Real-Time Task Recognition in Cataract Surgery Videos using Adaptive Spatiotemporal Polynomials

Gwenolé Quellec Mathieu Lamard Béatrice Cochener Guy Cazuguel

IEEE Trans Med Imaging — in press

Abstract

This paper introduces a new algorithm for recognizing surgical tasks in real-time in a video stream. The goal is to communicate information to the surgeon in due time during a video-monitored surgery. The proposed algorithm is applied to cataract surgery, which is the most common eye surgery. To compensate for eye motion and zoom level variations, cataract surgery videos are first normalized. Then, the motion content of short video subsequences is characterized with spatiotemporal polynomials: a multiscale motion characterization based on adaptive spatiotemporal polynomials is presented. The proposed solution is particularly suited to characterize deformable moving objects with fuzzy borders, which are typically found in surgical videos. Given a target surgical task, the system is trained to identify which spatiotemporal polynomials are usually extracted from videos when and only when this task is being performed. These key spatiotemporal polynomials are then searched in new videos to recognize the target surgical task. For improved performances, the system jointly adapts the spatiotemporal polynomial basis and identifies the key spatiotemporal polynomials using the multiple-instance learning paradigm. The proposed system runs in real-time and outperforms the previous solution from our group, both for surgical task recognition ($A_z = 0.851$ on average, as opposed to $A_z = 0.794$ previously) and for the joint segmentation and recognition of surgical tasks ($A_z = 0.856$ on average, as opposed to $A_z = 0.832$ previously).

Index terms — cataract surgery, real-time task recognition, spatiotemporal polynomials, multiple-instance learning

1 Introduction

In anterior eye segment surgeries, the surgeon wears a binocular microscope and the output of the microscope is video-recorded. Real-time analysis of these videos may be useful to automatically communicate information to the surgeon in due time. Typically, relevant information about the patient, the surgical tools or the implant may be communicated to the surgeon whenever he or
she begins a new surgical task. Such a system would be particularly useful to the less experienced surgeons: recommendations on how to best perform the current or the next task, given the patient’s specificities, may be communicated to them. To achieve this goal, we must be able to recognize surgical tasks in real-time during the surgery.

In this paper, we focus on cataract surgery, which is the most common eye surgery [1]. An algorithm has been proposed for the automatic segmentation of cataract surgery videos into surgical tasks [2]. Temporal segmentation is based either on Dynamic Time Warping or on a Hidden Markov Model [3]: the visual content of images is described by visual features extracted within the pupil area. However, that algorithm does not allow real-time recognition of the surgical tasks: temporal segmentation can only be performed when the surgical video is available in full, i.e. after the end of the surgery. In previous works, we have already presented a solution for the automated recognition of surgical tasks in manually-delimited cataract surgery videos [4]. The feature vectors used in that previous solution were unchanged by variations in duration and temporal structure among the target surgical tasks. Therefore, it was possible to retrieve similar video segments and categorize surgical tasks, in real-time, using simple and fast distance measures. Recently, we have presented a solution for the joint segmentation and recognition of surgical tasks in full cataract surgery videos [5], also in real-time. Segmentation relies on the detection of 'idle phases', during which little motion is detected in the surgical scene, which may indicate that the surgeon is changing tools. 'Action phases', which are delimited by two idle phases, are categorized using a Conditional Random Field (CRF) [6] whenever their end is detected. Both segmentation and categorization rely on the video analysis framework proposed in [4]. Although quite fast, that video analysis framework relied on a very simple motion description, which may limit the task recognition ability, hence the need for a new framework. Regarding other surgeries, an overview of existing methods for the automatic recognition and temporal segmentation of tasks or gestures can be found in our previous paper [4]; research in laparoscopic surgery is particularly active [7, 8, 9]. Note that none of these methods was designed to run in real-time.

In automatic video analysis systems, the visual content of a video is usually characterized by feature vectors that represent the shape, the texture, the color and, more importantly, the motion content of the video at different time instants [10, 11]. Motion feature extraction usually involves motion segmentation [12, 13] or salient point characterization [14, 15]. If moving objects have fuzzy borders or are deformable, like in a surgical video, then segmenting the motion content is challenging. Detecting salient point, on the other hand, is possible, but useful information does not necessarily lie where the salient points have been detected. A different solution is proposed in this paper: the motion content of short video subsequences is characterized globally, using a deformable motion model; a polynomial model was adopted. A few authors proposed the use of spatial polynomials [16, 17] or spatiotemporal polynomials [18, 19] for motion analysis. However, the order of the spatiotemporal polynomials was limited to 2 [18] or 3 [19]; a generalization to arbitrary spatiotemporal polynomial orders is proposed in this paper. More importantly, we propose to adapt the polynomial basis to each detection problem.

Recognizing surgical tasks is challenging: a task may be composed of multiple gestures, some of which are specific to this task, some of which are not. To achieve high task recognition performance, feature extraction and classification need to be trained. Given a target surgical task, we need to provide the classifier with examples of videos where this task is visible. Ideally, we would ask an expert to temporally segment key surgical gestures in these videos, in order to help the classifier identify what it should look for in videos. However, this solution would require too much annotation work. Therefore, the classifier should be able to recognize these key gestures itself: for
supervision, we simply indicate which videos contain the target surgical task (and therefore the key surgical gestures). Once this is done, detecting those key gestures in new videos is straightforward. Duchenne et al. [20] proposed a solution based on support-vector machines and the bag-of-visual-words (BoW) model [21, 22, 23]. A solution based on Multiple-Instance Learning (MIL) [24], a variation on supervised learning, is presented in this paper. In MIL, the supervision labels are not assigned to instances (surgical gestures in our case) but to bags containing multiple instances (surgical tasks as a whole in our case): a bag is labeled positive if and only if at least one of its instances is positive [25, 26, 27, 28]. The proposed system is trained to recognize which spatiotemporal polynomials are usually extracted in training videos containing the target surgical task, but not in training videos that do not. These key spatiotemporal polynomials can be used to detect key surgical gestures in new videos and therefore to categorize surgical tasks. Our previous solution for the automated recognition of surgical tasks was also based on the MIL principle [4]. However, motion was analyzed separately along the spatial dimensions and along the temporal dimension. Besides, the proposed characterizations could not be adapted to each surgical task in order to improve performance. In this paper, the system jointly adapts the spatiotemporal polynomial basis and identifies the key spatiotemporal polynomials, for improved performances.

