Real-Time Recognition of Surgical Tasks in Eye Surgery Videos

Gwénolé Quellec[a,*], Katia Charrière[b,a], Mathieu Lamard[c,a], Zakarya Droueche[b,a], Christian Roux[b,a], Béatrice Cochener[c,a,d], Guy Cazuguel[b,a]

[a] *Inserm, UMR 1101, Brest, F-29200 France*
[b] *INSTITUT Mines-Télécom; TELECOM Bretagne; UEB; Dpt ITI, Brest, F-29200 France*
[c] *Univ Bretagne Occidentale, Brest, F-29200 France*
[d] *CHRU Brest, Service d'Ophtalmologie, Brest, F-29200 France*

Abstract

Nowadays, many surgeries, including eye surgeries, are video-monitored. We present in this paper an automatic video analysis system able to recognize surgical tasks in real-time. The proposed system relies on the Content-Based Video Retrieval (CBVR) paradigm. It characterizes short subsequences in the video stream and searches for video subsequences with similar structures in a video archive. Fixed-length feature vectors are built for each subsequence: the feature vectors are unchanged by variations in duration and temporal structure among the target surgical tasks. Therefore, it is possible to perform fast nearest neighbor searches in the video archive. The retrieved video subsequences are used to recognize the current surgical task by analogy reasoning. The system can be trained to recognize any surgical task using weak annotations only. It was applied to a dataset of 23 epiretinal membrane surgeries and a dataset of 100 cataract surgeries. Three surgical tasks were annotated in the first dataset. Nine surgical tasks were annotated in the second dataset. To assess its generality, the system was also applied to a dataset of 1,707 movie clips in which 12 human actions were annotated. High task recognition scores were measured in all three datasets. Real-time

*LaTIM - Bâtiment 2bis (I3S) - CHU Morvan - 5, Av. Foch
29609 Brest CEDEX - FRANCE
Tel.: +33 2 98 01 81 29 / Fax: +33 2 98 01 81 24
Email address: gwenole.quellec@inserm.fr (Gwénolé Quellec)

task recognition will be used in future works to communicate with surgeons (trainees in particular) or with surgical devices.

Keywords: CBVR, real-time, surgical task recognition, eye surgery

1. Introduction

Nowadays, many surgeries are video-monitored. We believe real-time video monitoring may be useful to automatically communicate information to the surgeon in due time. Typically, whenever the surgeon begins a new surgical task, relevant information about the patient, the surgical tools, etc., in connection to this task, may be communicated to him or her (either visually or phonically). The advantage is obvious for 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: these recommendations would derive from the experience of their peers in similar surgeries. A first step towards that goal is presented in this paper: we describe an algorithm to detect surgical tasks in real-time during the surgery.

In recent years, a few systems have been presented for the automatic recognition of surgical tasks or gestures. A first group of systems assumes that the surgical tasks or gestures follow a predefined order: the goal is to find when each task or gesture ends and when the next one begins. Blum et al. (2010) proposed a system to segment such surgical tasks in laparoscopic videos. During the training phase, tool usage is analyzed to perform dimension reduction on visual features, using canonical correlation analysis. At the end of the surgery, the video is registered to a manually segmented average surgery, using Dynamic Time Warping (DTW). Note that a similar system was presented by Padoy et al. (2012): the main difference is that it processes tool usage directly as observations, rather than visual features. Lalys et al. (2011) also proposed a similar system for microscope videos. Many visual features are extracted from images, including color histograms, Haar-based features and SIFT descriptors. Then, the surgery is temporally segmented in surgical tasks using the DTW. In a second group of systems, the DTW is replaced by a Hidden Markov Model (HMM) in order to relax the 'predefined order' hypothesis, although transitions between surgical tasks or gestures that are not seen in training will have a null probability. Blum et al. (2010), Padoy et al. (2012) and Lalys et al. (2011) proposed a variation on their technique described above, where the DTW is replaced by a HMM. Tao et al.

(2012) proposed a system for segmenting a surgical task into a sequence of gestures, in laparoscopic videos. The system relies on *sparse HMMs*, whose observations are sparse linear combinations of elements from a dictionary of basic surgical motions; a dictionary is learnt for each gesture. A third group of systems assumes that the tasks or gestures have already been segmented and the goal is to classify each segmented task or gesture without contextual information. In that case, the predefined order hypothesis is completely relaxed. Haro et al. (2012) evaluated two approaches to surgical gesture classification in video clips. The first one is based on a linear dynamical system; the other is based on the Bag-of-Words (BoW) model (Harris, 1954; Huang et al., 2012; Tamaki et al., 2013; Lalys et al., 2011). These two approaches combined perform equally well as gesture classification based on kinematic data (Haro et al., 2012). Finally, note that several systems have been designed for related tasks: surgical tool detection and tracking (Cano et al., 2008), surgical task detection without categorization (Cao et al., 2007; Giannarou and Yang, 2010), surgical skill evaluation (Reiley and Hager, 2009), etc. Like the third group of recognition methods, the proposed system does not assume that the surgical tasks follow a predefined order and it requires segmented surgical tasks for training. After training, the proposed system can detect, in real-time, key video subsequences that typically occur during a given task, but not during other tasks. This detection does not require any segmentation. The proposed system can also categorize a task as a whole in real-time. But in that case, like the third group of methods, it assumes that the task is segmented.

Similar systems, without the real-time constraint, have been proposed outside the scope of surgery videos. They all rely on a collection of videos containing instances of the target actions for supervision. Piriou et al. (2006) proposed an action recognition framework for sport video indexing. A global probabilistic motion model is trained for each target action. To process a new video, the camera motion is first estimated and removed. Then, the residual motion is analyzed to classify the current action by maximum a posteriori estimation. Duchenne et al. (2009) presented a weakly supervised action recognition framework for movie clip indexing. First, spatiotemporal interest points are detected in videos. Then, video subsequences are characterized using a BoW model. Finally, subsequences containing the target action are detected using a weakly supervised SVM classifier. Xu and Chang (2008) proposed an event recognition framework for news video indexing. A BoW representation was also adopted to characterize varying-sized video

subsequences. To compare two sequences, a variation on the EMD distance between subsequence characterizations was used. These frameworks were primarily designed for offline indexing of broadcast video, so they do not need to be run in real-time.

In order to detect key subsequences of surgical tasks, the proposed system relies on the Content-Based Video Retrieval (CBVR) paradigm. Given a video query, CBVR systems search for similar video contents in a video archive. Initially popularized in broadcasting (Naturel and Gros, 2008) and video surveillance (Smeaton et al., 2006; Hu et al., 2007), the use of CBVR is now emerging in medical applications (André et al., 2010; Syeda-Mahmood et al., 2005). We present a novel CBVR system able to perform real-time searches. In this paper, it is used to recognize the current surgical task by analogy reasoning. Section 2 presents the state of the art of CBVR and discusses the specific challenge of real-time CBVR.

The proposed system is applied to eye surgery. In those surgeries, the surgeon wears a binocular microscope and the output of the ophthalmoscope is recorded. Two of the most common eye surgeries are considered in this paper: epiretinal membrane surgery (Dev et al., 1999) and cataract surgery (Castells et al., 1998). Recently, Lalys et al. (2012) adapted their general system (Lalys et al., 2011) for segmenting cataract surgery videos. In the improved system, visual features are only extracted within the pupil only; an automatic pupil segmentation procedure is presented. Good temporal segmentation performances were measured (Lalys et al., 2012). However, that system does not allow real-time recognition of the surgical tasks. It needs to process the entire surgical video before segmenting it, which implies that the segmentation is only available after the end of the surgery. To our knowledge, this paper is the first attempt to recognize eye surgical tasks in real-time.

2. State of the Art of Content-Based Video Retrieval

Many CBVR systems have been presented in the literature. These systems differ by the nature of the objects placed as queries. First, queries can be images (Patel et al., 2010). In that case, the goal is to select videos containing the query image in a reference dataset; these systems are very similar to image retrieval systems. Second, queries can be video shots (Naturel and Gros, 2008; Dyana et al., 2009). In that case, the goal is to find other occurrences of the query shot (Naturel and Gros, 2008), or similar shots (Dyana

et al., 2009), in the reference dataset. Third, queries can be entire videos (André et al., 2010; Syeda-Mahmood et al., 2005). In that case, the goal is to select the most similar videos, overall, in the reference dataset.

CBVR systems also differ by the way videos or video subsequences are characterized. Several systems rely mainly on the detection and characterization of key frames (Juan and Cuiying, 2010; Patel et al., 2010). Others characterize videos or video subsequences directly (Dyana et al., 2009; Gao and Yang, 2010). In the system by Dyana et al. (2009), video shots are characterized by shape parameters and by the evolution of motion vectors over time. In the system by Gao and Yang (2010), spatiotemporal salient objects (i.e. moving objects) are detected in videos and characterized individually; videos are then compared using the Earth-Mover's Distance (EMD), which may be time consuming. The combination of multimodal (visual, audio and textual) information in a retrieval engine has also been proposed (Hoi and Lyu, 2007; Bruno et al., 2008).

Finally, CBVR systems differ by how flexible the distance metrics should be. First, several systems have been proposed to find objects that are almost identical to the query. For instance, Douze et al. (2010) proposed a copy detection system to protect copyrighted videos. In this system, images are compared individually and their temporal ordering is checked after hand. Another system has been proposed by Naturel and Gros (2008) to detect repeating shots in a video stream, in order to automatically structure television video content. However, in most CBVR systems, we are interested in finding videos or video subsequences that are semantically similar but whose visual content can significantly vary from one sequence to another (Juan and Cuiying, 2010; Xu and Chang, 2008; André et al., 2010). In other words, we need distance metrics able to bridge the so-called semantic gap (Smeulders et al., 2000).

In this paper, we present a CBVR system able to detect key subsequences in a video stream and also to categorize surgical tasks. Short video subsequences extracted from the video stream play the role of the query objects. A flexible distance metric is needed. As mentioned above, similar methods have been presented in the literature to solve this problem (Piriou et al., 2006; Duchenne et al., 2009; Xu and Chang, 2008). When searching for similar video subsequences, and not simply video files as a whole, the number of items that should be compared to the query item explodes. And, as opposed to above methods, the proposed system needs to run in real-time. In order to meet the real-time constraint, a very fast similarity metric must

therefore be used to compare video subsequences. In particular, the use of temporally flexible distance metrics such as DTW (Sakoe and Chiba, 1978; Xu and Chang, 2008) is prohibited for time reasons. An alternative solution is proposed: temporal flexibility is directly introduced in the way video subsequences are characterized. The idea is that video subsequences only need to be characterized once, whereas distances need to be computed every time the system processes a new subsequence, for as long as the video archive is used. So it is worth spending time computing a smart characterization for each subsequence.

