	$\beta = 0.5$	$\beta = 0.1$
Sum						
90.68 (5.01)	86.61 (4.98)	79.96 (4.82)	39.04 (4.81)	89.33 (5.22)	89.03 (5.12)	89.94 (5.12)
Max						
87.97 (5.62)	88.49 (4.12)	85.81 (4.71)	64.73 (4.11)	88.49 (4.12)	85.81 (4.71)	64.73 (4.11)

Table 4
Percentages of correctly identified surgemes for different tasks (suturing, needle passing, and knot tying) and different test setup (LOSO and LOUO). Results are averaged across five different random samples for each of the five (LOSO) and eight (LOUO) test sets. Standard deviation are in parenthesis. Bold numbers indicate the best performance for each task and setup.

BOF			
k-Means	Suturing	Needle passing	Knot tying
LOSO	90.68 (5.01)	74.14 (4.15)	88.39 (2.38)
LOUO	**79.95** (10.69)	65.53 (10.25)	84.92 (6.40)
SDL	Suturing	Needle passing	Knot tying
LOSO	**91.13** (4.79)	**75.37** (5.30)	**90.66** (2.33)
LOUO	79.32 (8.89)	**66.98** (9.48)	**85.06** (6.05)

another popular choice when dealing with histograms: the intersection kernel.

As expected the linear kernel does not provide very good classification results yielding correct classification rates around 52% for both k-means and SDL. On the other hand, both the intersection and the χ^2 kernels lead to rates of about 91% of correctly classified surgemes independently from the dictionary learning algorithm. In terms of kernels both χ^2 and intersection are reasonable choices, we decide to use the χ^2 although it is possible to expect very similar results when using the intersection kernel.

6.3.5. Effect of different tasks and setups

We now discuss the complete results for different tasks (suturing, needle passing, and knot tying) and different test setups (LOSO, and LOUO). Table 4 compares the results of the BoF framework with k-means and SDL, hard thresholding, sum-pooling and χ^2 kernel, with the results of the BoF framework with SDL, hybrid thresholding, sum-pooling and χ^2 kernel. In practice, both BoF frameworks perform equally well, with the one based on sparsity being marginally better.

Particularly interesting is the LOUO test, which provides an insight into the ability of the algorithms to generalize and recognize gestures performed by users that were unseen during the training phase. Overall, when switching from the LOSO to the LOUO setup, we observe a decrease in performance of around 10% points for the first two tasks, while the results for knot tying task degrade only of about 4% points. Such a difference among tasks suggests that knot tying is possibly performed in a more similar way across users, while for suturing and needle passing the difference in the style might be more accentuated.

Since the clustering and the encoding step when using SDL is computationally more costly than when k-means is used, for the MKL tests we will use the setup with k-means, hard thresholding, sum-pooling and χ^2 kernel.

6.4. Results of combining LDS with BoF using MKL

In this section we provide the experimental results of combining the two considered approaches for surgeme classification, namely LDS and BoF. For the BoF approach we use the χ^2 kernel

Table 5
Average classification rates for different approaches and for the combination of BoF and LDS with MKL. The configuration with the best classification performance is highlighted in bold.

	Kinematic		Video			Hybrid	
Task	SHMM	LDS	BoF	LDS	BoF + LDS	BoF + LDS (kin)	BoF + LDS (all)
LOSO							
Suturing	79.37	87.25	90.68	87.15	91.79	93.52	**93.95**
Needle passing	76.43	78.77	74.14	68.91	77.84	85.32	**86.04**
Knot tying	86.78	85.07	88.39	87.25	90.76	**93.75**	92.76
LOUO							
Suturing	60.85	74.63	79.95	74.22	81.17	86.28	**86.56**
Needle passing	45.26	67.28	65.53	58.77	66.88	80.08	**80.16**
Knot tying	71.94	78.89	84.92	77.36	86.70	90.08	**90.38**

