Recherche en temps réel de séquences vidéo similaires par le contenu

Gwénolé Quellec [1], **Mathieu Lamard** [2, 1], **Guy Cazuguel** [3, 1], **Zakarya Droueche** [3, 1], **Béatrice Cochener** [2, 1, 4], **Christian Roux** [3, 1]

1. *Laboratoire de Traitement de l'Information Médicale*
   UMR 1101, Inserm
   Bâtiment 2bis (I3S), CHU Morvan - 5 avenue Foch, F-29609 Brest cedex, France
   gwenole.quellec@inserm.fr

2. *Université de Bretagne Occidentale*
   3 rue des Archives, CS 93837, F-29238 Brest cedex 3, France
   mathieu.lamard@univ-brest.fr

3. *Département Image et Traitement de l'Information*
   TELECOM Bretagne
   Technopôle Brest-Iroise, CS 83818, F-29285 Brest cedex, France
   {guy.cazuguel, mohammed.droueche, christian.roux}@telecom-bretagne.eu

4. *Service d'Ophtalmologie*
   CHU Morvan
   Bâtiment 4, 2 avenue Foch, F-29609 Brest cedex, France
   Beatrice.Cochener@ophtalmologie-chu29.fr

RÉSUMÉ. *Nous proposons dans cet article une méthode originale pour rechercher, dans des séquences vidéo, des sous-séquences similaires. En introduisant de la flexibilité temporelle dans la caractérisation des sous-séquences, cette méthode permet d'éviter l'utilisation de mesures de distance flexibles (telles que le Dynamic Time Warping) qui ont l'inconvénient d'être lentes. La méthode proposée permet donc de rechercher, en temps réel, des sous-séquences vidéo similaires parmi plusieurs centaines de milliers d'exemples. La méthode proposée est adaptative ; un algorithme d'apprentissage rapide est présenté. Les performances ont été évaluées avec succès sur un ensemble de 1 707 clips vidéo (> 800 000 sous-séquences). A terme, notre objectif est de proposer un système de génération d'alertes et/ou de préconisations, en temps réel, dans le cadre de l'aide à la chirurgie sous contrôle vidéo.*

ABSTRACT. *A novel Content-Based Video Retrieval (CBVR) framework is presented in this paper: its purpose is to find similar video sub-sequences in videos. By introducing temporal flexibility in the description of video sub-sequences, this framework makes the use of flexible, but slow,*

distance measures (such as Dynamic Time Warping) optional. As a consequence, real-time retrieval of similar video sub-sequences, among hundreds of thousands of examples, is now possible. The proposed method is adaptive; a fast training procedure is presented. Performances have been successfully assessed on a dataset of 1,707 video clips (> 800,000 sub-sequences). Ultimately, we plan to design a real-time alert (and/or recommendation) generation system for computed-aided video-guided surgery.

MOTS-CLÉS : *recherche de vidéos par le contenu, traitement en temps réel, ondelettes.*

KEYWORDS: *content-based video retrieval, real-time processing, wavelets.*

DOI:10.3166/TS.29.83-100

Extended abstract

Context

Content-Based Video Retrieval (CBVR) is an increasingly active research field. The goal of CBVR systems is to select, from a digital archive, video sequences that resemble a query video. Similarity measurements between videos rely on motion, shape, texture or color analysis. Initially popularized in broadcasting (Naturel, Gros, 2008) and video surveillance (Hu *et al.*, 2007), CBVR is now developing in other domains. For instance, its use for medical training is considered (André *et al.*, 2010). Typical CBVR systems allow users to submit a query video file and return similar video files on output (Xu, Chang, 2008). A more ambitious scenario is studied in this paper: we propose to analyze the video stream (captured by a digital camera, for instance) in real-time and to continually search for similar video subsequences. The goal is to constantly generate warnings and recommendations. In video surveillance, it might be used to detect suspicious behaviors. In medicine, it might be used to predict complications during a surgery. When we work with video subsequences, and not simply entire videos, the number of items to be compared with the query explodes. Due to the real-time constraint, the use of a very fast similarity measures, to compare subsequences, is therefore mandatory. In particular, the use of temporally flexible, but slow, distance measures, such as *Dynamic Time Warping* (Sakoe, Chiba (1978) ; Xu, Chang (2008)), is prohibited. However, temporal flexibility is required to cope with speed differences among surgeons. An alternative solution is proposed in this paper: temporal flexibility is directly introduced in the way video subsequences are characterized, while meeting the real-time constraint.