3. System Overview

Let $\mathcal{A}_j$ be a type of surgical tasks that we would like to recognize in videos, $j = 1, ..., n_{\mathcal{A}}$. The proposed system tries to detect short subsequences that typically occur during tasks of type $\mathcal{A}_j$, but not during other tasks. These "key subsequences" are detected, in real-time, in subsequences of n_j images. Let $V = \{V_1, V_2, ..., V_{n_V}\}$ be a video sequence of n_V images. To detect key subsequences in V, the following steps are performed at each time instant t_i, $i = 2, ..., n_V$:

1. Texture and color features are extracted from V_i, the current image.
2. Motion features are extracted from the optical flow between V_{i-1}, the previous image, and V_i.
3. All features extracted in 1. and 2. are concatenated to form an instant feature vector, noted $\boldsymbol{f}(V_i)$.
4. In order to detect if a key subsequence has occurred in the time interval $]t_{i-n_j}; t_i]$, $i > n_j$, this instant feature vector is combined with previously computed instant feature vectors $\boldsymbol{f}(V_k)$, $k = i - n_j + 1, ..., i - 1$. The resulting feature vector is noted $\boldsymbol{h}(V_{]i-n_j;i]})$, where $V_{]i-n_j;i]}$ denotes the $\{V_{i-n_j+1}, ..., V_i\}$ video subsequence.
5. Feature vector $\boldsymbol{h}(V_{]i-n_j;i]})$ is compared to a pool of feature vectors that were extracted from a dataset of manually annotated videos and the nearest neighbors are retrieved.
6. Based on the number of neighbors that were extracted during surgical tasks of type $\mathcal{A}_j$, the probability p_{ij} that a key subsequence occurred in $V_{]i-n_j;i]}$ is computed.
7. A specific action can be taken if p_{ij} is above a given threshold.

The probability that surgical tasks of type $\mathcal{A}_j$ occurred in video sequence V is defined as the average p_{ij} instant probability, $i = n_j + 1, ..., n_V$. The maximum average probability defines the most likely surgical task in V.

The main novelty in this paper lies in the way instant feature vectors $\boldsymbol{f}$ are combined into subsequence feature vectors $\boldsymbol{h}$. The combined feature vector is built in such a way that it is unchanged by variations in duration and temporal structure among key subsequences of surgical tasks. Therefore, it is possible to perform simple nearest neighbor searches to compute instant probabilities. Without this computational trick, we would have to use complex distance metrics that are unchanged by such variations, but that are also very slow. In other words, the proposed $\boldsymbol{f} \mapsto \boldsymbol{h}$ mapping allows real-time searches. A novel algorithm is proposed to learn the $\boldsymbol{f} \mapsto \boldsymbol{h}$ mapping offline, from a dataset of weakly annotated videos, for each target task. The use of weakly annotated videos, which requires less annotation work from the experts, makes system adaptation more challenging. In particular, a novel feature weighting technique had to be designed.

4. Real-Time Detection of Key Subsequences

This section details how the proposed system processes video V at each time instant t_i, $i \geq 2$. Then, system adaptation is presented in section 5.

4.1. Extracting Instant Features Vectors

The first step is to extract an instant feature vector $\boldsymbol{f}(V_i)$ for the current image, V_i. Any visual feature may be used in this purpose, with one limitation. If we want the system to process videos in real-time, then feature vector extraction must be faster than the image aquisition rate. So, simple features should be used. The feature vector we propose in this paper combines simple texture and color features extracted from V_i and simple motion features extracted from the optical flow between V_{i-1} and V_i. All these features are global features, so they can be computed fast.

In order to characterize both the textural content and the color content of V_i, the wavelet transform of each color channel of V_i is characterized as described hereafter. To allow real-time image characterization, a fast wavelet-based image characterization from our group was used (Quellec et al., 2012). The wavelet transform of each color channel consists of several *subbands*, each containing information extracted at a given scale and along a given direction. For each combination of color channel (3 channels), scale (3 scales)

and direction (3 directions), the distribution of the wavelet coefficients in the associated subband is characterized by a two-parameter model (Quellec et al., 2012). A total of 54 parameters are extracted from V_i.

Motion features are extracted from the optical flow between V_{i-1} and V_i. To this end, strong corners are first detected in V_{i-1}. These corners are selected, among all image pixels p, with respect to the smallest eigen value of matrix M_p below:

$$\begin{cases} M_p = \begin{pmatrix} A_p & B_p \\ B_p & C_p \end{pmatrix} \\ A_p = \sum_{(x,y)\in\mathcal{N}_p} \left(\frac{\partial I_n^i}{\partial x}(x,y)\right)^2 \\ B_p = \sum_{(x,y)\in\mathcal{N}_p} \frac{\partial I_n^i}{\partial x}(x,y) \cdot \frac{\partial I_n^i}{\partial y}(x,y) \\ C_p = \sum_{(x,y)\in\mathcal{N}_p} \left(\frac{\partial I_n^i}{\partial y}(x,y)\right)^2 \end{cases} \quad (1)$$

where $\mathcal{N}_p$ is a neighborhood of pixel p. Then, the optical flow between V_{i-1} and V_i is computed at each strong corner by the Lucas-Kanade iterative method (Lucas and Kanade, 1981). The OpenCV[1] library was used to select strong corners and compute the optical flow. Finally, motion is characterized by one 8-bin amplitude histogram, two 8-bin amplitude-weighted spatial histograms (one for the x-coordinates and one for the y-coordinates) and one 8-bin amplitude-weighted directional histogram. A total of 32 features are extracted from the optical flow between V_{i-1} and V_i. Overall, the size of $\boldsymbol{f}(V_i)$ is 86 (54+32).

4.2. Extracting Subsequence Feature Vectors

Before describing precisely how subsequence feature vectors are extracted, let us first illustrate the underlying idea with an example taken from cataract surgery, namely "implant insertion". Depending on how weak the expert annotations are, implant insertion can either be a task in itself or a key subsequence in the more general "implant setting-up" task, which would also include implant positioning after the insertion. In Fig. 1, implant insertions are decomposed into four basic actions that are always visible in implant insertion subsequences: tool insertion, implant injection, tool removal, termination. Their duration can vary. The duration of the time interval between basic actions can vary. The same action can occur several times (as

[1] `http://opencv.willowgarage.com/wiki/`

illustrated in V^3). Other actions may occur within the subsequence (as also illustrated in V^3). In fact, the only constraint is that each basic action occurs at least once into a predefined time interval, relative to the current image, called "basic image interval". These basic image intervals are illustrated respectively in violet, in green, in blue and in red in Fig. 1. Whenever all four basic actions have been detected in their respective basic image intervals, an implant injection is detected.

Without any time constraints, this problem could be solved using DTW on the instant feature vectors, for instance. However, to achieve real-time detection, we need to perform fast comparisons. And search algorithms allowing fast comparisons only work with simple distance metrics, such as the Euclidean distance. So we need to create a fixed-size feature vector that is unchanged by all the allowed variations mentioned above.

4.2.1. Temporal Setup of the Subsequence Feature Vector

Let m_j denote the number of basic image intervals for the target surgical task $\mathcal{A}_j$. Each basic image interval is defined by a relative starting point $t_{j,b}$ and a length $\Delta t_{j,b}$, $b = 1, ..., m_j$. The size of the video subsequences is defined as follows:

$$n_j = \max_{b=1,...,m_j} \{t_{j,b} + \Delta t_{j,b}\} \tag{2}$$

Let $\mathcal{T} = \{(t_{j,b}, \Delta t_{j,b}), b = 1, ..., m_j\}$ denote the temporal setup. This temporal setup should be adapted to each target task. Its length in particular: short tasks without early warning signs do not need long video subsequences, but long tasks or tasks with early warning signs do. Typical examples of temporal structures are given in Fig. 2. Note that some images in a subsequence may not appear in any basic image interval. It happens if early basic image intervals are defined to detect early warning signs and late basic image intervals are defined to detect the task itself, but what happens in between is not relevant.

In order to detect that each basic action occurred at least once into the associated basic image interval, through a nearest neighbor search, two video subsequences should have similar feature vectors if their first basic image intervals are similar *and* their second basic image intervals are similar *and* etc. This behavior can be achieved if one feature vector is extracted from each basic image interval and if those feature vectors are further concatenated. Let $\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]})$ denote the feature vector extracted from the $V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]}$ basic image interval.

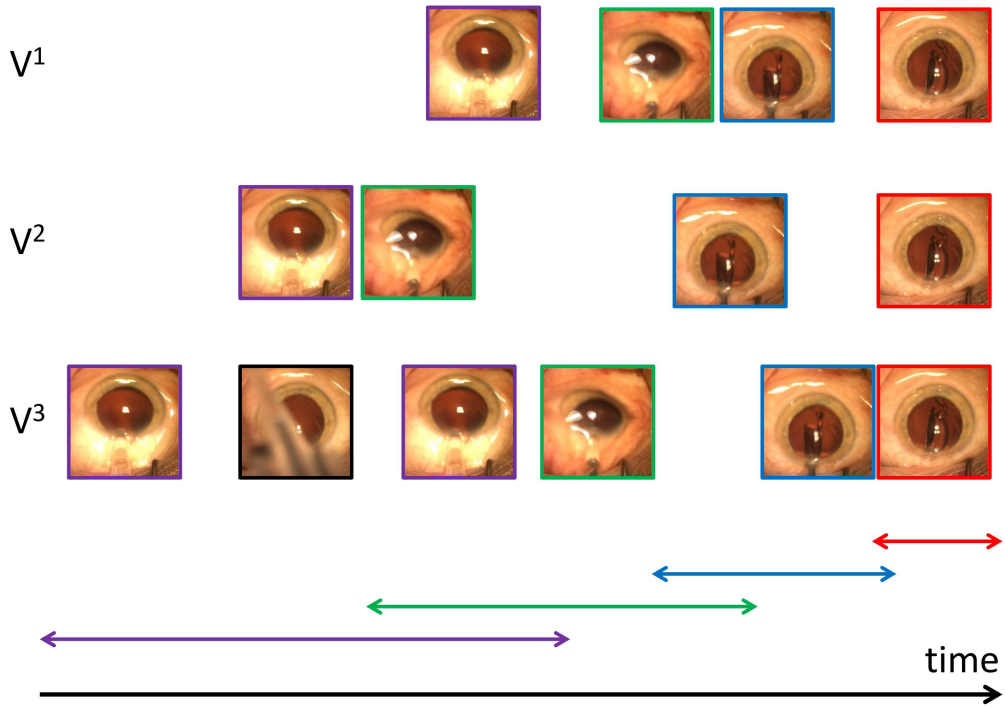


Figure 1: Temporal setup of implant insertion. This figures shows three video subsequences, V^1, V^2 and V^3, in which an implant is inserted into the patient's eye. This task is performed at different paces from one subsequence to another. These subsequences are registered at the end of the task. Note that a small complication occurred in the third subsequence: the surgeon had to remove his tool and insert it again later. The violet interval indicates a range of dates for tool insertion, relatively to the end of the task. The green interval indicates a range of dates for implant injection. The blue interval indicates a range of dates for tool removal. The red interval indicates a range of dates for task termination (when the tool becomes invisible and the implant starts to unfold).