2 Overview of the Method

Let $V^{(i)}$ denote a video containing one surgical task. To categorize the surgical task in a real-time in $V^{(i)}$, short video subsequences are analyzed within $V^{(i)}$. For each video subsequence, a feature vector characterizing motion throughout this spatiotemporal volume is defined. To allow meaningful feature vector comparisons, each frame in $V^{(i)}$ needs to be registered to a coordinate system attached to the patient. The main goal is to remove irrelevant motion information, such as camera motion and patient motion: only the relative surgical tool-patient motions will be extracted. A secondary goal is to remove size variations, due to zoom level variations or camera-patient distance variations. This step is surgery specific: the particular case of anterior eye surgeries, including cataract surgery, is presented in section 3.

Once video subsequences are normalized, their motion content is extracted as described in section 4. Then, the extracted motion content is characterized using a spatiotemporal polynomial model. Two solutions are presented. The first solution relies on the canonical basis of spatiotemporal polynomials (§5). The second solution relies on an adaptive basis of spatiotemporal polynomials (§6).

To train the system, a collection $D$ of surgical task videos is needed. Given a target surgical task (e.g. incision), each video $V^{(i)} \in D$ in which the target task appears is referred to as a relevant video and is noted $V^{(i,+)}$. All other videos are referred to as irrelevant videos and are noted $V^{(i,-)}$. This dataset is divided into a training subset $D_{\text{train}}$ and a test subset $D_{\text{test}}$.

Once video subsequences are characterized, relevant subsequences are detected in $V^{(i)}$. In order to detect relevant video subsequences, the idea is to identify spatiotemporal polynomials that tend to appear in relevant training videos ($V^{(i,+)} \in D_{\text{train}}$) but not in irrelevant ones ($V^{(i,-)} \in D_{\text{train}}$). These spatiotemporal polynomials are called key spatiotemporal polynomials. They are learnt using the Diverse Density (DD) criterion [25], a well-known MIL algorithm (§7). If an adaptive basis of spatiotemporal polynomials is used, a variation on DD is used jointly for basis adaptation and key spatiotemporal polynomial learning.

Finally, the detected relevant subsequences are used to estimate the binary label (+ or -) for video $V^{(i)}$ as a whole, using one DD model trained per target surgical task (§8). There are essentially
two novelties in this paper: the design of an adaptive basis of spatiotemporal polynomials for motion characterization (§6) and the generalization of DD for basis adaptation (§7.4). Cataract surgery video normalization was recently presented at a conference [29] and is summarized hereafter.

3 Normalizing Cataract Surgery Videos

In order to normalize motion information extracted from videos, each frame in these videos is registered to a coordinate system attached to the anterior segment of the eye (see Fig. 1). The main anatomical landmarks in these videos are the pupil boundaries and the limbus (the outer iris boundary). However, segmenting them is challenging due 1) to occlusion, 2) to the variety of colors and textures in the pupil, the iris, the sclera and the lids and 3) to the variety of zoom factors. In the proposed solution, the pupil center and a scale factor are tracked in videos without explicitly segmenting the pupil or the iris.

3.1 Pupil Center Tracking

The proposed pupil center detector uses the fact that the pupil boundaries, the limbus and the sclera / lid interface are concentric. Using a Hough transform to detect circle centers in images, a prominent response is observed at the pupil center, which is also the center of the limbus and of the sclera / lid interface. To improve performance, the Hough transform is modified in such a way that edge information is only accumulated on the darkest side of the edge, on which the pupil likely is. Performance is improved further by smoothing the accumulator both spatially and temporally. The distance between the manually-indicated and the estimated pupil centers was equal to 8.0 ± 6.9% of the manually-measured limbus radius [29].

![Figure 1: Video normalization](image-url)
3.2 Scale factor Tracking

The image scale factor is estimated from the illumination pattern reflected on the cornea: the three bright spots inside the pupil (see Fig. 1). It is defined as the height of the triangular pattern. Note that the three spots are not always visible, in particular when the cornea is distorted due to a tool insertion (see Fig. 3 (b), (g), (i)). The solution is to keep track of the estimated scale factor and only update it when the three spots are clearly visible. It should be noted that surgeons don’t change the zoom level while they are performing a surgical gesture: the zoom level only changes when no tools are inserted. The limbus diameter is the best indicator of the zoom level in images: 1) it does not change over time, unlike the pupil diameter, and 2) it is little variant across the population. The correlation between the estimated zoom level and the manually-measured limbus size in images was $R = 0.834$ [29].

3.3 Region of Interest

A squared region of interest (ROI) is defined around the estimated pupil center (see Fig. 1). Its size equals 1.5 times the estimated limbus diameter. The content of this ROI is resized to a fixed-size 513 x 513 image, which is then processed by the motion extraction module ($§4$). If the ROI is not fully contained inside the original image (when the pupil is close to the image border), then part of the new fixed-size image is undefined. This undefined area is ignored by the motion extraction module.

4 Motion Extraction

Once videos are normalized, motion is extracted from the optical flow between consecutive frames.

4.1 Optical Flow

Let $V_k^{(i)}$ and $V_{k+1}^{(i)}$ denote two consecutive frames in video $V^{(i)}$. Dense optical flows seem more suitable than sparse optical flows in surgical applications: while sparse optical flow algorithms rely on the detection of strong corners, dense optical flows can capture soft tissue deformations and track line-shaped tools, with little corners. Farnebäck’s algorithm is used to compute the dense optical flow between $V_k^{(i)}$ and $V_{k+1}^{(i)}$ [30]. This algorithm relies on local approximations of image neighborhoods by polynomial expansions; a multiscale implementation is used [31]. For faster computations, the dense optical flow is not estimated in all pixel positions: displacements are only estimated in $\Delta$ uniformly sampled measurement points.