Method

Let $\mathcal{A}$ be a target type of actions that the user would like to search in video subsequences (*walk*, *run*, *drive*, etc.) and let $\mathcal{D}$ be a set of video sequences. For training and evaluation purposes, we assume that the user indicated which video sequences $V \in \mathcal{D}$ contain actions of type $\mathcal{A}$, without specifying when exactly these actions occured. The

user is simply expected to assign a binary label $\delta(\mathcal{A},V)$ to each video sequence $V \in \mathcal{D}$: $\delta(\mathcal{A},V) = true$ if V contains at least one action of type $\mathcal{A}$, $\delta(\mathcal{A},V) = false$ otherwise. In other words, training is weakly supervised.

The goal of the proposed method is to analyze video sequences in real-time, in order to detect actions of type $\mathcal{A}$. Let $V = \{V_1, V_2, ..., V_{n_V}\}$ be a video sequence, consisting of n_V images, that should be analyzed. At each time instant i:

1. a *video subsequence* $V_{]i-n;i]}$, consisting of the current image V_i plus the $n-1$ preceding images, is characterized,
2. the nearest neighbors of $V_{]i-n;i]}$, in a reference dataset, are selected.

In order to select semantically-relevant video subsequences, the parameters of the similarity measure are adapted to each target action. These parameters are automatically tuned, during the training phase, in order to maximize the relevance of selected video subsequences when query video subsequences are in a training dataset. To tune these parameters, the trick is to convert the similarity metric between video subsequences into a similarity metric between entire video sequences. This transformed similarity metric can more easily be trained using the $\delta(\mathcal{A},V)$ labels.

Experiment

The proposed framework was applied to the HOLLYWOOD2 [1] human action dataset, a publicly-available dataset of video clips extracted from Hollywood movies. 1707 movie clips were used in this experiment (average duration: 20 seconds). The presence of twelve action types was annotated in each video clip. About 800,000 video subsequences were extracted from those video clips.

The proposed method was compared to Sivic, Zisserman (2003)'s *Video Google*. In this experiment, the goal was simply to detect the presence or the absence of each action type in each video. The criterion retained to evaluate both methods in the HOLLYWOOD2 dataset was the area under the receiver operating characteristic curve (AUC).

Results and conclusion

With the proposed method, the AUC ranged from 0.667 (HugPerson action) to 0.880 (FightPerson action). With *Video Google*, the AUC ranged from 0.533 (HandShake action) to 0.853 (Kiss action). For eleven actions out of twelve, the proposed method outperformed *Video Google*. The total processing time to process a query video subsequence was 37 milliseconds, while the frame period was 40 milliseconds. Therefore, real-time retrieval of similar video sub-sequences, among hundreds of thousands of examples, is now possible.