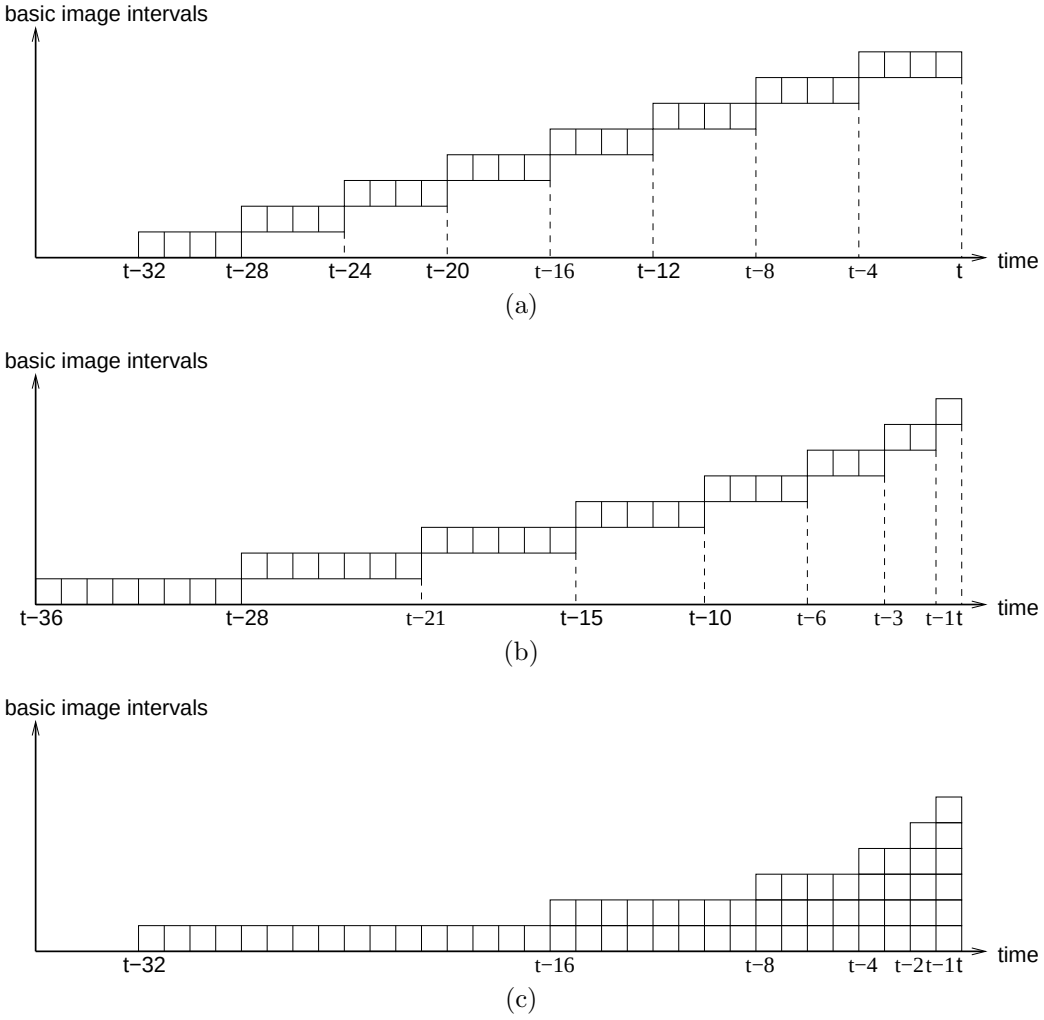


Figure 2: Examples of temporal setups. In those examples, each image is represented by one column. In examples (a) and (c), each video subsequence consists of $n_j = 32$ images; in example (b), each video subsequence consists of $n_j = 36$ images. Each basic image interval is represented by one row. In examples (a) and (b), each video subsequence consists of $m_j = 8$ basic image intervals; in example (c), each video subsequence consists of $m_j = 6$ basic image intervals. If one image appears in several rows, like in example (c), it means that two or more basic image intervals overlap. Note that there are no overlaps in examples (a) and (b).

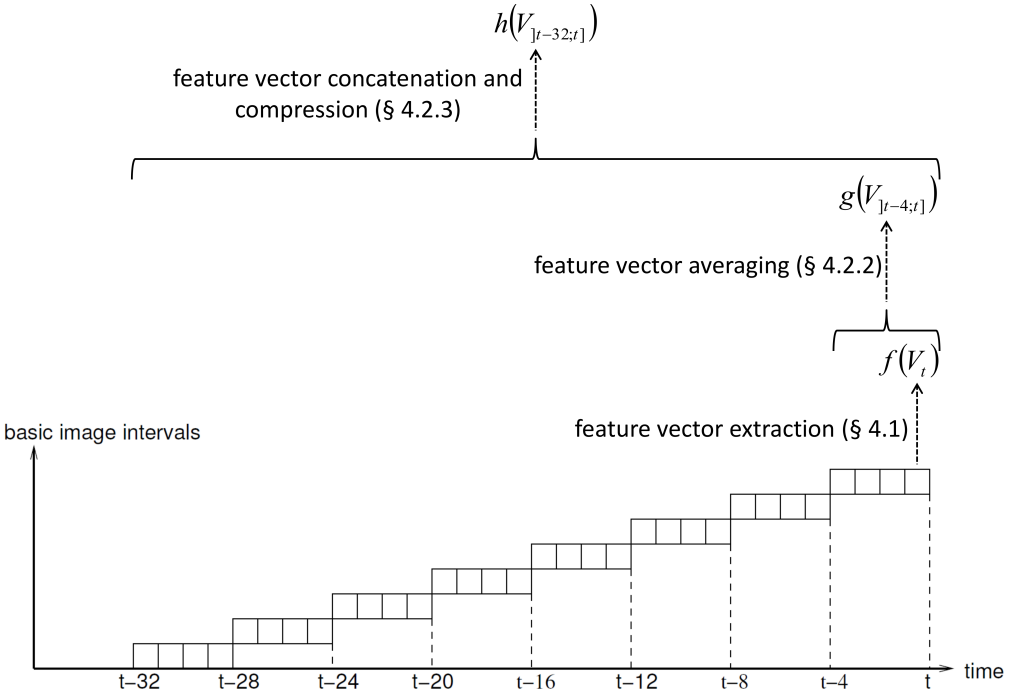


Figure 3: Extracting subsequence feature vectors

In order to characterize the presence of the b^{th} basic action into the $V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]}$ interval, given the allowed variations, $\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]})$ should be unchanged if its temporal structure changes. In particular, whether some discriminative event occurs in image V_k *or* in image V_l should not affect the characterization of $V_{]i-n_j;i]}$ so long as, for all b,

- $i - t_{j,b} - \Delta t_{j,b} \geq k, l,$
- or $i - t_{j,b} - \Delta t_{j,b} < k, l \leq i - t_{j,b},$
- or $k, l > i - t_{j,b}.$

Subsequence feature vector extraction is summarized in Fig. 3 and detailed in the following paragraphs.

4.2.2. Characterizing one Basic Image Interval

Each basic image interval $V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]}$ is characterized by the following average feature vector:

$$\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]}) = \frac{1}{\Delta t_{j,b}} \sum_{k=i-t_{j,b}-\Delta t_{j,b}+1}^{i-t_{j,b}} \boldsymbol{f}(V_k) \tag{3}$$

Clearly, whether some discriminative event occurs in image V_k or in image V_l, with $i - t_{j,b} - \Delta t_{j,b} < k, l \leq i - t_{j,b}$, will not affect $\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]})$.

4.2.3. Characterizing the Video Subsequence

As mentioned above, subsequence $V_{]i-n_j;i]}$ may first be characterized by the concatenation of all $\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]})$ vectors, $b = 1, ..., m_j$. Let $\bar{\boldsymbol{h}}(V_{]i-n_j;i]})$ be that compound feature vector. Note, however, that basic image intervals in a subsequence are likely correlated. It follows that feature vectors $\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]})$, $b = 1, ..., m_j$, also are. In order to obtain more compact feature vectors, with less correlated components, $\bar{\boldsymbol{h}}(V_{]i-n_j;i]})$ is projected onto the principal components $\boldsymbol{\Gamma}_{j,c}$, $c = 1, ..., C_j$, obtained by a principal component analysis (Pearson, 1901). Finally, the projection on each principal component is weighted by $\lambda_{j,c} \geq 0$, $c = 1, ..., C_j$. The $\lambda_{j,c}$ weights are chosen to fill the semantic gap between low-level feature vectors and the high-level concept of surgical task, as described in section 5. Let $\boldsymbol{h}(V_{]i-n_j;i]}) = \left\{\lambda_{j,c}\left(\bar{\boldsymbol{h}}(V_{]i-n_j;i]}) \cdot \boldsymbol{\Gamma}_{j,c}\right), c = 1, ..., C_j\right\}$ denote the resulting feature vector.