4.2 Video Subsequences

In order to detect surgical subtasks or gestures, motion information extracted from two consecutive frames may not be enough: the time interval likely is too short. More generally, motion is analyzed inside video subsequences of $n$ frames. These subsequences may overlap: one video subsequence is analyzed every $m$ frames. Parameter $m$ is chosen to tradeoff retrieval precision and computation times ($§8.2$). Let $V_{jm+1:jm+n}^{(i)}$ denote the $j^{th}$ video subsequence in $V^{(i)}$:

$$V_{jm+1:jm+n}^{(i)} = \{ V_{jm+1}^{(i)}, V_{jm+2}^{(i)}, ..., V_{jm+n}^{(i)} \}$$  \hspace{1cm} (1)
All motion vectors extracted from consecutive frames of $V_{jm+1;jm+n}$ are put into a single motion field $\mathcal{F}^{(i,j)}$, or $\mathcal{F}$ for short. Each element $f_d \in \mathcal{F}$ maps a spatiotemporal coordinate $(x_d = x_{k,p}, y_d = y_{k,p}, t_d = k - jm)$, i.e. a time-indexed measurement point, to a displacement $(u_d, v_d)$ provided by Farnebäck’s algorithm, $d = 1..D$, where $D = (n-1)\Delta$.

5 Motion Characterization using Canonical Spatiotemporal Polynomials

In order to characterize the motion field within subsequence $V_{jm+1;jm+n}$, the motion vectors in $\mathcal{F}$ are approximated by two spatiotemporal polynomials. The first polynomial maps the spatiotemporal coordinate $(x, y, t)$ to the horizontal displacement $u$. The second maps the spatiotemporal coordinate to the vertical displacement $v$.

5.1 Spatiotemporal Polynomials of Maximal Order $p$

Let $p$ denote the maximal polynomial order. Given a basis of canonical polynomials, we search for a matrix of polynomial coefficients, noted $P^{(p,i,j)}$ or $P$ for short, that minimize the sum of the squared errors between the true motion field $\mathcal{F}$ and the motion field approximated by a polynomial model of maximal order $p$. Matrix $P$ is referred to as the canonical motion characterization of subsequence $V_{jm+1;jm+n}$.

5.2 Canonical Polynomial Bases

Canonical polynomial bases of maximal order $p$ are noted $C^{(p)}$. They have the following structure:

$C^{(1)} = \{1, x, y, t\}$, $C^{(2)} = \{1, x, y, t, xy, xt, yt, x^2, y^2, t^2\}$, etc. Let $L_p = |C^{(p)}|$ denote the number of canonical polynomials. The number of canonical polynomials of order $k$ is the number of combinations of $k$ elements in $\{x, y, t\}$ with repetition, therefore:

$$L_p = \sum_{k=0}^{p} \binom{2 + k}{k}$$  \hspace{1cm} (2)

5.3 Canonical Motion Characterization

Let $(x_d, y_d, t_d)$, $d = 1..D$, be the spatiotemporal coordinates of the salient points detected in subsequence $V_{jm+1;jm+n}$. Let $C_{d}^{(p)}$, or $C_d$ for short, denote the vector formed by the canonical polynomials in $C^{(p)}$, evaluated at coordinate $(x_d, y_d, t_d)$: for instance, $C_{d}^{(1)} = (1, x_d, y_d, t_d)$. The approximated motion vector at coordinate $(x_d, y_d, t_d)$ can be expressed as a matrix product $C_dP$ (see Fig. 2). Matrix $P$ is defined as follows:

$$P = \arg\min_{X} \sum_{d=1}^{D} \|(u_d, v_d) - C_dX\|^2$$  \hspace{1cm} (3)
Figure 2: Motion field measurement and approximation. The green motion field is the one measured by Farnebäck’s algorithm between the previous and the current frame. The blue motion field is the one obtained through the spatiotemporal polynomial approximation (polynomial order $p=6$, subsequence length: $n=5$ frames). The resized region of interest is shown below each frame. Figure (a) shows that blurry tool motions can be detected. Figure (b) shows that images with unusual zoom levels can be processed correctly. Figure (c) indicates that soft tissue deformation can be detected. Figure (d) shows that fluid displacements can also be detected.

The optimal solution is found when the derivative of the sum, with respect to matrix $X$, equals 0. This solution can be rewritten as follows:

$$
\begin{align*}
AP &= E \\
A &= \sum_{d=1}^{D} C_d^T C_d \\
E &= \sum_{d=1}^{D} C_d^T (u_d, v_d)
\end{align*}
$$

with $A \in \mathcal{M}_{L_p \times L_p}$ and $E \in \mathcal{M}_{L_p \times 2}$. Matrix $P$ is obtained by solving two systems of linear equations of order $L_p$: one for the horizontal displacements and one for the vertical displacements. In the first system, the right-hand side of the equation is the first column of $E$. In the second system, the right-hand side is the second column of $E$. These systems are solved using the LU decomposition of $A$. The solution of the first (respectively the second) system is stored in the first (respectively the second) column of $P$.

5.4 Complexity analysis

The most complex step in the computation of the canonical motion characterization $P$, given the motion field $F$, is the computation of matrix $A$. Since $A$ is symmetric, its computation requires $O(\frac{1}{2}DL_p^3)$ operations. In comparison, the complexity of its LU decomposition is in $O(\frac{1}{2}DL_p^3)$. So, the complexity of the entire process increases linearly with $D$, the number of time-index measurement points ($\S 4$).
6 Motion Characterization using Adaptive Spatiotemporal Polynomials

In this section, the basis polynomials are no longer canonical polynomials, but rather a linear combination of canonical polynomials.

6.1 Adaptive Polynomial Basis

Let \( l \) denote the number of adaptive basis polynomials. Let \( \Pi^{(p,l)} \in M_{L_p,l} \), or \( \Pi \) for short, denote the projection matrix between the canonical polynomial basis of maximal order \( p \) and the adaptive polynomial basis of dimension \( l \). In this new basis, the polynomial coefficients evaluated at coordinate \((x_d, y_d, t_d)\) are stored in a \( l \)-dimensional vector \( \bar{C}^{(p,l)} \), or \( \bar{C} \) for short:

\[
\bar{C}_d = C_d \Pi \tag{5}
\]

Let \( \bar{P}^{(p,l,i,j)} \), or \( \bar{P} \) for short, denote the motion characterization of a video subsequence \( V_{[jm+1;jm+n]}^{(i)} \) in the new basis. \( \bar{P} \) can be expressed as follows:

\[
\begin{align*}
\bar{A} \bar{P} &= \bar{E} \\
\bar{A} &= \sum_{d=1}^{D} (C_d \Pi)^T (C_d \Pi) = \Pi^T A \Pi \\
\bar{E} &= \sum_{d=1}^{D} (C_d \Pi)^T (u_d, v_d) = \Pi^T E
\end{align*} \tag{6}
\]

where \( \bar{A} \in M_{l,l} \) is an adapted version of matrix \( A \) (see equation 4) and \( \bar{E} \in M_{l,2} \) is an adapted version of matrix \( E \).