1. http://www.irisa.fr/vista/actions/hollywood2/

1. Introduction

La recherche automatique de séquences vidéo similaires par le contenu (*Content-Based Video Retrieval* — CBVR) est un champ de recherche de plus en plus actif. L'objectif des systèmes CBVR est de sélectionner, au sein d'une archive numérique, des séquences vidéo similaires à une séquence placée en requête. La notion de similitude entre séquences vidéo se fonde sur une analyse du mouvement, des formes, de la texture ou encore de la couleur. Popularisée à l'origine dans le domaine de l'audio-visuel (Naturel, Gros, 2008) et celui de la vidéosurveillance (Hu *et al.*, 2007), la CBVR commence à se développer dans d'autres domaines. Son utilisation pour la formation des jeunes médecins est ainsi envisagée dans le domaine médical (André *et al.*, 2010). Typiquement, les systèmes CBVR permettent à un utilisateur de soumettre un fichier vidéo en requête, avec en retour la présentation de fichiers vidéo similaires (Xu, Chang, 2008). Ces systèmes généralisent ainsi les systèmes de recherche d'images par le contenu (Quellec *et al.*, 2010). Une utilisation plus ambitieuse de la CBVR est étudiée dans cet article : nous proposons d'analyser, en temps réel, le flux vidéo (issu par exemple d'une caméra) et de rechercher, à chaque instant, des sous-séquences vidéo similaires. L'objectif est de pouvoir générer, à chaque instant, des alertes ou des recommandations. Dans le domaine de la vidéosurveillance, l'intérêt peut être de détecter des comportements suspects. Dans le domaine médical, il peut être de prédire des complications au cours d'une chirurgie. Lorsque l'on s'intéresse à des sous-séquences vidéo, et non plus à des séquences entières, le nombre d'objets à comparer à la requête explose. De par la contrainte de temps imposée, il est alors impératif d'avoir une mesure de distance très rapide pour comparer des sous-séquences. Ainsi, les méthodes flexibles de comparaison de données temporelles, telles que le *Dynamic Time Warping*, sont à proscrire (Sakoe, Chiba (1978) ; Xu, Chang (2008)). Il est cependant souhaitable de prendre en compte les variations de durée entre deux actions à comparer : une solution est proposée dans cet article pour introduire de la flexibilité temporelle dans la manière de caractériser chaque sous-séquence, tout en respectant la contrainte de temps réel.

2. Etat de l'art

Il existe une grande variété de systèmes CBVR dans la littérature. Ces systèmes se différencient tout d'abord par la nature des objets placés en requête. Dans le système proposé par Patel *et al.* (2010), une image est placée en requête et il s'agit de trouver la ou les vidéo(s) contenant cette image dans un ensemble de référence : nous nous ramenons donc à un problème classique de recherche d'images. Dans les systèmes proposés par Naturel, Gros (2008) et Dyana *et al.* (2009), chaque plan d'un flux vidéo est placé en requête et il s'agit de rechercher d'autres occurrences de ce même plan (Naturel, Gros, 2008), ou des plans similaires (Dyana *et al.*, 2009), dans un ensemble de vidéos de référence. Dans le système proposé par André *et al.* (2010), une séquence est placée en requête et il s'agit de sélectionner les séquences globalement les plus proches dans un ensemble de vidéos de référence.

Les systèmes CBVR se différencient également par la manière dont les séquences ou sous-séquences vidéo sont caractérisées. Plusieurs d'entre eux s'appuient principalement sur la détection et la caractérisation d'images-clés (Juan, Cuiying (2010) ; Patel *et al.* (2010)). D'autres systèmes reposent directement sur la caractérisation des séquences ou des sous-séquences (Dyana *et al.* (2009) ; Gao, Yang (2010)). Ainsi, dans le système proposé par Dyana *et al.* (2009), les plans vidéo sont caractérisés par des paramètres de forme et par l'évolution des vecteurs de mouvement au cours du plan. Dans le système proposé par Gao, Yang (2010), des *objets spatio-temporels* (c'est-à-dire des objets en mouvement) saillants sont détectés au sein d'un flux vidéo, puis des caractéristiques sont extraites de chacun de ces objets. Enfin, les vidéos les plus proches sont sélectionnées à l'aide de la distance EMD (*Earth-Mover's Distance*) entre les caractéristiques de ces objets. Notons que, dans le domaine de l'audio-visuel, certains auteurs (Hoi, Lyu (2007) ; Bruno *et al.* (2008)) proposent d'intégrer des informations multimodales (informations visuelles, audio et textuelles) dans le système de recherche.

Ensuite, les systèmes CBVR se différencient par le besoin de flexibilité de la mesure de distance entre séquences ou sous-séquences vidéo à comparer. Plusieurs méthodes ont été proposées dans le domaine de l'audio-visuel pour rechercher des sous-séquences vidéo identiques (à de faibles déformations près). Ainsi, une méthode a été proposée par Douze *et al.* (2010) pour détecter des copies de vidéos protégées par droit d'auteur. Cette méthode consiste à comparer individuellement les images des séquences vidéo puis à contrôler leur cohérence spatio-temporelle *a posteriori*. Une autre méthode a été proposée par Naturel, Gros (2008) dans le but de détecter, dans un flux de télévision, des plans vidéo qui se répètent, dans un but de structuration du flux vidéo. Chaque image est caractérisée à l'aide de sa transformée en cosinus discrète ; la signature ainsi extraite sert de clé à une table de hachage pour trouver rapidement les plans similaires. En revanche, dans la majorité des systèmes CBVR, nous recherchons des séquences ou sous-séquences vidéo sémantiquement proches, mais dont le contenu visuel peut varier considérablement d'une séquence à l'autre (Juan, Cuiying (2010) ; Xu, Chang (2008) ; André *et al.* (2010)).