4.3. Key Subsequence Detection Probability

Working with fixed-length feature vectors, such as $\boldsymbol{h}(V_{]i-n_j;i]})$, has one major advantage: similar feature vectors can be searched with fast algorithms, such as k-d trees (Arya and Mount, 1993) or Locality-Sensitive Hashing (Gionis et al., 1999). ANN[2], a fast variation on k-d trees, was used in this paper to perform nearest neighbor searches in a reference dataset $\mathcal{R}_j$, $j = 1, ..., n_{\mathcal{A}}$. In this reference dataset, every feature vector is associated with one binary label, indicating whether or not this feature vector was extracted during a surgical task of type $\mathcal{A}_j$. The probability p_{ij} that a key subsequence of type $\mathcal{A}_j$ occurred in $V_{]i-n_j;i]}$ is defined as the proportion of feature vectors with

[2] `http://www.cs.umd.edu/~mount/ANN/`

a positive label among the K nearest neighbors of $\boldsymbol{h}(V_{]i-n_j;i]})$ in $\mathcal{R}_j$. In all experiments, the number of nearest neighbors was set to $K = 5$.

4.4. Real-time Processing Pipeline

This section describes, in practical terms, how the system can be used to recognize several surgical tasks $\mathcal{A}_j$, $j = 1, ..., n_{\mathcal{A}}$, simultaneously, in real-time. Let $N = \max_{j=1,...,n_{\mathcal{A}}}(n_i)$ denote the maximal subsequence length. At each time instant t_i, $i \geq 2$:

1. The instant feature vector $\boldsymbol{f}(V_i)$ is characterized.
2. $\boldsymbol{f}(V_i)$ is stored in a FIFO queue of size N.
3. For each surgical task $\mathcal{A}_j$, the last n_j instant feature vectors in the FIFO queue are combined: the subsequence feature vector $\boldsymbol{h}(V_{]i-n_j;i]})$ is obtained.

It should be noted that the first step, which is typically the most computationally intensive, is independent of the target surgical task; therefore, it is only performed once. The FIFO queue is only useful if two consecutive subsequences overlap, that is if $N > 1$. In that case, instant feature vectors are stored and are used up to $\sum_{j=1}^{n_{\mathcal{A}}} n_j$ times to compute subsequence feature vectors.

5. System Adaptation

Let $\mathcal{D}$ be a dataset of video sequences, divided into a training subset $\mathcal{D}_{train}$ and a test subset $\mathcal{D}_{test}$. For training and testing purposes, it is assumed that experts indicated whether or not each target surgical task occurred in each video sequence $V \in \mathcal{D}$. Note that experts are not expected to indicate precisely when target tasks occurred in each video sequence. They are only expected to assign $n_{\mathcal{A}}$ binary labels $\delta(\mathcal{A}_j, V)$, $j = 1, ..., n_{\mathcal{A}}$, to each video $V \in \mathcal{D}$: $\delta(\mathcal{A}_j, V) = true$ if at least one $\mathcal{A}_j$ task occurred in V, $\delta(\mathcal{A}_j, V) = false$ otherwise.

The proposed system has three undetermined sets of parameters: a temporal setup, principal components, and a weight vector (see section 4.2). These parameters need to be tuned specifically for each target task $\mathcal{A}_j$, $j = 1, ..., n_{\mathcal{A}}$. System adaptation can be summarized as follows. After the summary, a more detailed explanation of each point is given.

1. Temporal setups are generated at random using a stochastic algorithm.

2. For each temporal setup,
   (a) the principal components are computed,
   (b) a weight vector is optimized,
   (c) a performance score is computed. Because experts are only expected to interpret video sequences as a whole (through binary labels $\delta(\mathcal{A}_j, V)$), semantic relevance is assessed at video sequence level.
3. The temporal setup maximizing the performance score in the training set is retained.

5.1. Adapting the Temporal Setup

The main novelty in this paper comes from the way instant feature vectors are combined into subsequence feature vectors, using a task specific temporal setup. The main question we need to answer for system adaptation is the following: what temporal setup best captures $\mathcal{A}_j$ tasks? This question is actually twofold. First, we need to know how many basic image intervals are needed (i.e. what is the value of m_j?). Second, we need to choose the relative starting point and the length of each basic image interval $(t_{j,b}, \Delta t_{j,b})$, $b = 1, ..., m_j$. Given the high dimensionality of this problem, a machine learning solution is desirable. Optimization techniques able to work with variable-length vectors $\mathcal{T} = \{(t_{j,b}, \Delta t_{j,b}), b = 1, ..., m_j\}$ are suitable candidates. A stochastic solution based on Grammatical Evolution (GE) is proposed hereafter (O'Neill and Ryan, 2001). A boosting strategy is also possible (Schapire, 1990).

In GE, a genetic algorithm is used to generate variable-length *strings* or genotypes. These strings are then mapped to *programs* through a user-defined *grammar*. The objective is to find the program that maximizes a user-defined objective function (O'Neill and Ryan, 2001).

In order to fit into the GE paradigm, the temporal setup $\mathcal{T} = \{(t_{j,b}, \Delta t_{j,b}), b = 1, ..., m_j\}$ is regarded as a program. The objective function that should be maximized is the performance score described in section 5.4. Let $\mathcal{S}_t$ and $\mathcal{S}_{\Delta t}$ denote a set of relative starting points and a set of lengths, respectively. The grammar used to generate temporal setups is given in algorithm 1. In a grammar, symbols are delimited by angled brackets ($< . >$). To generate programs, GE recursively replaces each symbol, starting with symbol $< \mathcal{T} >$, by one expression. The list of valid expressions for each symbol starts with ::= and uses | as delimiter.

Algorithm 1 Temporal Setup Grammar

$< \mathcal{T} >::= \{< bbi >\}$
$< bbi >::=< bbi >, < bbi > \ | \ (< t >, < \Delta_t >)$
$< t >::= \mathcal{S}_{t,1} \mid \mathcal{S}_{t,2} \mid ...$
$< \Delta_t >::= \mathcal{S}_{\Delta t,1} \mid \mathcal{S}_{\Delta t,2} \mid ...$

In all experiments, the following set of relative starting points was used: $\mathcal{S}_t = \{0, 1, 2, 5, 10, 20, 50, 100\}$. The following set of lengths was used: $\mathcal{S}_{\Delta t} = \{1, 2, 5, 10, 20, 50, 100\}$. This grammar generates a variable-length list of $(< t >, < \Delta_t >)$ tuples. In the first line of the grammar, the list is initialized by a symbol with undefined value: $< bbi >$. In the second line of the grammar, the list is expanded by recursively duplicating the $< bbi >$ symbol whenever the first valid expression in that line $(< bbi >, < bbi >)$ is randomly chosen by the GE engine. The expansion stops when all $< bbi >$ symbols have been replaced by a $(< t >, < \Delta_t >)$ tuple: this happens when the second valid expression of line 2 is chosen. Then, the GE engine randomly chooses a value for each relative starting point $< t >$ and each length $< \Delta_t >$ using the third and the fourth line of the grammar, respectively. Populations of 30 programs were evolved by grammatical evolution. Grammatical evolution was stopped when the optimal solution did not evolve for more than 5 generations. The proposed learning algorithm does not ensure that the first basic image interval starts at $t = 0$. Therefore, as a post processing step, all relative starting points are translated such that the first one equals 0.

5.2. Principal Component Analysis

The principal components $\mathbf{\Gamma}_{j,c}$, $c = 1, ..., C_j$, are obtained by a Principal Component Analysis (PCA) (Pearson, 1901). The dataset analyzed by the PCA algorithm is formed by $\bar{\boldsymbol{h}}(V_{]i-n_j;i]})$ subsequence feature vectors extracted from the training subset $\mathcal{D}_{train}$, using the current temporal setup. As commonly done in the literature, C_j was chosen such that 90% of the energy is preserved (Ricci et al., 2011). Note that the projections $\bar{\boldsymbol{h}}(V_{]i-n_j;i]}) \cdot \mathbf{\Gamma}_{j,c}$ on the principal components do not depend on the order of the features in the input training samples. It implies that the order in which the $\boldsymbol{g}(V_{]i-t_{j,b}-\Delta t_{j,b};i-t_{j,b}]})$ vectors are concatenated to form the $\bar{\boldsymbol{h}}(V_{]i-n_j;i]})$ vectors does not matter.

5.3. Adapting the Weight Vector

A weight vector $\boldsymbol{\lambda}_j$ has been introduced in section 4.2 to bridge the semantic gap between low-level subsequence feature vectors and the high-level concept of 'similar surgical tasks'. Because experts are only expected to interpret video sequences as a whole, weight adaptation cannot be supervised at video subsequence level: it is supervised at video sequence level. A novel feature weighting technique is proposed for this specific setup.

First, a semantic and a low-level distance between video sequences are defined. $D_j^{sem}(U,V)$, the semantic distance between two video sequences U and V, is defined as follows: $D_j^{sem}(U,V)$=0 if $\delta(\mathcal{A}_j,U) = \delta(\mathcal{A}_j,V)$ and $D_j^{sem}(U,V)$=1 otherwise. $D_j^{low}(U,V)$, the low-level distance between U and V, derives from the weight vector $\boldsymbol{\lambda}_j$ and all feature vectors $\boldsymbol{h}(U_{]i-n_j;i]})$ and $\boldsymbol{h}(V_{]i-n_j;i]})$ (see section 5.3.1). In order to maximize the semantic relevance of retrieved video subsequences, $\boldsymbol{\lambda}_j$ is tuned so that D_j^{sem} and D_j^{low} become as close as possible in the training subset $\mathcal{D}_{train}$ (see section 5.3.2). In other words, $\boldsymbol{\lambda}_j$ is chosen to bridge the semantic gap in the least-squares sense.

5.3.1. Low-Level Distance

For each feature vector component $c = 1,...,C_j$, a partial low-level distance between video sequences U and V is defined: $D_{j,c}^{low}(U,V)$. Assuming that U and V both contain the target task, they are considered similar no matter when the target tasks occurred in these videos. Therefore, we simply need to compare the distributions $\mathcal{H}_c(U) = \left\{h_c(U_{]i-n_j;i]}), i = n_j,...,n_U\right\}$ and $\mathcal{H}_c(V) = \left\{h_c(V_{]i-n_j;i]}), i = n_j,...,n_V\right\}$, regardless of time sequencing. $D_{j,c}^{low}(U,V)$ is defined as the maximal deviation between the Cumulative Distribution Function (CDF) of $\mathcal{H}_c(U)$ and the CDF of $\mathcal{H}_c(V)$. In other words, $D_{j,c}^{low}(U,V)$ is the Kolmogorov-Smirnov statistic of the test "$\mathcal{H}_c(U) = \mathcal{H}_c(V)$" (von Mises, 1964).