6.2 Basis Adaptation

For basis adaptation purposes, we need to know how each coefficient of the projection matrix \( (\Pi) \) impacts the motion characterization \( (\bar{P}) \) of the video subsequence. In other words, we need to compute the partial derivatives of \( \bar{P} \) with respect to each coefficient of \( \Pi \). In order to reduce the complexity of basis adaptation, the dimension \( l \) of the adaptive basis should be as low as possible.

6.3 Deriving the Motion Characterization with respect to Projection Coefficients

In order to use standard derivation rules, equation 6 is inverted as follows:

\[
\bar{P} = \bar{A}^{-1} \bar{E} \tag{7}
\]

According to the rule of matrix product derivation, the partial derivative of \( \bar{P} \) with respect to coefficient \( \Pi_{u,v} \) of \( \Pi \) is given below:

\[
\frac{\partial \bar{P}}{\partial \Pi_{u,v}} \triangleq \frac{\partial \bar{A}^{-1}}{\partial \Pi_{u,v}} \bar{E} + \bar{A}^{-1} \frac{\partial \bar{E}}{\partial \Pi_{u,v}} \tag{8}
\]
According to the rule of inverse matrix derivation, the partial derivative of \( \bar{A}^{-1} \) with respect to \( \Pi_{u,v} \) is given below:

\[
\frac{\partial \bar{A}^{-1}}{\partial \Pi_{u,v}} \triangleq -\bar{A}^{-1} \frac{\partial \bar{A}}{\partial \Pi_{u,v}} \bar{A}^{-1}
\]  

(9)

Therefore, equation 8 becomes:

\[
\frac{\partial \bar{P}}{\partial \Pi_{u,v}} = \bar{A}^{-1} \left[ \frac{\partial \bar{E}}{\partial \Pi_{u,v}} - \frac{\partial \bar{A}}{\partial \Pi_{u,v}} \bar{A}^{-1} \bar{E} \right]
\]  

(10)

Matrix \( \frac{\partial \bar{E}}{\partial \Pi_{u,v}} \) has the following format:

\[
\frac{\partial \bar{E}}{\partial \Pi_{u,v}} = \begin{pmatrix}
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 \\
E_{u,1} & E_{u,2} & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0
\end{pmatrix}
\]  

(11)

where the only non-zero row is the \( v^{th} \) row. This is because \( \sum_{r=1}^{L} \Pi_{s,q} E_{s,r} \), whose derivative with respect to \( \Pi_{u,v} \) is zero whenever \( q \neq v \). As for matrix \( \frac{\partial \bar{A}}{\partial \Pi_{u,v}} \), it has the following format:

\[
\frac{\partial \bar{A}}{\partial \Pi_{u,v}} = \begin{pmatrix}
0 & 0 & \cdots & 0 & a_{u,1} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & a_{u,v-1} & 0 & \cdots & 0 \\
a_{u,1} & a_{u,2} & \cdots & a_{u,v-1} & a_{u,v} & a_{u,v+1} & \cdots & a_{u,l} \\
0 & 0 & \cdots & 0 & a_{u,v+1} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & a_{u,1} & 0 & \cdots & 0
\end{pmatrix}
\]  

(12)

where the only non-zero row is the \( v^{th} \) row, the only non-zero column is the \( v^{th} \) column and \( a_{u,q} = \sum_{r=1}^{L} \Pi_{r,q} A_{u,r} \). These formulas take advantage of the symmetry of matrix \( \bar{A} \) (see equation 4).

### 7 Finding the Key Spatiotemporal Polynomials

Once each video subsequence has been characterized, key spatiotemporal polynomials are identified through Multiple-Instance Learning (MIL) in the training subset \( D_{train} \).
7.1 Multiple-Instance Learning

Multiple-instance learners are supervised learners that receive a set of bags of instances. A binary label (relevant or irrelevant) is assigned to each bag [24]. A bag is labeled irrelevant if all the instances in it are irrelevant. On the other hand, a bag is labeled relevant if it contains at least one relevant instance (or one key instance). From a collection of labeled bags, multiple-instance learners are trained to detect relevant instances.

In this paper, the characterization of each video subsequence $V^{i(m)}_{jm+1: jm+n}$ is regarded as an instance. Each video $V^{i(m)}_{jm}$ is regarded as a bag of instances. The relevant instances we are looking for are the key spatiotemporal polynomials.

The most popular MIL frameworks are Andrews’ SVM [26], Diverse Density (DD) [25], and their derivatives. In Andrews’ SVM, a support-vector machine processes the instance labels as unobserved integer variables, subjected to constraints defined by the bag labels. The goal is to maximize the soft-margin over hidden label variables and a discriminant function. DD measures the intersection of the relevant bags minus the union of the irrelevant bags. The location of relevant instances in feature space, and also the best weighting of the features, is found by maximizing DD. DD was chosen for its simplicity and its generality: in particular, it is suitable for spatiotemporal polynomial basis adaptation (§7.4).

7.2 Notations

Irrelevant and relevant videos are noted $V^{i(-m)}$ and $V^{i(+m)}$, respectively (§2). In the canonical model (§5), the characterizations of their subsequences are noted $P^{p,i,j,-}$ and $P^{p,i,j,+}$, respectively, or $P^{i,j,-}$ and $P^{i,j,+}$ for short. In the adaptive model (§6), these characterizations are noted $\bar{P}^{p,i,j,-}$ and $\bar{P}^{p,i,j,+}$, respectively, or $\bar{P}^{i,j,-}$ and $\bar{P}^{i,j,+}$ for short. Let $\bar{P}$ denote the key spatiotemporal polynomial.