with a dictionary size of 2000 words (built with k-means) and hard thresholding with sum-pooling, while for the LDS we use an RBF kernel. In this case, we did not perform cross-validation over the γ parameter of the RBF kernels. Instead, we feed the MKL algorithm of (Varma and Babu, 2009) with many kernels by using different values of γ and let the algorithm find the weight on each kernel. Effectively, we are letting MKL choose the kernels that give the best performance. For the LDS approach over video data we only consider raw pixel intensities here since, in the case of SVM, they always perform better than optical-flow data. We also leave the Determinant metric out here for the same reasons and consider only the Frobenius, Martin and Align distances. The order of the dynamical models was set to an average value of $n = 15$ since the classification results when using LDSs did not vary much around that order. We tried several experiments varying the value of the SVM penalty parameter C and the regularizer on the kernel weights (ℓ_1- or ℓ_2-norm). It turned out that, in our experiments, the ℓ_1-norm regularizer on the kernel weights always performed better than the ℓ_2-norm regularizer. One possible explanation for this behavior is that when using many kernels, the ℓ_1-norm regularization promotes the use of a sparse set of kernels, which ultimately, leads to the use of only the best kernels. Instead, in the ℓ_2-norm case, even bad kernels might end up with a significant weight, thus resulting in an overall decrease of the kernel quality.

The results for the MKL approach for different combinations of the BoF and the LDS techniques are given in Table 5 where we have also included the (kinematic-based) sparse-HMM approach of Tao et al. (2012) assuming known boundaries. From Table 5 it can be seen that the BoF approach over the video data outperforms the state-of-the-art methods based on kinematic data. Furthermore, the combination of BoF with LDS always produces an improvement over BoF only. In fact, the combination of BoF with LDS over kinematic data always outperforms that of BoF and LDS over images. When merging BoF with LDS over both video and kinematic data (BoF + LDS (all) in Table 5) we still observe an improvement but it is usually small compared to the BoF with LDS (kin). The only exception is Knot Tying in the LOSO setup where there is a small decrease when adding the LDS over the images. The results for LDS with kinematic and pixel intensities alone have also been included to better illustrate the gain achieved when combining them with BoF using multiple kernel learning. It is worth to remark that, even the LDS approach alone can outperform the sparse-HMM method.

From these results we can conclude that, contrary to prior belief, video-based methods can be equally, if not more, discriminative than kinematic-based techniques. It is also important to mention that the combination of heterogeneous data and approaches using MKL generally outperforms the individual approaches alone. This suggests that, as we previously argued, kinematic and video data can be seen as complementary, that is why their combination outperforms any of the other techniques that uses only one type of data.

Even if the presented algorithms are meant to be executed off-line in a batch fashion, and therefore were not written with real-time applications in mind, we would like to give a sense of the computational power required.

For the LDS part, we first need to perform system identification for each surgical gesture. For kinematic data, system identification for each surgeme only takes a fraction of a second since the signals are of low-dimension (78 variables) and of a duration of a few hundreds of frames, while for videos it can take a few seconds for each system identification due to the high dimensionality of the video signal ($320 \times 240 = 76,800$ pixels). The computation of the distances between each surgeme in the test and all training samples, and of the SVM classification only takes a fraction of a second. Therefore, training requires some hours due to the many system identification required and to the computation of all pairwise distances. However, classifying a task composed of around 20 surgemes takes a total of a few seconds for kinematic data and a few minutes with video data.

As far as the BoF algorithm is concerned, once the model is trained (training requires around 1 h), testing on a video of about 2 min length (at 30 frame per seconds) requires only few seconds. However, this is not accounting for the time required for features extraction. Features extraction is actually the most costly operation since it would take for a 2 min video around 10 min of computation. This being said, this operation could easily be implemented in hardware. Hence, we believe that time for testing is not a barrier to on-line applications, in such respect it is the model that should be re-thought with the specific aim of real-time applications in mind.