5.3.2. Bridging the Semantic Gap

Let $T = |\mathcal{D}_{train}|$ be the number of video sequences in the training set. Let $T' = \frac{1}{2}T(T-1)$ be the number of pairs of video sequences. For each video sequence pair (U,V), one semantic distance $D_j^{sem}(U,V)$ and C_j partial low-level distances $D_{j,c}^{low}(U,V)$ have been computed. Semantic distances are grouped together in a vector $\boldsymbol{d}_j^{sem}$ of size T'. Low-level distances are grouped together in a matrix $\mathcal{D}_j^{low}$ of size $(T' \times C_j)$. The weight vector $\boldsymbol{\lambda}_j$ minimizing the sum of the squared errors between $\boldsymbol{d}_j^{sem}$ and $\mathcal{D}_j^{low} \cdot \boldsymbol{\lambda}_j$ is found by multi-

parameter linear fitting with positivity constraints: the nonnegative least squares algorithm is used (Lawson and Hanson, 1974).

5.4. Performance Assessment at Video Sequence Level

The performance of the system is assessed using a reference dataset $\mathcal{R}$ (from which the nearest neighbors are extracted) and a validation dataset $\mathcal{V}$. During system adaptation, $\mathcal{R}$ is a portion of the training subset $\mathcal{D}_{train}$ and $\mathcal{V}$ is the complement of $\mathcal{R}$ in $\mathcal{D}_{train}$. After system adaptation, $\mathcal{R}$ is the entire training subset and $\mathcal{V}$ is the test subset $\mathcal{D}_{test}$.

Performances are assessed in terms of A_z, the area under the Receiver Operating Characteristic (ROC) curve (Hanley and McNeil, 1982). The following procedure is applied to build the ROC curve for a given task $\mathcal{A}_j$. First, for each video sequence $V \in \mathcal{V}$ in the validation set, $p(\delta(\mathcal{A}_j, V))$, the probability that V contains at least one $\mathcal{A}_j$ task, is defined as the average instant probability p_{ij}, $i = n_j + 1, ..., n_V$. Instant probabilities are computed using nearest neighbors extracted from $\mathcal{R}$. Then, the ROC curve of $p(\delta(\mathcal{A}_j, V))$, against the ground truth $\delta(\mathcal{A}_j, V)$, is built using every $p(\delta(\mathcal{A}_j, V))$ value, $V \in \mathcal{V}$, as a threshold.

6. Video Datasets

The proposed framework was evaluated in two eye surgery datasets: a dataset of retinal surgeries (see section 6.2) and a dataset of cataract surgeries (see section 6.3). To assess its generality, the method was also evaluated in a dataset of human actions in general (walking, sitting down, etc. - see section 6.4).

6.1. Weak Supervision for Surgery Videos

In order to evaluate the performance of surgical task detectors, surgical tasks need to be manually delimited in full surgery videos. Note that a surgery is generally not a predefined linear sequence of surgical tasks. For instance, surgeons may have to widen an incision afterwards because they realized the initial incision was not large enough. The "current task" may be combined with recurring tasks such as moisturizing the patient, sucking out blood, etc. Also, surgeons generally have two hands, so two "consecutive tasks" may actually overlap. Therefore, partitioning a surgery into task-specific video sequences is not always possible. For each surgical task, a specialist was asked to indicate the date of the first appearance of one tool

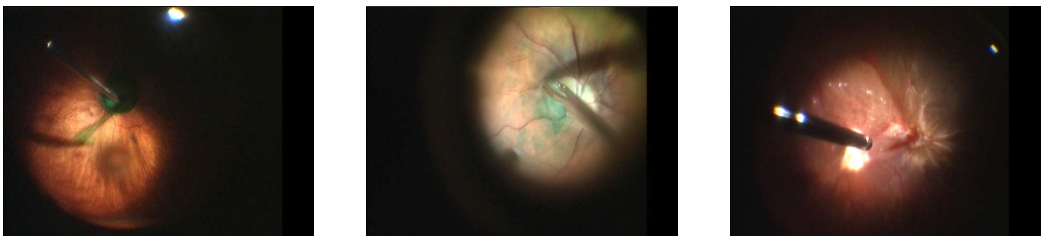
related to this task into the field of view. Similarly, he was asked to indicate the date of the last disappearance of one of these tools from the field of view. These two dates define the beginning and the end of each manually delimited video. As a consequence, these manually delimited videos are weakly annotated: other tasks may appear into the video. It means that, given a manually delimited video V, $\delta(\mathcal{A}_j, V)$ may be true for multiple A_j tasks. Also, these videos often contain time intervals when no task is visible into the field of view: it happens, for instance, if the surgeon needs to change one of his or her tools. Every manually delimited video V was classified (the value of $\delta(\mathcal{A}_j, V)$ was determined for each task $\mathcal{A}_j$) and assigned to the dataset $\mathcal{D}$.

6.2. Retinal Surgery Dataset (RSD)

The first dataset consists of 69 videos of Epiretinal Membrane Surgery (EMS) collected at Brest University Hospital (France). EMS is the most common vitreoretinal surgery (Dev et al., 1999). It involves a pars plana vitrectomy procedure with membrane peeling. Twenty-three consecutive surgeries of twenty-three patients have been video-recorded with a CCD-IRIS device (Sony, Tokyo, Japan) and the videos were stored in MPEG2 format with the highest quality settings. The frame frequency is 25 frames per seconds. Images have a definition of 720x576 pixels. Three video sequences were manually delimited in each surgery video as explained in section 6.1. Each video sequence corresponds to one step of the EMS: Injection, Coat and Vitrectomy (see Fig. 4). The duration of each surgical task is reported in table 1. The dataset was randomly divided into two subsets of 12 and 11 surgeries, respectively. In order to deal with the small number of surgeries in this dataset (23), a special training procedure was adopted. At first, surgical tasks detectors were trained in the first subset and their performance was tested in the second subset. Then, new surgical task detectors were trained in the second subset and their performance was tested in the first subset. Therefore, performance scores were computed for each video sequence.

6.3. Cataract Surgery Dataset (CSD)

The second dataset consists of 900 videos of phacoemulsification cataract surgery collected at Brest University Hospital (France). Cataract surgery is the most common ophthalmic surgery and most of the cataract surgeries are phacoemulsification procedures (Castells et al., 1998). In this procedure, an ultrasonic device is used to break up and then remove a cloudy lens, or



(a) Injection (b) Coat (c) Vitrectomy

Figure 4: Images from different high-level surgical tasks in the Retinal Surgery Dataset (RSD)

Table 1: Duration of surgical tasks

Surgical task		Duration (s)	Percentage of the video
RSD	Injection	39 ± 33	5.2 %
	Coat	380 ± 300	50.6 %
	Vitrectomy	203 ± 151	27.0 %
CSD	Incision	55 ± 77	5.4 %
	Rhexis	87 ± 85	8.5 %
	Hydrodissection	36 ± 52	3.5 %
	Phacoemulsification	205 ± 138	20.0 %
	Epinucleus removal	131 ± 115	12.8 %
	Viscous agent injection	18 ± 36	1.8 %
	Implant setting-up	53 ± 61	5.2 %
	Viscous agent removal	79 ± 155	7.7 %
	Stitching up	182 ± 191	17.7 %

cataract, from the eye to improve vision. One hundred consecutive surgeries of one hundred patients have been video-recorded. Some surgeries 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. The frame frequency is 25 frames per seconds. Images have a definition of 720x576 pixels. Nine video sequences were manually delimited in each surgery video as explained in section 6.1. Each video sequence corresponds to one step of the cataract surgery: Incision, Rhexis, Hydrodissection, Phacoemulsification, Epinucleus removal, Viscous agent injection, Implant setting-up, Viscous agent removal and Stitching up (see Fig. 5). The duration of each surgical task is reported in table 1. The dataset was randomly divided into two subsets of 50 surgeries: one was used as training set, the other was used as test set.

6.4. Movie Clip Dataset (MCD)

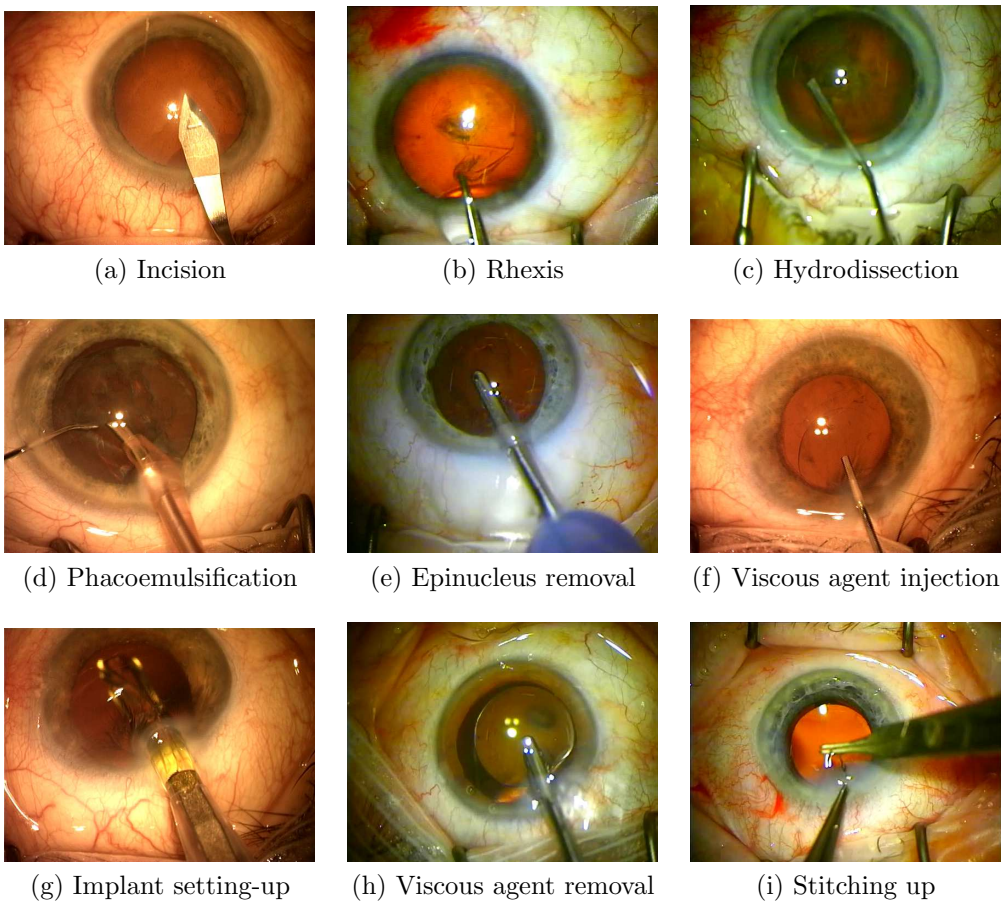
The third dataset consists of 1,707 video sequences extracted by the IRISA lab from 69 Hollywood movies[3] (Marszałek et al., 2009). The authors indicated which human actions were visible in each video sequence, out of 12 possible actions. In most of the dataset, a semi-automatic procedure was used: 1) clips were automatically annotated using text mining techniques in the movie scripts and 2) the automatic annotations were controlled by a human observer. In the *Training subset (automatic)*, the annotations were fully automatic: they were not controlled by a human observer. The frame frequency is 25 frames per seconds. Typical image definitions include 640x352, 576x312 and 548x226. Videos have an average duration of 20 seconds. The test subset consists of 884 video sequences. The training set consists of 823 video sequences: we decided not to use the *Training subset (automatic)* because the annotations have a lower quality.