7.3 Maron’s Diverse Density

DD, in its simplest form, is defined as follows:

$$\bar{P} = \arg \max_P \left\{ \prod_i P_{r(+|V^{i,+}, P)} \times \prod_i P_{r(-|V^{i,-}, P)} \right\} \quad (13)$$

where $\delta$ stands for the class label (+ or -). For convenience, instead of maximizing the DD criterion, the opposite of its logarithm, noted $f(P)$, is minimized instead:

$$\left\{ \begin{array}{l}
    f(P) = f^+(P) + f^-(P) \\
    f^+(P) = -\sum_i \log \left(1 - e^{V^+_i}\right) \\
    f^-(P) = -\sum_i V^-_i \\
    V^+_i = \sum_j \log \left(1 - e^{-\|P^{i,j,-} - P\|^2}\right)
\end{array} \right. \quad (14)$$

10
The key spatiotemporal polynomial, \( \hat{P} \), is found by gradient descents controlled by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [32]. In this paper, \( n_d \) gradient descents are performed to find a key spatiotemporal polynomial. Each descent is initialized by a randomly selected instance within the relevant bags [27]. The solution maximizing the DD criterion (i.e. minimizing the opposite of its logarithm) is retained.

Considering the canonical model (§5), the gradient of the objective function \( f(P) \) consists of the \( 2L_p \) partial derivatives with respect to the polynomial coefficients. The partial derivative of \( f(P) \) with respect to one polynomial coefficients \( P_{u,v} \) is given by:

\[
\left\{ \begin{array}{ll}
\frac{\partial f(P)}{\partial P_{u,v}} &= \frac{\partial f^+(P)}{\partial P_{u,v}} + \frac{\partial f^-(P)}{\partial P_{u,v}} \\
\frac{\partial f^+(P)}{\partial P_{u,v}} &= \sum_i \frac{1}{e^{V_i}} \frac{\partial V_i}{\partial P_{u,v}} \\
\frac{\partial f^-(P)}{\partial P_{u,v}} &= -\sum_i \frac{1}{1-e^{-V_i}} \frac{\partial V_i}{\partial P_{u,v}} \\
\frac{\partial V_i^\pm}{\partial P_{u,v}} &= 2 \sum_j \frac{e^{V_i^+(i,j,\delta) - P_{u,v}}}{1-e^{-V_i^+(i,j,\delta) - P_{u,v}}} (\hat{P}_{i,j,\delta} - P_{u,v})
\end{array} \right. \tag{15}
\]

Note that features can be weighted simultaneously: the Euclidean distance simply needs to be replaced by a weighted Euclidean distance [25]:

\[
(\hat{P}, \hat{s}) = \arg\max_{P,s} \left\{ \prod_i \Pr(+|V^{(i,+)}), P, s \right\} \times \prod_i \Pr(-|V^{(i,-)}, P, s) \right\} \tag{16}
\]

In this case, the gradient of the objective function \( f(P) \) (see equation 14) consists of the \( 2L_p \) partial derivatives with respect to the polynomial coefficients and the \( 2L_p \) partial derivatives with respect to the weights. The weights are initialized to 1 in each gradient descent.

### 7.4 Joint Basis Adaptation and Key Spatiotemporal Polynomial Learning

In this section, the adaptive model (§6) is considered. The goal is to find the key spatiotemporal polynomial and the best polynomial basis simultaneously:

\[
(\hat{P}, \hat{\Pi}) = \arg\max_{P,\Pi} \left\{ \prod_i \Pr(+|V^{(i,+)}), P, \Pi \right\} \times \prod_i \Pr(-|V^{(i,-)}, P, \Pi) \right\} \tag{17}
\]

Searching for the best polynomial basis alters the distance between instances, so there is no need to search for optimal weights: modifying the polynomial basis is more general. In this case, the gradient of the objective function \( f(P) \) (see equation 14) consists of the \( 2l \) partial derivatives with respect to the polynomial coefficients (obtained similarly to 15) and the \( L_p \times l \) partial derivatives...
with respect to the projection coefficients. The partial derivative of the objective function, with 
respect to one projection coefficient $\Pi_{u,v}$, is given by:

$$
\begin{align*}
\frac{\partial f(P)}{\partial \Pi_{u,v}} &= \frac{\partial f^+(P)}{\partial \Pi_{u,v}} + \frac{\partial f^-(P)}{\partial \Pi_{u,v}} \\
\frac{\partial f^+(P)}{\partial \Pi_{u,v}} &= \sum_i \frac{e^{v_i^+}}{1-e^{v_i^+}} \frac{\partial \bar{V}^+}{\partial \Pi_{u,v}} \\
\frac{\partial f^-(P)}{\partial \Pi_{u,v}} &= -\sum_i \frac{1}{1-e^{v_i^+}} \frac{\partial \bar{V}^-}{\partial \Pi_{u,v}} \\
\frac{\partial \bar{V}^\delta}{\partial \Pi_{u,v}} &= \sum_j \log \left(1 - e^{-\|\hat{P}(i,j,\delta) - P\|^2}\right) \\
\frac{\partial \bar{V}^\delta}{\partial \Pi_{u,v}} &= 2 \sum_j \frac{e^{-\|\hat{P}(i,j,\delta) - P\|^2}}{1-e^{-\|\hat{P}(i,j,\delta) - P\|^2}} \\
&\times c_i \left(\frac{\partial \hat{P}(i,j,\delta)}{\partial \Pi_{u,v}}\right)^T c \left(\hat{P}(i,j,\delta) - P\right)
\end{align*}
$$

where $\delta$ stands for the class label (+ or -) and $c_i$ is an operator that concatenates the columns 
of a matrix. The partial derivatives of the adapted motion characterizations $\hat{P}(i,j,\delta)$ are given in 
equation 10. Note that the main difference with system 15 lies in the last equation.

The optimal basis and the key spatiotemporal polynomial are found by $n_q$ gradient descents 
controlled by the BFGS algorithm (§7.3). For each descent, the polynomial basis is initialized 
though a principal component analysis of the spatiotemporal polynomials in input space ($\mathbb{R}^{T_T}$): 
the spatiotemporal polynomial predicting horizontal displacements and those predicting vertical 
displacements altogether. The first $l$ components are used as the $l$ columns of the initial projection 
matrix.

8 Surgical Task Recognition

Once key spatiotemporal polynomials are characterized, the surgical task performed in video $V^{(i)}$ is 
recognized.