Dans cet article, nous proposons un système CBVR capable de détecter des actions d'intérêt dans un flux vidéo. Dans cette application, les objets placés en requête sont de courtes sous-séquences extraites du flux et la mesure de distance a un grand besoin de flexibilité. Quelques méthodes ont été proposées dans la littérature pour résoudre ce problème. Tout d'abord, une méthode a été proposée par Piriou *et al.* (2006) pour catégoriser une action à partir d'informations de mouvement. Un modèle du mouvement est construit pour chaque type d'action. Puis, une fois éliminé le mouvement dû au déplacement de la caméra, l'action courante est classifiée par maximum *a posteriori*, en analysant le mouvement résiduel. Ensuite, une méthode faiblement supervisée a été proposée par Duchenne *et al.* (2009) pour détecter des actions cibles dans des séquences cinématographiques. Dans cette méthode, une représentation par sac de mots est utilisée pour caractériser des sous-séquences, puis les sous-séquences contenant l'action cible sont détectées à l'aide d'un partitionnement de données discriminant à noyaux. Une autre méthode a été proposée par Xu, Chang (2008) pour détecter

des événements cibles dans un flux d'informations télévisées. Là encore, une représentation par sac de mots est utilisée pour caractériser des sous-séquences de différentes tailles. Pour comparer deux séquences, Xu, Chang (2008) utilisent une version *multi-taille* de la distance EMD entre les caractéristiques des sous-séquences. Contrairement aux méthodes précédentes, la méthode proposée a l'avantage de fonctionner en temps réel, même lorsque de très grands ensembles de référence sont utilisés.

3. Vue d'ensemble de la méthode

Soit $\mathcal{A}$ un type d'actions que l'utilisateur souhaite rechercher dans les séquences vidéo (*marcher*, *courir*, *conduire*, etc.) et soit $\mathcal{D}$ un ensemble de séquences vidéo. Pour l'entraînement et l'évaluation de la méthode, nous supposons que l'utilisateur a indiqué quelles séquences vidéo $V \in \mathcal{D}$ contiennent des actions de type $\mathcal{A}$, sans préciser à quel moment ces actions apparaissent dans les séquences. L'utilisateur doit simplement affecter un label binaire $\delta(\mathcal{A},V)$ à chaque séquence vidéo $V \in \mathcal{D}$: $\delta(\mathcal{A},V) = vrai$ si V contient au moins une action de type $\mathcal{A}$, $\delta(\mathcal{A},V) = faux$ sinon. Autrement dit, l'apprentissage est faiblement supervisé. $\mathcal{D}$ est divisé en un sous-ensemble d'apprentissage $\mathcal{D}_{train}$ et un sous-ensemble de test $\mathcal{D}_{test}$. Après l'apprentissage, $\mathcal{D}_{train}$ joue également le rôle d'ensemble de référence.

Le but de la méthode proposée est d'analyser des séquences vidéo en temps réel, afin d'identifier des actions de type $\mathcal{A}$. Nous supposons que chaque séquence vidéo contient un unique plan : il peut s'agir d'une vidéo chirurgicale, d'une vidéo de surveillance, d'un plan extrait d'un film, etc. Soit $V = \{V_1, V_2, ..., V_{n_V}\}$ une séquence vidéo, constituée de n_V images, à analyser. A chaque instant i :

1. une *sous-séquence vidéo* $V_{]i-n;i]}$, composée de l'image courante V_i plus les $n-1$ images précédentes, est caractérisée (voir section 4),

2. les plus proches voisins de $V_{]i-n;i]}$, dans l'ensemble de référence $\mathcal{D}_{train}$, sont sélectionnés (voir section 5).