7. Results

The optimal selections of basic image intervals, obtained in the training subset of each dataset[4], are reported in table 2. In practice, basic image

[3] `http://www.irisa.fr/vista/actions/hollywood2/`

[4] In RSD, a 2-fold cross-validation was performed: we only reported the basic image intervals obtained in the first fold.



(a) Incision (b) Rhexis (c) Hydrodissection

(d) Phacoemulsification (e) Epinucleus removal (f) Viscous agent injection

(g) Implant setting-up (h) Viscous agent removal (i) Stitching up

Figure 5: Images from different high-level surgical tasks in the Cataract Surgery Dataset (CSD)

intervals overlap a lot and such an overlapping allows capturing the variability in the temporal length of the key subsequences. The size of the subsequence feature vectors $\boldsymbol{h}$ ranged from 8 to 32 elements.

The proposed method does not detect the appearance or disappearance of tools in the field of view, unlike the manual segmentation provided by experts. It detects clinically-significant actions, which can happen anytime between the appearance and the disappearance of the associated tools. So the system was assessed at the surgical task level. The performance of the proposed framework in the test subset, in terms of A_z, the area under the

Table 2: Optimal selection of basic image intervals

RSD	Injection	$\{0, 2\}$
	Coat	$\{0, 5\}, \{0, 50\}, \{2, 50\}$
	Vitrectomy	$\{0, 20\}, \{100, 50\}$
CSD	Incision	$\{0, 50\}, \{4, 20\}, \{9, 2\}, \{49, 2\}$
	Rhexis	$\{0, 5\}, \{10, 20\}$
	Hydrodissection	$\{0, 1\}, \{0, 100\}, \{1, 100\}, \{19, 1\}, \{49, 50\}, \{99, 5\}$
	Phacoemulsification	$\{0, 2\}, \{10, 1\}$
	Epinucleus removal	$\{0, 2\}, \{1, 20\}, \{10, 1\}, \{50, 1\}, \{50, 5\}$
	Viscous agent injection	$\{0, 20\}, \{1, 10\}, \{1, 100\}, \{9, 10\}, \{9, 20\}$
	Implant setting-up	$\{0, 100\}, \{1, 2\}, \{1, 10\}, \{100, 2\}$
	Viscous agent removal	$\{0, 1\}$
	Stitching up	$\{0, 100\}$
MCD	AnswerPhone	$\{0, 1\}, \{50, 5\}$
	DriveCar	$\{0, 2\}, \{2, 1\}$
	Eat	$\{0, 10\}, \{1, 10\}, \{1, 100\}, \{10, 5\}, \{20, 2\}, \{20, 5\}, \{50, 1\}, \{100, 100\}$
	FightPerson	$\{0, 20\}, \{2, 1\}$
	GetOutCar	$\{0, 2\}$
	HandShake	$\{0, 2\}, \{1, 2\}$
	HugPerson	$\{0, 1\}, \{4, 2\}, \{99, 100\}$
	Kiss	$\{0, 5\}, \{0, 10\}, \{2, 10\}, \{10, 1\}$
	Run	$\{0, 1\}, \{0, 10\}, \{19, 1\}$
	SitDown	$\{0, 2\}, \{0, 20\}, \{20, 5\}$
	SitUp	$\{0, 1\}, \{3, 20\}, \{18, 5\}$
	StandUp	$\{0, 2\}, \{18, 1\}$

ROC curve, is reported in table 3. The associated ROC curves are reported in Fig. 6, 7 and 8. The proposed framework was compared to the framework by Duchenne et al. (2009) for human action recognition, both in terms of A_z and in terms of computation times. Results are reported in tables 3 and 4, respectively. Our method was implemented in C++ using OpenCV[5]. The most computationally-intensive part of Duchenne's method, namely Space-

[5] `http://opencv.willowgarage.com/wiki/`

Time Interest Point extraction (Laptev, 2005), is OpenCV code provided by the authors[6]. The rest of the method was implemented in C++, also using OpenCV and LIBSVM[7]. All computations were performed using one core of an Intel Xeon(R) E5649 processor running at 2.53GHz. In all three datasets, most of the computation time was dedicated to subsequence characterization, and to image characterization in particular (see section 4.1). Retrieval only took about eight milliseconds per query on average. Because image sizes are lower in MCD, computation times are lower in that dataset (see table 4).

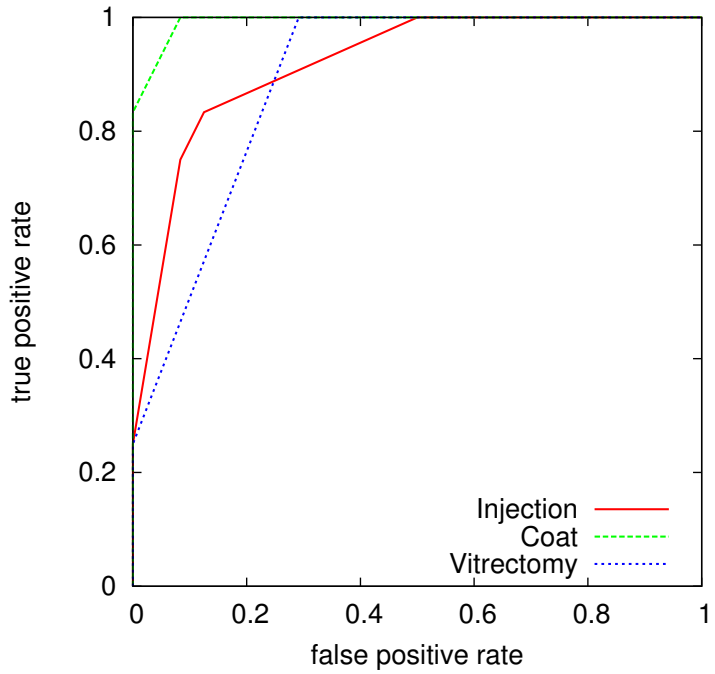


Figure 6: Receiver Operating Characteristic (ROC) curves for the RSD dataset (see section 6.2)

To assess the overall accuracy of the system, we performed an additional experiment where all target task detectors are run simultaneously, as described in section 4.4. This defines the most likely surgical task in each video

[6] http://www.di.ens.fr/~laptev/download.html
[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 3: Performance evaluation (A_z) in the test set

Dataset / Action		Proposed method	Method by Duchenne et al.
RSD	Injection	**0.923**	0.500
	Coat	**0.995**	0.978
	Vitrectomy	0.898	**0.906**
CSD	Incision	0.741	**0.801**
	Rhexis	**0.878**	0.837
	Hydrodissection	**0.762**	0.719
	Phacoemulsification	**0.923**	0.912
	Epinucleus removal	**0.969**	0.946
	Viscous agent injection	0.561	**0.614**
	Implant setting-up	0.703	**0.792**
	Viscous agent removal	**0.729**	0.695
	Stitching up	0.883	**0.982**
MCD	AnswerPhone	**0.801**	0.602
	DriveCar	**0.914**	0.910
	Eat	**0.848**	0.799
	FightPerson	0.905	**0.932**
	GetOutCar	**0.794**	0.782
	HandShake	**0.830**	0.751
	HugPerson	0.691	**0.715**
	Kiss	**0.852**	0.696
	Run	0.873	**0.883**
	SitDown	0.777	**0.832**
	SitUp	**0.800**	0.729
	StandUp	**0.786**	0.778

Table 4: Computation times (frames per second)

Dataset	Proposed method	Method by Duchenne et al.
RSD	26.0 FPS	0.78 FPS
CSD	24.3 FPS	0.72 FPS
MCD	28.9 FPS	1.39 FPS

sequence. A success was recorded if the most likely task indeed happened during the sequence. Otherwise, a failure was recorded. This defines an overall accuracy for the system. Accuracies are reported in table 5 together with p-values computed with two-sided exact binomial tests.

Table 5: Overall accuracies

Dataset	Proposed method	method by Duchenne et al.	p-value
RSD	87.0%	73.9%	0.00269
CSD	72.9%	69.3%	0.0514
MCD	75.0%	71.9%	0.0207

8. Discussion and Conclusion

A novel framework for real-time retrieval of similar video *subsequences* was presented in this paper. Given a target surgical task, video subsequences are decomposed into an optimal set of overlapping basic image intervals. The proposed video description ignores the temporal structure within basic image intervals by averaging instant feature vectors. The temporal structure is only encoded, in a fuzzy fashion, by concatenating the feature vectors describing the basic image intervals composing the subsequence. The proposed framework was applied to real-time recognition of high-level surgical tasks in video-monitored eye surgeries. Two types of surgeries were considered: epiretinal membrane surgery and cataract surgery. For most surgical tasks (7 out of 12), the system compared favorably to a state-of-the-art human action recognition system in terms of area under the ROC curve (see table 3). The proposed method failed to recognize one surgical step, namely viscous agent injections, but so did the baseline method. This was to be expected: injections mostly involve subtle fluid motions within the eye. Note that the largest dataset group together videos recorded with different devices, which makes the recognition task more challenging. To assess its generality, the proposed system was also applied to human action recognition. For most human actions (8 out of 12), the proposed system turned out to be more efficient than the baseline method as well (see table 3). Overall, the proposed method significantly outperforms the baseline method in RSD and MCD, but not in CSD ($p > 0.05$, see table 5). The proposed system is very fast: while the baseline method processes less than one image per second (see table 4),

the proposed system runs nearly in real-time (= 25 frames per second). Note that the cataract surgery phase segmentation system by Lalys et al. (2012) also processes less than one image per second (one image every 3 seconds). One limitation of the proposed method, in terms of computation times, is that training is slow, due to the use of grammatical evolution: on average, training lasts 16 hours per surgical task or action on an Intel Xeon(R) E5649 12-core processor. One strength of the method is that it does not assume that consecutive surgical tasks follow a predefined order, unlike methods by Lalys et al. (2012) and Blum et al. (2010) for instance. One such method (Blum et al., 2010) was implemented and run on our cataract surgery dataset (CSD). For several reasons, surgical videos could not be registered to a manually segmented average surgery. First, several surgeons, with varying skill levels and varying techniques, participated in the study. Second, surgical tasks usually overlapped in those videos. Third, in some surgeries, additional tasks had to be performed, such as removing an old implant for instance.