8.1 Multiclass Multiple Instance Learning

For surgery monitoring, we are interested in detecting multiple surgical tasks in videos: $T_t, t = 1, ..., T$. 
For each key surgical task $T_t$, one key spatiotemporal polynomial $\hat{P}_t$ and either one weight 
vector $\tilde{s}_t$ (see equation 16) or one polynomial basis $\tilde{P}_t$ (see equation 17) have been learnt. The 
probability that task $T_t$ occurs in video $V^{(i)}$ is given either by $P_t = Pr(+|V^{(i)}, \hat{P}_t, \tilde{s}_t)$ or by $P_t = 
Pr(+|V^{(i)}, \tilde{P}_t, \tilde{\Pi}_t)$: $P_t$ may be used as the criterion to decide whether or not a task of type $t$ 
occurred in video $V^{(i)}$. But, to push recognition performance further, we should take advantage of 
all $P_u$ probabilities, $u = 1, ..., T$, to recognize task $t$. In that purpose, a classifier was trained in the 
$T$-dimensional space generated by all $P_u$ probabilities, $u = 1, ..., T$, using the $D_{train}$ dataset. 
The two leading classification frameworks nowadays are Random Forests (RFs) [33] and Support-Vector 
Machines (SVMs) [34]. The main advantage of SVMs over RFs is that they are good at avoiding 
over-fitting: they can perform well with less training data. One drawback of SVMs is that their 
performance depends on how the data is normalized, and normalization is particularly challenging 
in case of heterogeneous data. However, in this application, input data are all probabilities, so 
they are homogeneous and do not need to be normalized. An SVM classifier was built using the 
well-known Gaussian kernel. The training procedure for the full system is described in the following 
section.
8.2 Training Surgical Task Recognition

In the weight adaptation solution, key spatiotemporal polynomial finding has five parameters (§2, §5, §7.3): the number of frames per subsequence \(n\), the delay between the beginning of two consecutive subsequences \(m\), the number of selected measurement points per frame \(\Delta\), the maximal polynomial order \(p\) and the number of gradient descents \(n_d\). In the basis adaptation solution, the task recognition system has a sixth parameter (§6): the number of adaptive basis polynomials \(l\). Two parameters were chosen empirically to allow real-time fitting of the polynomial models: \(m=5\) frames and \(\Delta=400\) measurement points per frame. One parameter was chosen empirically to have fast training times: \(n_d=5\). The other two \((n, p)\) or three \((n, p, l)\) were chosen by two-fold cross-validation in the training set. Each tested \((n, p)\) or \((n, p, l)\) tuple was graded by the average, over all surgical tasks and both folds, of the area under the Receiver Operating Characteristic (ROC) of the SVM classifier. In each fold, the parameters of the SVM (§8.1) were trained by a nested two-fold cross-validation. Once the optimal parameters were found, the full system was retrained in the entire training set using the optimal parameters.

8.3 Joint Segmentation and Recognition of Surgical Tasks

For the task of jointly segmenting and recognizing surgical tasks in real time, the framework proposed in [5] is used, after replacing our previous video analysis framework [4] with the proposed framework. Because the framework in [5] relies on analogy reasoning, the SVM classifier was replaced by a k-nearest neighbors classifier in the recognition step.

The segmentation step also relies on analogy reasoning. But in that case, instances are assessed individually [5], so a distance between spatiotemporal polynomials needs to be defined (as opposed to a distance between bags of instances in the recognition step). In that purpose, the presence of surgical tools is manually detected in a few training videos: a set of video portions is obtained. Those containing surgical tools are labeled positive, the others are labeled negative. These video portions are used to train an ‘action detector’, defined as a key spatiotemporal polynomial detector (§7). In the case of \((\breve{P}, \breve{\Pi})\) adaptation (§7.4), the distance between instances is defined as the Euclidean distance in the adapted polynomial space. In the case of \((\breve{P}, \breve{s})\) adaptation, it is defined as the weighted Euclidean distance (see equation 16).

9 Application to a Cataract Surgery Dataset

The proposed system was applied to cataract surgery. Two experiments were conducted. The first one was about surgical task recognition in manually-delimited cataract surgery videos. The second experiment was about the joint segmentation and recognition of surgical tasks in full cataract surgery videos. The dataset used in these experiments is presented hereafter.

9.1 The Cataract Surgery Video Dataset

A dataset of 186 videos from 186 consecutive cataract surgeries was collected at Brest University Hospital (Brest, France) between February and July 2011. Surgeries were performed by 10 different surgeons of various experience levels. Some videos were recorded with a CCD-IRIS device (Sony, Tokyo, Japan), the others were recorded with a MediCap USB200 video recorder (MediCapture,
Philadelphia, USA). They were stored in MPEG2 format, with the highest quality settings, or in DV format. Image definition is 720 x 576 pixels. The frame frequency is 25 frame per seconds.

In each video, a temporal segmentation was provided by cataract experts for each surgical task. The following surgical tasks were temporally segmented in videos: incision, rhexis, hydrodissection, phacoemulsification, epinucleus removal, viscous agent injection, implant setting-up, viscous agent removal and stitching up (see Fig. 3).

9.1.1 Surgical Task Recognition Experiment

in order to compare the proposed framework with our previous surgical task recognition solution [4], a subset of 100 videos was used in the first experiment. Nine manually-delimited clips were obtained per surgery. Overall, 900 clips were obtained. Clips have an average duration of 94 seconds. The dataset was randomly divided into two subsets of 50 surgeries: one was used as training set, the other was used as test set. For comparison purposes, the dataset split defined in [4] was used.
9.1.2 Joint Segmentation and Recognition Experiment

The full dataset was randomly divided into two subsets of 93 surgeries: one was used as training set, the other was used as test set. For comparison purposes, the dataset split defined in [5] was used. Note that the training set (respectively the test set) include the smaller training set (respectively test set) defined above. For this experiment, a miscellaneous task category was created to account for all the optional surgical phases: iris retractor setting-up, iris retractor removal, angle measurement, landmark tracing, etc. In order to train the idle phase detector (§8.3), the presence of surgical tools was manually delimited in a subset of 10 surgery videos from the training set [5].