7. Discussion and conclusion

We have proposed three methods for surgical gesture classification from video data: linear dynamical systems, bag of features and a combination of these frameworks by multiple kernel learning. Results showed that, if the visual features are selected from meaningful locations, video data can be as, or even more, discriminative, than kinematic data. It is fair to say that at this stage our approaches could not be applied directly to real surgeries, where issues such as smoke and blood, could appear. Nevertheless, our aim is to provide a system to assess trainees, who develop their skills from bench setups like the ones we have used in our experiments.

This paper also represents a step toward the recognition of surgical gestures in video. This is because we have used fairly low-level visual features, such as image intensities, image gradients and optical flow. Future work includes using more advanced visual features, such as detection and tracking of surgical tools, and interactions among surgical tools. Another road for future

advances will involve the joint estimation of gestures as well as their temporal segmentation.

Finally, we also proposed a framework to integrate kinematic and video data, and the gain observed in the performance showed that the information carried by these two kinds of data can be considered, at least in part, complementary.

We would also like to stress the benefit of the proposed methodology and its potential applications. There are a number of ways in which gesture recognition can be used. As we have discussed in the article the ability to detect skill deficits at gesture level, and to provide appropriate feedback, is likely to have far greater impact than simple global measures such as time to completion or total motion. In addition, being able to compare gestures will allow us to better diagnose the type of deficit, and to present feedback specific to that deficit.

Furthermore, the ability to recognize gestures sets the stage for intelligent contextual assistance. Padoy and Hager (2011) demonstrated the idea of using gesture recognition as a way of accomplishing shared or cooperative control that is triggered based on what the user was doing. More broadly, this type of gesture recognition could be used to trigger contextually appropriate information displays. For example, if one gesture is recognized the system can predict what the surgeon is going to do after and what he may need, therefore, by using information displays a nurse could be warned to prepare the tools for the following step in an automatic fashion.

A future technical challenge will be to build generalizable models that transfer well across users, and across tasks, so that we really achieve a reasonably universal "language of surgery" that has broad applicability.

Acknowledgements

All the authors were supported in part by NSF Grants 0931805 and 0941362. Dr. Béjar was also supported in part by the Talentia Fellowships Programme of the Andalusian Regional Ministry of Economy, Innovation and Science. Prof. Vidal was supported in part by the European Research Council grant VideoWorld. The authors thank Intuitive Surgical and Carol Reiley for providing the dataset, Nicolas Padoy for discussions about the use of dynamical models for surgical gesture recognition, and Lingling Tao for providing the results of the sparse HMM algorithm with our data setup.

References

Abbou, C.C., Hoznek, A., Salomon, L., Olsson, L.E., Lobontiu, A., Saint, F., Cicco, A., Antiphon, P., Chopin, D., 2001. Laparoscopic radical prostatectomy with a remote controlled robot. The Journal of Urology 165, 1964–1966.

Afsari, B., Chaudhry, R., Ravichandran, A., Vidal, R., 2012. Group action induced distances for averaging and clustering linear dynamical systems with applications to the analysis of dynamic visual scenes. In: IEEE Conference on Computer Vision and Pattern Recognition.

Béjar, B., Zappella, L., Vidal, R., 2012. Surgical gesture classification from video data. In: Medical Image Computing and Computer-Assisted Intervention, pp. 34–41.

Blum, T., Feussner, H., Navab, N., 2010. Modeling and segmentation of surgical workflow from laparoscopic video. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 400–407.

Blum, T., Padoy, N., Feussner, H., Navab, N., 2008. Workflow mining for visualization and analysis of surgeries. International Journal of Computer Assisted Radiology and Surgery 3, 379–386.

Chan, A., Vasconcelos, N., 2005. Probabilistic kernels for the classification of auto-regressive visual processes, in: IEEE Conference on Computer Vision and, Pattern Recognition, pp. 846–851.

Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A., 2011. The devil is in the details: an evaluation of recent feature encoding methods. In: British Machine Vision Conference.

Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R., 2009. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: IEEE Conference on Computer Vision and Pattern Recognition.

Chaudhry, R., Vidal, R., 2009. Recognition of Visual Dynamical Processes: Theory, Kernels and Experimental Evaluation. Technical Report 09-01. Department of Computer Science, Johns Hopkins University.

Cock, K.D., Moor, B.D., 2002. Subspace angles and distances between ARMA models. System and Control Letters 46, 265–270.

Csurka, G., Dance, C., Willamowski, J., Fan, L., Bray, C., 2004. Visual categorization with bags of keypoints. In: European Conference on Computer Vision.

Datta, V., Mackay, S., Mandalia, M., Darzi, A., 2001. The use of electromagnetic motion tracking analysis to objectively measure open surgical skill in laboratory-based model. Journal of the American College of Surgery 193, 479–485.

Doretto, G., Chiuso, A., Wu, Y., Soatto, S., 2003. Dynamic textures. International Journal of Computer Vision 51, 91–109.

Dosis, A., Bello, F., Gillies, D., Undre, S., Aggarwal, R., Darzi, A., 2005. Laparoscopic task recognition using hidden Markov models. Studies in Health Technology and Informatics 111, 115–122.

Gönen, M., Alpaydin, E., 2011. Multiple kernel learning algorithms. Journal of Machine Learning Research 12, 2211–2268.

Judkins, T., Oleynikov, D., Stergiou, N., 2008. Objective evaluation of expert and novice performance during robotic surgical training tasks. Surgical Endoscopy, 21.

Klank, U., Padoy, N., Feussner, H., Navab, N., 2008. Automatic feature generation in endoscopic images. IJCARS 3, 331–339.

Lalys, F., Bouget, D., Riffaud, L., Jannin, P., 2013. Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures. International Journal on Computer Assisted Radiology and Surgery 8, 39–49.

Lalys, F., Riffaud, L., Bouget, D., Jannin, P., 2011. An application-dependent framework for the recognition of high-level surgical tasks in the OR. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 331–338.

Laptev, I., 2005. On space-time interest points. International Journal of Computer Vision 64 (2–3), 107–123.

Lenihan, J., Kovanda, C., Seshadri-Kreaden, U., 2008. What is the learning curve for robotic assisted gynecologic surgery? Journal of Minimally Invasive Gynecology 15, 589–594.

Leong, J., Nicolaou, M., Atallah, L., Mylonas, G., Darzi, A., Yang, G., 2006. HMM assessment of quality of movement trajectory in laparoscopic surgery. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 752–759.

Lin, H., 2010. Structure in Surgical Motion. Ph.D. Thesis. Johns Hopkins University.

Lin, H.C., Shafran, I., Yuh, D., Hager, G.D., 2006. Towards Automatic Skill Evaluation: Detection and Segmentation of Robot-Assisted Surgical Motions. Computer Aided Surgery.

Lowe, D.G., 1999. Object recognition from local scale-invariant features. In: IEEE Conference on Computer Vision and, Pattern Recognition, pp. 1150–1157.

Lowrance, W.T., Elkin, E.B., Jacks, L.M., Yee, D.S., Jang, T.L., Laudone, V.P., Guillonneau, B.D., Scardino, P.T., Eastham, J.A., 2010. Comparative effectiveness of prostate cancer surgical treatments: a population based analysis of postoperative outcomes. The Journal of Urology 183, 1366–1372.

Martin, A., 2000. A metric for ARMA processes. IEEE Transactions on Signal Processing 48, 1164–1170.

McKenzie, C., Ibbotson, J., Cao, C., Lomax, A., 2001. Hierarchical decomposition of laparoscopic surgery: a human factors approach to investigating the operating room environment. Journal of Minimally Invasive Therapy and Allied Technologies 10 (3), 121–127.

Menon, M., Tewari, A., 2003. Robotic radical prostatectomy and the Vattikuti Urology Institute technique: an interim analysis of results and technical points. Urology 61, 15–20.