We believe this success comes from the high flexibility in the adaptation process. It can be seen in table 2 that very different temporal setups were obtained depending on the target task. However, it should be noted that most temporal setups are quite simple (see table 2): they usually consist of two or three basic image intervals. This prevents overfitting and ensures good performances in the test set. This good property comes from the use of grammatical evolution, which penalizes complex solutions in accordance with Occam's razor principle (O'Neill and Ryan, 2001).

Note that, although the proposed system is particularly well suited to eye surgeries, it could be applied to other surgeries where video-recording is available, for instance minimally invasive surgeries. Indeed, all the visual cues are application-independent. This is an advantage, but also a drawback: it means that the visual cues are probably not optimal for a specific application. To push performance further, we should include eye-related and tool-related visual cues into the image characterizations (Lalys et al., 2012). It may also be possible to push performance further by taking into account the temporal relationships between the composing tasks of a surgery, using an HMM for instance. However, as discussed above, a strict task ordering should not be assumed. Depth information should also be included whenever stereo-recording of the microscope output will be widely available.

As discussed in the introduction section, the proposed video monitoring system was primarily designed to communicate information to the surgeon in due time. In particular, it was designed to send recommendations to the

less experienced surgeons slightly before the beginning of each task. We have shown that it can detect and categorize surgical tasks approximately when they happen. So, when task $n-1$ is detected, we can provide recommendations for task n based on observations from tasks $1, ..., n-1$. Note that giving recommendations while the surgeon has already started performing task n would not be optimal: the decision on how to perform that task has already been made. Note that the proposed system is a case-based reasoning system: it relies on a dataset of surgical videos in which experts have temporally segmented the surgical tasks. If, in addition to segmenting the surgical tasks, experts were also asked to temporally segment surgical complications, then we may be able to go one step further: knowing what the surgeon is doing and what problem he or she is facing, recommendations could be refined. We believe that, one day, such an automatic video monitoring system may also be used to communicate with surgical devices directly. The idea would be, for instance, to preventively shut down a device in case of complication. Finally, note that the system may also be used offline, for video structuring. Video content structuring will allow efficient video browsing and therefore, efficient video archive browsing. Browsing video archives may be useful for retrospective clinical studies, skill assessment or training. It may be used to generate surgery reports at the end of each surgery (Lalys et al., 2012).

In conclusion, we have presented a system that can retrieve similar video subsequences in real-time. It was successfully applied to real-time recognition of surgical tasks in eye surgeries. And we have discussed its potential clinical applications.

References

André, B., Vercauteren, T., Buchner, A. M., Shahid, M. W., Wallace, M. B., Ayache, N., 2010. An image retrieval approach to setup difficulty levels in training systems for endomicroscopy diagnosis. In: Proc MICCAI'10. Vol. 13. pp. 480–487.

Arya, S., Mount, D. M., 1993. Approximate nearest neighbor queries in fixed dimensions. In: Proc ACM-SIAM SODA'93. pp. 271–280.

Blum, T., Feussner, H., Navab, N., 2010. Modeling and segmentation of surgical workflow from laparoscopic video. In: Proc. MICCAI'10. Vol. 13. pp. 400–407.

Bruno, E., Moenne-Loccoz, N., Marchand-Maillet, S., 2008. Design of multimodal dissimilarity spaces for retrieval of video documents. IEEE Trans Pattern Anal Mach Intell 30 (9), 1520–1533.

Cano, A. M., Gayá, F., Lamata, P., Sánchez-González, P., Gómez, E. J., 2008. Laparoscopic tool tracking method for augmented reality surgical applications. In: Proc LNCS'08. Vol. 5104. pp. 191–196.

Cao, Y., Liu, D., Tavanapong, W., Wong, J., Oh, J., de Groen, P., 2007. Computer-aided detection of diagnostic and therapeutic operations in colonoscopy videos. IEEE Trans Biomed Eng 54 (7), 1268–1279.

Castells, X., Comas, M., Castilla, M., Cots, F., Alarcón, S., 1998. Clinical outcomes and costs of cataract surgery performed by planned ECCE and phacoemulsification. Int Ophthalmol 22 (6), 363–367.

Dev, S., Mieler, W. F., Pulido, J. S., Mittra, R. A., 1999. Visual outcomes after pars plana vitrectomy for epiretinal membranes associated with pars planitis. Ophthalmology 106 (6), 1086–1090.

Douze, M., Jégou, H., Schmid, C., 2010. An image-based approach to video copy detection with spatio-temporal post-filtering. IEEE Trans Multimedia 12 (4), 257–266.

Duchenne, O., Laptev, I., Sivic, J., Bach, F., Ponce, J., 2009. Automatic annotation of human actions in video. In: Proc ICCV'09. pp. 1491–1498.

Dyana, A., Subramanian, M. P., Das, S., 2009. Combining features for shape and motion trajectory of video objects for efficient content based video retrieval. In: Proc ICAPR'09. pp. 113–116.

Gao, H. P., Yang, Z. Q., 2010. Content based video retrieval using spatiotemporal salient objects. In: Proc IPTC'10. pp. 689–692.

Giannarou, S., Yang, G.-Z., 2010. Content-based surgical workflow representation using probabilistic motion modeling. In: LNCS MIAR'10. Vol. 6326. pp. 314–323.

Gionis, A., Indyk, P., Motwani, R., 1999. Similarity search in high dimensions via hashing. In: Proc VLDB'99. pp. 518–529.

Hanley, J. A., McNeil, B. J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36.

Haro, B. B., Zappella, L., Vidal, R., 2012. Surgical gesture classification from video data. In: Proc. MICCAI'12. Vol. 15. pp. 34–41.

Harris, Z., 1954. Distributional structure. Word 10 (23), 146–62.

Hoi, S. C. H., Lyu, M. R., 2007. A multimodal and multilevel ranking framework for content-based video retrieval. In: Proc ICASSP'07. Vol. 4. pp. 1225–1228.

Hu, W., Xie, D., Fu, Z., Zeng, W., Maybank, S., 2007. Semantic-based surveillance video retrieval. IEEE Trans Image Process 16 (4), 1168–1181.

Huang, M., Yang, W., Yu, M., Lu, Z., Feng, Q., Chen, W., 2012. Retrieval of brain tumors with region-specific bag-of-visual-words representations in contrast-enhanced MRI images. Comput Math Methods Med 2012, 280538.

Juan, K., Cuiying, H., 2010. Content-based video retrieval system research. In: Proc ICCSIT'10. Vol. 4. pp. 701–704.

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: Proc. MICCAI'11. Vol. 14. pp. 331–338.

Lalys, F., Riffaud, L., Bouget, D., Jannin, P., 2012. A framework for the recognition of high-level surgical tasks from video images for cataract surgeries. IEEE Trans Biomed Eng 59 (4), 966–76.

Laptev, I., 2005. On space-time interest points. Int J Comput Vis 64 (2-3), 107–123.

Lawson, C. L., Hanson, B. J., 1974. Solving Least Squares Problems. Prentice-Hall (Englewood Cliffs, NJ).

Lucas, B. D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: Proc IUW'81. pp. 121–130.

Marszałek, M., Laptev, I., Schmid, C., 2009. Actions in context. In: Proc IEEE CVPR'09. pp. 2929–2936.

Naturel, X., Gros, P., 2008. Detecting repeats for video structuring. Multimed Tool Appl 38 (2), 233–252.

O'Neill, M., Ryan, C., 2001. Grammatical evolution. IEEE Trans Evol Comput 5 (4).

Padoy, N., Blum, T., Ahmadi, S., Feussner, H., Berger, M., Navab, N., 2012. Statistical modeling and recognition of surgical workflow. Med Image Anal 16 (3), 632–641.

Patel, B. V., Deorankar, A. V., Meshram, B. B., 2010. Content based video retrieval using entropy, edge detection, black and white color features. In: Proc ICCET'10. Vol. 6. pp. 272–276.

Pearson, K., 1901. On lines and planes of closest fit to systems of points in space. Philos Mag 2 (6), 559–572.

Piriou, G., Bouthemy, P., Yao, J.-F., 2006. Recognition of dynamic video contents with global probabilistic models of visual motion. IEEE Trans Image Process 15 (11), 3417–3430.

Quellec, G., Lamard, M., Cazuguel, G., Cochener, B., Roux, C., 2012. Fast wavelet-based image characterization for highly adaptive image retrieval. IEEE Trans Image Process 21 (4), 1613–1623.

Reiley, C. E., Hager, G. D., 2009. Task versus subtask surgical skill evaluation of robotic minimally invasive surgery. In: Proc. MICCAI'09. Vol. 12. pp. 435–442.

Ricci, F., Rokach, L., Shapira, B., (Eds.), P. B. K., 2011. Recommender Systems Handbook. Vol. XXIX.

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26 (1), 43–49.

Schapire, R., 1990. Strength of weak learnability. Mach Learn 5, 197–227.

Smeaton, A. F., Over, P., Kraaij, W., 2006. Evaluation campaigns and TRECVid. In: Proc ACM Int Workshop MIR'06. pp. 321–330.

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22 (12), 1349–1380.

Syeda-Mahmood, T., Ponceleon, D., Yang, J., 2005. Validating cardiac echo diagnosis through video similarity. In: Proc ACM Multimedia. pp. 527–530.

Tamaki, T., Yoshimuta, J., et al., M. K., 2013. Computer-aided colorectal tumor classification in NBI endoscopy using local features. Med Image Anal 17 (1), 78–100.