9.2 Results of the Surgical Task Recognition Experiment

Each parameter was trained separately, using a default value \((n = 10, p = 3, l = 5)\) for the other two parameters. As an illustration, the optimal projection parameters obtained for the ‘incision’ class for these parameters are reported in Fig. 4. Cross-validation results for \((\hat{P}, \hat{\Pi})\) adaptation in the training set are reported in table 1. The optimal tuples of parameters were \((n = 20, p = 4)\) and \((n = 20, p = 4, l = 5)\). For each surgical task, the ROC curve of the SVM classifier in the test set is reported in Fig. 5. The area under these curves \((A_z)\) for the proposed method using either \((\hat{P}, \hat{s})\) adaptation or \((\hat{P}, \hat{\Pi})\) adaptation are reported in table 2. To show the advantage of the proposed multiclass MIL extension (§8.1), the performance of the standard single-class MIL (i.e. the area under the ROC curve of \(Pr(+|V, \hat{P}, \hat{\Pi})\)) is also reported in table 2. The performance of the proposed method was compared to our previous method for cataract surgical task categorization [4] and to our implementation of Duchenne’s method [20] in terms of \(A_z\) (see table 2) and in terms of computation times.
Table 1: Cross-validation results for $(\hat{P}, \hat{\Pi})$ adaptation in the training set

<table>
<thead>
<tr>
<th>varying $n$</th>
<th>$n$</th>
<th>$p$</th>
<th>$l$</th>
<th>$A_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>5</td>
<td>0.806</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>5</td>
<td>0.827</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>5</td>
<td><strong>0.853</strong></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>3</td>
<td>5</td>
<td>0.815</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>5</td>
<td>0.797</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>varying $p$</th>
<th>$n$</th>
<th>$p$</th>
<th>$l$</th>
<th>$A_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>5</td>
<td>0.810</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>5</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>5</td>
<td><strong>0.854</strong></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>5</td>
<td>0.843</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>5</td>
<td>0.824</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>varying $l$</th>
<th>$n$</th>
<th>$p$</th>
<th>$l$</th>
<th>$A_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>2</td>
<td>0.809</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>3</td>
<td>0.814</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>4</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>5</td>
<td><strong>0.841</strong></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>10</td>
<td>0.826</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>20</td>
<td>0.752</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Performance of surgical task recognition in the 50-video test set ($A_z$)

<table>
<thead>
<tr>
<th>Method</th>
<th>$(P, \hat{s})$ adaptation</th>
<th>$(P, \hat{\Pi})$ adaptation</th>
<th>$(P, \hat{\Pi})$ adaptation / single-class MIL</th>
<th>Previous [4]</th>
<th>Duchenne et al. [20]</th>
</tr>
</thead>
<tbody>
<tr>
<td>incision</td>
<td>0.771</td>
<td><strong>0.816</strong></td>
<td>0.780</td>
<td>0.741</td>
<td>0.801</td>
</tr>
<tr>
<td>rhexis</td>
<td>0.796</td>
<td>0.814</td>
<td>0.805</td>
<td><strong>0.878</strong></td>
<td>0.837</td>
</tr>
<tr>
<td>hydrodissection</td>
<td>0.702</td>
<td>0.753</td>
<td>0.742</td>
<td><strong>0.762</strong></td>
<td>0.719</td>
</tr>
<tr>
<td>phacoemulsification</td>
<td>0.911</td>
<td><strong>0.944</strong></td>
<td>0.911</td>
<td>0.923</td>
<td>0.912</td>
</tr>
<tr>
<td>epinucleus removal</td>
<td>0.882</td>
<td>0.924</td>
<td>0.900</td>
<td><strong>0.969</strong></td>
<td>0.946</td>
</tr>
<tr>
<td>viscous agent injection</td>
<td>0.860</td>
<td><strong>0.920</strong></td>
<td>0.641</td>
<td>0.561</td>
<td>0.614</td>
</tr>
<tr>
<td>implant setting-up</td>
<td>0.779</td>
<td><strong>0.821</strong></td>
<td>0.798</td>
<td>0.703</td>
<td>0.792</td>
</tr>
<tr>
<td>viscous agent removal</td>
<td><strong>0.765</strong></td>
<td>0.737</td>
<td>0.764</td>
<td>0.729</td>
<td>0.695</td>
</tr>
<tr>
<td>stitching up</td>
<td>0.883</td>
<td>0.932</td>
<td>0.955</td>
<td>0.883</td>
<td><strong>0.982</strong></td>
</tr>
<tr>
<td>average</td>
<td>0.817</td>
<td><strong>0.851</strong></td>
<td>0.811</td>
<td>0.794</td>
<td>0.811</td>
</tr>
<tr>
<td>standard error</td>
<td>0.023</td>
<td>0.027</td>
<td>0.032</td>
<td>0.043</td>
<td>0.041</td>
</tr>
</tbody>
</table>
Figure 5: Performance of surgical task recognition with $(\hat{P}, \hat{\Pi})$ adaptation in the 50-video test set
Using one core of an Intel Xeon(R) processor running at 2.53GHz, the proposed approach processed 24.4 frames per seconds (FPS) while our previous method processed 24.3 FPS and Duchenne’s method processed 0.72 FPS. Our methods were implemented in C++ using OpenCV\(^1\) and LIBSVM\(^2\). The most computationally-intensive part of Duchenne’s method, namely Space-Time Interest Point extraction [14], is OpenCV code provided by the authors\(^3\). The rest of the method was implemented in C++, also using OpenCV and LIBSVM. SVM classification takes less than 0.01 milliseconds on average, due to the limited number of input dimensions (one dimension per task) and the limited number of samples (there are 132 support vectors per classifier on average).

To assess the overall accuracy of the system, we performed an additional experiment where the nine task recognition SVMs are run simultaneously. The largest SVM prediction response defines the most likely surgical task. An overall accuracy of 75.9% was achieved by the system, which is significantly larger than the overall accuracy achieved by Duchenne’s method (69.3%, \(p < 0.0001\)) or by the previous system (72.9%, \(p = 0.0468\)) [4]. P-values were computed using two-sided exact binomial tests.

9.3 Results of the Joint Segmentation and Recognition Experiment

To segment idle phases, an ‘action detector’ was trained using (\(\hat{P}, \hat{\Pi}\)) adaptation (§8.3). Using the same segmentation parameters as in [5], a false positive rate (FPR) of 3.2 was measured while achieving the sensitivity reported in [5] (0.846). This is comparable to the FPR reported in [5] (3.5).

Regarding the recognition step, the optimal parameter values obtained in the first experiment, including the polynomial bases and the adapted spatiotemporal polynomials, were used in this second experiment. Other parameters were trained as described in [5]. The performance of the proposed method was compared to our previous method for surgical task segmentation and categorization [5] in terms of \(A_z\) (see table 3). ROC curves are reported in Fig. 6. An overall accuracy of 81.2% was achieved by the system, which is significantly higher than the overall accuracy achieved by the previous system (79.3%, \(p = 0.0342\)) [5].