Afin de sélectionner des sous-séquences vidéo sémantiquement pertinentes, les paramètres de comparaison des sous-séquences vidéo doivent être adaptés à chaque action. Ils sont ajustés automatiquement, pendant la phase d'apprentissage, afin de maximiser la pertinence des sous-séquences sélectionnées lorsque des sous-séquences de l'ensemble d'apprentissage $\mathcal{D}_{train}$ sont placées en requête (voir section 5.3). Puisque les utilisateurs n'interprètent que les séquences vidéo dans leur ensemble (*via* les labels binaires $\delta(\mathcal{A},V)$), la pertinence de la méthode est évaluée au niveau des séquences vidéo, et non au niveau des sous-séquences (voir section 6.2).

4. Caractérisation d'une sous-séquence vidéo

Soit $V \in \mathcal{D}$ une séquence vidéo constituée de n_V images : $V = \{V_1, V_2, ..., V_{n_V}\}$. $n_V - n + 1$ sous-séquences vidéo de taille n sont caractérisées dans V : $V_{]0;n]}$, $V_{]1;n+1]}$,

..., $V_{]n_V-n;n_V]}$. La sous-séquence $V_{]i-n;i]}$ est constituée de l'image V_i plus les $n-1$ images précédentes :

$$V_{]i-n;i]} = \{V_{i-n+1}, V_{i-n+2}, ..., V_{i-1}, V_i\} \quad (1)$$

Notons que deux sous-séquences consécutives $V_{]i-n;i]}$ et $V_{]i-n+1;i+1]}$ se chevauchent. En effet, si nous partitionnions chaque séquence vidéo en sous-séquences ne se chevauchant pas, alors de nombreuses actions se retrouveraient à cheval sur deux sous-séquences consécutives et ne pourraient donc pas être détectées.

4.1. Structure d'une sous-séquence vidéo

Afin d'introduire de la flexibilité temporelle dans la caractérisation d'une sous-séquence, nous organisons les images qui la composent en *m intervalles élémentaires d'images*. Chaque intervalle élémentaire d'images est défini par un point de départ t_b et une durée Δt_b, $b = 1..m$:

$$V_{]i-n;i]} = \bigcup_{b=1..m} V_{]i-t_b-\Delta t_b;i-t_b]} \quad (2)$$

Deux sous-séquences vidéo sont considérées comme semblables si leurs premiers intervalles élémentaires d'images sont semblables *et* leurs deuxièmes intervalles élémentaires d'images sont semblables, etc. Notons que deux intervalles élémentaires d'images peuvent se chevaucher (voir figure 1c). La longueur des sous-séquences vidéo est donnée par :

$$n = \max_{b=1..m} \{t_b + \Delta t_b\} \quad (3)$$

Considérons l'exemple suivant : $\mathcal{A}$=*sauter*. Un saut peut grossièrement être décomposé en deux phases : *monter* et *descendre*. Il peut donc être judicieux d'utiliser $m = 2$ intervalles élémentaires d'images. Le système sera alors adapté pour caractériser de manière optimale les actions de type *monter* dans l'intervalle élémentaire d'images $(t_1, \Delta t_1)$ et les actions de type *descendre* dans l'intervalle élémentaire d'images $(t_2, \Delta t_2)$. Parce que les actions de type *monter* sont de durée variable et qu'elles sont immédiatement suivies par une action de type *descendre*, il est judicieux de faire se chevaucher $(t_1, \Delta t_1)$ et $(t_2, \Delta t_2)$.

Afin de comparer des sous-séquences vidéo de manière flexible, la notion d'ordonnancement temporel est ignorée au sein de chaque intervalle élémentaire d'images. En particulier, la caractérisation d'une sous-séquence $V_{]i-n;i]}$ sera identique qu'un événement important survienne dans l'image V_k *ou* dans l'image V_l, pourvu que pour tout b :

- $i - t_b - \Delta t_b \geq k, l$,
- ou $i - t_b - \Delta t_b < k, l \leq i - t_b$,
- ou $k, l > i - t_b$.