Tao, L., Elhamifar, E., Khudanpur, S., Hager, G. D., Vidal, R., 2012. Sparse hidden markov models for surgical gesture classification and skill evaluation. In: Proc IPCAI'12. Vol. 7330. pp. 167–177.

von Mises, R., 1964. Mathematical Theory of Probability and Statistics. Academic Press, New York.

Xu, D., Chang, S. F., 2008. Video event recognition using kernel methods with multilevel temporal alignment. IEEE Trans Pattern Anal Mach Intell 30 (11), 1985–1997.

9. Vitae

Gwénolé Quellec was born in Saint-Renan, France, on November 29, 1982. He received the engineering degree in computer science and applied mathematics from ISIMA, Clermont-Ferrand, France, in 2005, the M.S. degree in image processing from the University of Clermont-Ferrand II, in 2005, and the Ph.D. degree in signal processing from TELECOM Bretagne, Brest, France, in 2008. He is a Research Associate at the LaTIM Inserm Research Unit 1101, Brest, France. His research interests include retinal image processing, content-based image and video retrieval, and information fusion for medical applications.

Katia Charrière was born in Sallanches, France, on December 13, 1986. She received the engineering degree in engineering and life sciences from TPS (previously named ENSPS), Strasbourg, France, in 2011, and the M.S degree in imaging, robotics and biomedical engineering from the

University of Strasbourg, France in 2011. She is currently a 1st year Ph.D. Student at the LaTIM Inserm Research Unit 1101 and Telecom Bretagne, Brest, France. Her research interests include content-based video retrieval for medical applications.

Mathieu Lamard was born in Bordeaux, France, on May 18, 1968. He received the M.S. degree in applied mathematics from the University of Bordeaux, France, in 1995, and the Ph.D. degree in signal processing and telecommunication from the University of Rennes, France, in 1999. He joined the LaTIM Inserm Research Unit 1101 in 2000, where he is currently a Research Associate. His research interests include image processing, 3-D reconstruction, content-based image retrieval, and information fusion for medical applications.

Zakarya Droueche was born in Tlemcen, Algeria, on July 13, 1986, He received the engineering degree in computer science from the University of Tlemcen, Algeria, in 2008, the M.S. degree in signal and image processing from the University of Lille 1, France, in 2009, and the Ph.D. degree in signal processing from TELECOM Bretagne, Brest, France, in 2012. He is a research engineer at ECA Robotics, Paris, France. His research interests include image processing, 3-D reconstruction, and Simultaneous Localization And Mapping (SLAM).

Christian Roux received the Agregation degree in physics from the Ecole Normale Superieure, Cachan, France, in 1978, and the Ph.D. degree from the Institut National Polytechnique, Grenoble, France, in 1980. He joined TELECOM Bretagne, Brest, France, in 1982. He is the author of more than 100 papers and of four book chapters, has edited three books, and holds two patents. His current research interests include medical information processing, spatial and functional information modeling, and analysis in medical images. Dr. Roux is a member of the Editorial board of the IEEE Transactions on Information Technology and of the Proceedings of the IEEE.

Béatrice Cochener is a Professor and Head of the University Eye Clinic, Brest, France. Together with J. Colin, she developed a very active anterior segment surgery practice. She is currently involved in imaging research, clinical evaluation, and anterior segment surgery teaching. Vice president of the SAFIR, the French implant and refractive surgery

society, president of the French Academy of Ophthalmology and editorial board member of the Journal Français d'Ophtalmologie, she is a specialist of refractive techniques in vision correction. She participated in three books on surgical techniques and has published more than 30 peer reviewed journal articles.

Guy Cazuguel received the engineering degree from the Ecole Nationale de l'Aviation Civile, in 1975, the M.S. degree in advanced automatics, in 1976, and the Ph.D. degree in signal processing and telecommunications from the University of Rennes I, France, in 1994. He is currently a Professor in the Image and Information Processing Department, TELECOM Bretagne. His research interests include image analysis and content based image retrieval in medical applications, within the LaTIM Inserm Research Unit 1101.

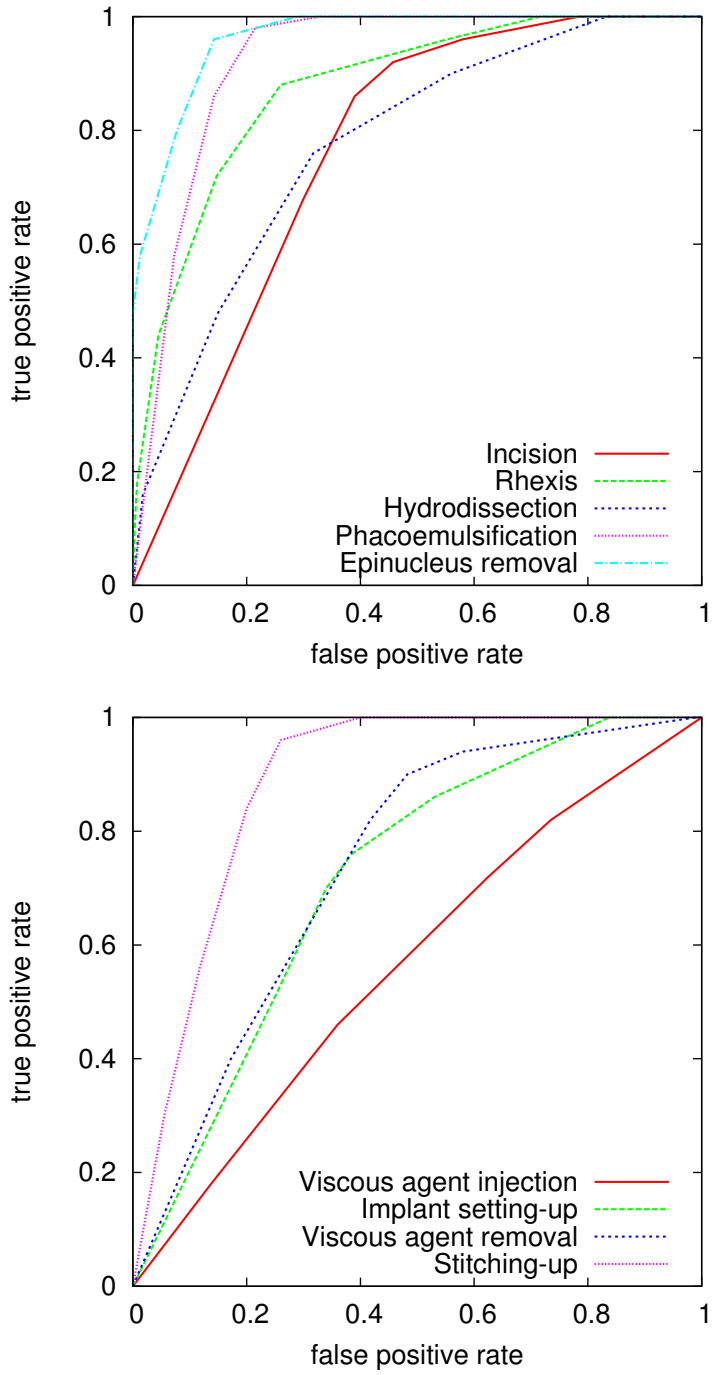


Figure 7: ROC curves for the CSD dataset (§6.3)

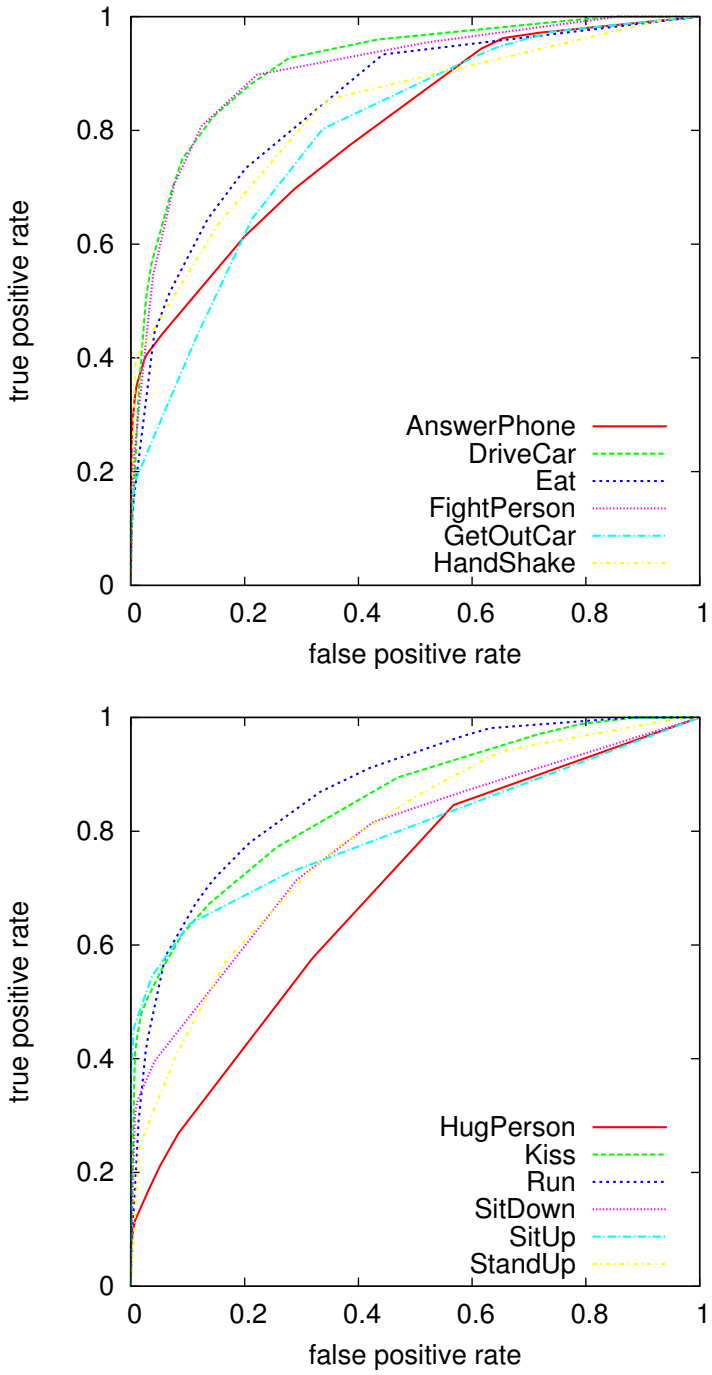


Figure 8: ROC curves for the MCD dataset (§6.4)