Note that surgical task segmentation is faster than categorization. It can be run concurrently in a different thread, so jointly segmenting and recognizing surgical tasks does not impact the real-time performance of the system.

10 Discussion

A novel framework for task segmentation and recognition in cataract surgery videos was presented in this paper. In order to normalize motion information in videos, each frame was registered to a coordinate system attached to the anterior segment of the eye. Then, the motion content of short video subsequences was modeled using spatiotemporal polynomials. Next, for each surgical task, key spatiotemporal polynomials were identified through multiple-instance learning. These key spatiotemporal polynomials were then searched in new videos to detect key surgical gestures and therefore recognize the target surgical tasks. To improve recognition performance, the basis of spatiotemporal polynomials was adapted and a support-vector machine was used to combine

\(^1\)http://opencv.org
\(^2\)http://www.csie.ntu.edu.tw/~cjlin/libsvm/
\(^3\)http://www.di.ens.fr/~laptev/download.html
Figure 6: Performance of surgical task recognition in the 93 automatically segmented test videos
Table 3: Performance of surgical task recognition ($A_x$) in the 93 automatically segmented test videos

<table>
<thead>
<tr>
<th>Method</th>
<th>Proposed</th>
<th>Previous [5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>incision</td>
<td>0.953</td>
<td>0.943</td>
</tr>
<tr>
<td>rhexis</td>
<td>0.784</td>
<td>0.850</td>
</tr>
<tr>
<td>hydrodissection</td>
<td>0.820</td>
<td>0.883</td>
</tr>
<tr>
<td>phacoemulsification</td>
<td>0.920</td>
<td>0.891</td>
</tr>
<tr>
<td>epinucleus removal</td>
<td>0.866</td>
<td>0.840</td>
</tr>
<tr>
<td>viscous agent injection</td>
<td>0.848</td>
<td>0.722</td>
</tr>
<tr>
<td>implant setting-up</td>
<td>0.875</td>
<td>0.810</td>
</tr>
<tr>
<td>viscous agent removal</td>
<td>0.855</td>
<td>0.768</td>
</tr>
<tr>
<td>stitching up</td>
<td>0.911</td>
<td>0.863</td>
</tr>
<tr>
<td>miscellaneous</td>
<td>0.728</td>
<td>0.748</td>
</tr>
<tr>
<td>average</td>
<td>0.856</td>
<td>0.832</td>
</tr>
<tr>
<td>standard error</td>
<td>0.021</td>
<td>0.022</td>
</tr>
</tbody>
</table>

multiple key spatiotemporal polynomial detectors. In a dataset of 900 surgical task videos from 100 cataract surgeries, the proposed method compared favorably to our previous method for cataract surgical task recognition [4]. It also compared favorably to Duchenne’s method [20] for human action recognition. In a dataset of 186 videos from full cataract surgeries, if compares favorably to our previous method for the joint segmentation and recognition of cataract surgery tasks [5].

Our previous method for cataract surgical task recognition also relied on the extraction of short video subsequences. However, motion was analyzed separately along the spatial dimensions and along the temporal dimension. The proposed approach does not have this limitation. The spatiotemporal analysis performed in this new solution does not imply larger computation times (24.4 FPS, as opposed to 24.3 FPS).

The comparison to Duchenne et al.’s method is particularly interesting since it is a typical example of the Bag-of-visual-Words (BoW) model frequently used in computer vision. In the BoW model, we are mainly interested in local image/motion patterns and the global image/motion pattern is ignored. We believe the proposed approach is more relevant in a surgery context for two reasons. First, motion information tends to be more global in surgery videos than in general purpose videos: typically, when the surgeon moves a tool, the surrounding tissues are affected. One advantage of the proposed method is that it captures both local and global motion, through high-order and low-order polynomials respectively. Second, Duchenne’s method only characterizes motion in the surrounding of salient points. Therefore, it fails to characterize the deformation of smooth tissues. To summarize, we believe the proposed method is good at capturing deformations at multiple scales. In terms of computation times, the proposed method is advantageous. As opposed to Duchenne’s method, it allows real-time analysis of videos (24.4 FPS, as opposed to 0.72 FPS). The main reason for that difference is that the proposed method relies on the extraction of 2-D interest points (§4) while Duchenne’s method relies on the extraction of 3-D (spatiotemporal) interest points. In Duchenne’s method, once 3-D interest points are detected, local spatiotemporal characterizations are extracted at their location, so that local motions can be finely analyzed. In the proposed method, the underlying 3-D motion information is captured afterwards using polynomials approximations, which is very fast.
Compared to previous works on multiple-instance learning and spatiotemporal polynomials, the main novelty lies in the design of an adaptive basis of spatiotemporal polynomials and the generalization of diverse density for basis adaptation. Adapting the polynomial basis, rather than training a set of weights improves performance significantly (see table 2). This solution increases the dimensionality of the optimization problem. However, the principal component analysis seems to provide a good initial solution for the adaptive polynomial basis, which makes the optimization process easier. We believe the use of a classifier to combine multiple key instance detectors, one detector being trained for a different target class, is also novel. This solution is particularly helpful when no good key instance detector is found for a given class. In our experiment, this was the case for the ‘viscous agent injection’ class: an area $A_z = 0.641$ was found under the ROC curve of $Pr(+|V, \hat{P}, \bar{P})$, as opposed to $A_z = 0.920$ using the classifier (see table 2). Overall, this multiclass MIL solution improves performance significantly.

One limitation of the proposed method is that motion is the only visual feature used for surgical task recognition, although color is used to detect the pupil center and therefore normalize motion information. However, including other visual features may significantly improve performance. In particular, linking the location of surgical tools with the detected motion seems promising. This may be done through the proposed spatiotemporal polynomial framework. So far, our spatiotemporal polynomials are two-dimensional: the first dimension is associated with horizontal motion, the second is associated with vertical motion. In future works, additional features such as tool detections will be included in the framework as additional dimensions of the spatiotemporal polynomials.

In conclusion, a novel action recognition framework has been presented. This framework seems particularly suited to surgery videos and an experiment on a cataract surgery dataset confirmed it. However, it could be beneficial to other problems where relevant visual information does not necessarily lie in the neighborhood of salient points and where motion, and possibly other visual features, cannot be easily segmented.

References