Chaque sous-séquence vidéo $V_{]i-n;i]}$ est caractérisée comme suit :

1. l'image V_i est caractérisée (section 4.3),
2. la caractérisation de V_i est stockée dans une file d'attente de taille n,
3. les n dernières caractérisations d'images dans la file d'attente sont combinées pour caractériser chaque intervalle élémentaire d'images (section 4.4) puis $V_{]i-n;i]}$ dans son ensemble (section 4.5).

4.2. Exemples d'organisation des sous-séquences en intervalles élémentaires d'images

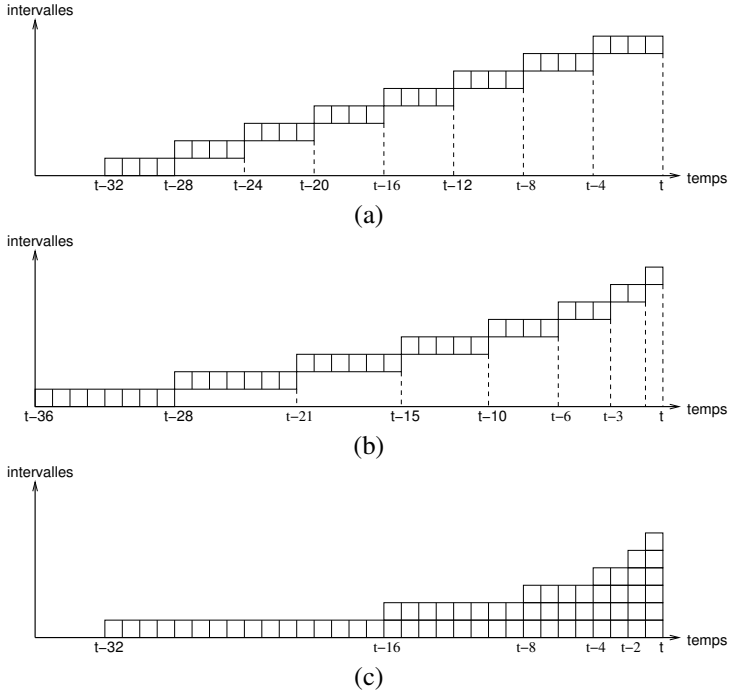

Figure 1. Exemples d'organisation des sous-séquences en intervalles élémentaires d'images

Dans les exemples (a) et (c) (resp. (b)) de la figure 1, chaque sous-séquence est composée de $n = 32$ (resp. $n = 36$) images. Dans les exemples (a) et (b) (resp. (c)), chaque sous-séquence est composée de $m = 8$ (resp. $m = 6$) intervalles élémentaires d'images. Dans l'exemple (b), les huit images comprises entre l'instant $t - 36$ et l'instant $t - 28$ appartiennent à un même intervalle : ainsi, qu'un événement survienne à l'instant $t - 34$ ou à l'instant $t - 29$ n'aura pas d'influence sur la caractérisation de la sous-séquence. Cela induit une certaine flexibilité temporelle dans la caractérisation des sous-séquences. Permettre aux intervalles de se chevaucher (voir exemple (c))

augmente encore la flexibilité temporelle. Notez que plus l'on s'éloigne de l'instant t, plus les décalages temporels entre actions à comparer sont susceptibles d'augmenter (les différences entre les deux actions se cumulant au cours du temps) : il semble donc judicieux d'augmenter progressivement la taille des intervalles élémentaires d'images (voir exemples (b) et (c)).

4.3. Caractérisation d'une image dans la sous-séquence

Pour décrire l'image V_i, nous extrayons différentes caractéristiques de V_i et du flot optique entre V_{i-1} et V_i. Pour calculer le flot optique, nous détectons d'abord des points d'intérêts dans V_{i-1}. Les points d'intérêts sont choisis parmi les pixels p de l'image, en fonction de la plus petite valeur propre de la matrice M_p ci-dessous :