Miyawaki, F., Masamune, K., Suzuki, S., Yoshimitsu, K., Vain, J., 2005. Scrub nurse robot system – intraoperative motion analysis of a scrub nurse and timed-automata-based model for surgery. Transactions on Industrial Electronics 52, 1227–1235.

Padoy, N., Blum, T., Ahmadi, S., Feussner, H., Berger, M., Navab, N., 2012. Statistical modeling and recognition of surgical workflow. Medical Image Analysis 16, 632–641.

Padoy, N., Blum, T., Essa, I., Feussner, H., Berger, M., Navab, N., 2007. A boosted segmentation method for surgical workflow analysis. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 102–109.

Padoy, N., Hager, G.D., 2011. Human-machine collaborative surgery using learned models. In: IEEE Conference on Robotics and Automation, pp. 5285–5292.

Reiley, C.E., Hager, G.D., 2009. Task versus subtask surgical skill evaluation of robotic minimally invasive surgery. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 435–442.

Reiley, C.E., Lin, H.C., Varadarajan, B., Vagolgyi, B., Khudanpur, S., Yuh, D.D., Hager, G.D., 2008. Automatic recognition of surgical motions using statistical modeling for capturing variability. In: Medicine Meets, Virtual Reality, pp. 396–401.

Richards, C., Rosen, J., Hannaford, B., Pellegrini, C., Sinanan, M., 2000. Skills evaluation in minimally invasive surgery using force/torque signatures. Surgical Endoscopy 14, 791–798.

Rosen, J., Solazzo, M., Hannaford, B., Sinanan, M., 2002. Task decomposition of laparoscopic surgery for objective evaluation of surgical residents' learning curve using hidden Markov model. Computer Aided Surgery 7, 49–61.

Schölkopf, B., Smola, A., 2002. Learning with Kernels. MIT Press, Cambridge, MA.

Sivic, J., Zisserman, A., 2003. Video google: A text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision, pp. 1470–1477.

Tao, L., Elhamifar, E., Khudanpur, S., Hager, G., Vidal, R., 2012. Sparse hidden markov models for surgical gesture classification and skill evaluation. In: Information Processing in Computed Assisted Interventions.

Tibshirani, R., 1994. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B 58, 267–288.

Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.

Varadarajan, B., 2011. Learning and Inference Algorithms for Dynamical System Models of Dextrous Motion. Ph.D. Thesis. Johns Hopkins University.

Varadarajan, B., Reiley, C., Lin, H., Khudanpur, S., Hager, G., 2009. Data-derived models for segmentation with application to surgical assessment and training. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 426–434.

Varma, M., Babu, R., 2009. More generality in efficient multiple kernel learning. In: International Conference on Machine Learning, pp. 1065–1072.

Vishwanathan, S., Smola, A., Vidal, R., 2007. Binet–Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. International Journal of Computer Vision 73, 95–119.

Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C., 2009. Evaluation of local spatio-temporal features for action recognition. In: British Machine Vision Conference, pp. 1–11.

Willems, G., Tuytelaars, T., Gool, L.J.V., 2008. An efficient dense and scale-invariant spatio-temporal interest point detector. In: European Conference on Computer Vision.

Yamauchi, Y., Yamashita, J., Morikawa, O., Hashimoto, R., Mochimaru, M., Fukui, Y., Uno, H., Yokoyama, K., 2002. Surgical skill evaluation by force data for endoscopic sinus surgery training system. In: Int. Conference on Medical Image Computing and Computer Assisted Intervention, pp. 44–51.

Yang, J., Yu, K., Gong, Y., Huang, T., 2009. Linear spatial pyramid matching using sparse coding for image classification. In: IEEE Conference on Computer Vision and, Pattern Recognition, pp. 1794–1801.

Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C., 2007. Local features and kernels for classification of texture and object categories: a comprehensive study. International Journal of Computer Vision 73 (2), 213–238.