$$M_p = \begin{pmatrix} \sum_{(x,y)\in\mathcal{V}_p} \left(\frac{\partial V_{i-1}}{\partial x}(x,y)\right)^2 & \sum_{(x,y)\in\mathcal{V}_p} \frac{\partial V_{i-1}}{\partial x}(x,y)\cdot\frac{\partial V_{i-1}}{\partial y}(x,y) \\ \sum_{(x,y)\in\mathcal{V}_p} \frac{\partial V_{i-1}}{\partial y}(x,y)\cdot\frac{\partial V_{i-1}}{\partial x}(x,y) & \sum_{(x,y)\in\mathcal{V}_p} \left(\frac{\partial V_{i-1}}{\partial y}(x,y)\right)^2 \end{pmatrix} \quad (4)$$

où $\mathcal{V}_p$ est un voisinage du pixel p. Ensuite, le flot optique entre V_{i-1} et V_i est calculé, en chaque point d'intérêt, par la méthode itérative de Lucas-Kanade (Lucas, Kanade, 1981). La librairie OpenCV [2] a été utilisée pour calculer les points d'intérêt et le flot optique. Les caractéristiques que nous avons choisi d'utiliser dans cette étude (des caractéristiques de texture, de couleur et de mouvement) sont décrites en annexe. Notez que ces caractéristiques peuvent être remplacées par n'importe quelles caractéristiques calculables en quelques millisecondes. Soit $f(V_i)$ le vecteur de paramètres regroupant toutes les caractéristiques extraites de V_i et du flot optique entre V_{i-1} et V_i.

4.4. Caractérisation d'un intervalle élémentaire d'images

Soit $f(V_{]i-t_b-\Delta t_b;i-t_b]})$ le vecteur de paramètres caractérisant l'intervalle élémentaire d'images $V_{]i-t_b-\Delta t_b;i-t_b]}$. Pour garantir le succès de la méthode, $f(V_{]i-t_b-\Delta t_b;i-t_b]})$ doit idéalement être indépendant de la position des images dans l'intervalle. Un moyen simple et rapide d'approcher cet objectif est de définir $f(V_{]i-t_b-\Delta t_b;i-t_b]})$ comme la moyenne des vecteurs de paramètres $f(V_j)$, $j \in]i-t_b-\Delta t_b; i-t_b]$:

$$f(V_{]i-t_b-\Delta t_b;i-t_b]}) = \frac{1}{\Delta t_b} \sum_{j=i-t_b-\Delta t_b+1}^{i-t_b} f(V_j) \quad (5)$$

4.5. Caractérisation de la sous-séquence

Dans un premier temps, la sous-séquence $V_{]i-n;i]}$ est quant à elle caractérisée par la concaténation des vecteurs de paramètres $f(V_{]i-t_b-\Delta t_b;i-t_b]})$, $b = 1..m$: soit $\bar{f}(V_{]i-n;i]})$

2. http://opencv.willowgarage.com/wiki/

ce vecteur de paramètres. Remarquons que les différents intervalles élémentaires d'images au sein d'une même sous-séquence ont de fortes chances d'être corrélés. Il en est de même pour les vecteurs de paramètres $f(V_{]i-t_b-\Delta t_b;i-t_b]})$. Afin d'obtenir un vecteur de paramètres plus compact, et dont les composantes soient moins corrélées, nous opérons une analyse en composantes principales de l'ensemble des vecteurs $\bar{f}(V_{]i-n;i]})$ au sein de l'ensemble d'apprentissage $\mathcal{D}_{train}$ (Pearson, 1901). Puis nous remplaçons chaque vecteur $\bar{f}(V_{]i-n;i]})$ par sa projection $f(V_{]i-n;i]})$ sur les C composantes principales [3].

5. Recherche de sous-séquences vidéo similaires

5.1. Mesure de distance entre caractérisations permettant une recherche rapide

L'avantage de manipuler des caractérisations $f(V_{]i-n;i]})$ de taille fixe est qu'il existe des algorithmes de recherche très rapide, tels que *k-d tree* (Arya, Mount, 1993) ou *Locality-Sensitive Hashing* (Gionis *et al.*, 1999). ANN [4], une approximation de l'algorithme *k-d tree*, a été utilisée dans cette étude. Le vecteur $f(V_{]i-n;i]})$ ne contenant que quelques dizaines de composantes (nombre dépendant de l'organisation en intervalles élémentaires choisie), un tel algorithme est approprié. Dans cet algorithme, les vecteurs de paramètres sont comparés à l'aide du carré de la distance euclidienne.

Afin de combler le fossé sémantique entre les descripteurs numériques de bas niveau et le concept haut-niveau de similitude sémantique, nous pondérons chaque composante $f_c(V_{]i-n;i]})$ du vecteur de paramètres $f(V_{]i-n;i]})$ par un réel λ_c, $c = 1..C$. Cela revient à pondérer le $c^{\text{ème}}$ terme de la mesure de distance par λ_c^2. Puisque les séquences vidéo manipulées n'ont été interprétées sémantiquement que dans leur intégralité, il n'est pas possible de superviser directement l'apprentissage des poids au niveau des sous-séquences. Pour y remédier, nous définissons une distance numérique $DN(U,V)$ entre deux séquences vidéo complètes U et V. Nous faisons intervenir les poids λ_c dans cette distance numérique $DN(U,V)$. Puis nous ajustons les poids pour que $DN(U,V)$ corresponde le mieux possible à la distance sémantique $DS(U,V)$ définie comme suit :

- $DS(U,V) = 0$ si U et V appartiennent à la même classe
- $DS(U,V) = 1$ sinon.

L'apprentissage des poids est présenté en section 5.3.

3. C est choisi de telle sorte que 90 % de l'énergie soit conservée.
4. http://www.cs.umd.edu/ mount/ANN/

5.2. Définition d'une distance entre séquences vidéo complètes

Pour chaque composante $c = 1..C$ des vecteurs de paramètres, nous définissons une distance numérique partielle $DN_c(U,V)$ entre les séquences vidéo U et V. Soit $\mathcal{F}_c(V)$ la $c^{\text{ème}}$ composante de l'ensemble des caractérisations extraites d'une séquence vidéo V :

$$\mathcal{F}_c(V) = \left\{ f_c(V_{]i-n;i]}), i = n..n_V \right\} \quad (6)$$

$DN_c(U,V)$ est définie comme l'écart maximum entre la fonction de répartition de $\mathcal{F}_c(U)$ et celle de $\mathcal{F}_c(V)$, c'est-à-dire la distance de Kolmogorov-Smirnov entre $\mathcal{F}_c(U)$ et $\mathcal{F}_c(V)$ (Mises, 1964).

5.3. Pondération des composantes de la caractérisation

Soit $B = |\mathcal{D}_{train}|$ le cardinal de l'ensemble d'apprentissage. Pour chaque couple de séquences $(U,V) \in \mathcal{D}^2_{train}$ (au nombre de $\frac{B(B-1)}{2}$), nous calculons une distance sémantique, ainsi que C distances numériques partielles, notées $DN_c(U,V)$ (section 5.2). Les distances sémantiques sont regroupées au sein d'un vecteur ds de taille $\frac{B(B-1)}{2}$. Les distances numériques sont quant à elles regroupées au sein d'une matrice $\mathcal{DN}$ de taille $(\frac{B(B-1)}{2} \times C)$. Nous recherchons le vecteur de coefficients $\nu = \{\nu_c, c = 1..C\}$ qui minimise l'erreur, au sens des moindres carrés, entre ds et $\mathcal{DN} \cdot \nu$. Nous utilisons pour cela la méthode d'ajustement linéaire de la *GNU Scientific Library* [5]. A partir de ces coefficients d'ajustement ν_c, nous définissons les poids λ_c qui seront utilisés lors de la recherche en temps réel de séquences vidéo similaires (section 5.1). Nous définissons $\lambda_c = |\nu_c|$ car $|\nu_c|$ traduit l'importance du rôle joué par la composante c dans l'ajustement entre la distance numérique et la distance sémantique. Un ajustement sous contrainte de positivité nous permettrait de définir $\lambda_c = \nu_c$, mais la complexité d'un tel ajustement est plus élevée.

6. Application à la base de données HOLLYWOOD2

6.1. Description de la base de données

La méthode proposée a été évaluée sur un ensemble de séquences vidéo extraites de 69 films hollywoodiens : *HOLLYWOOD2 human action* [6] (voir figure 2). L'ensemble d'apprentissage $\mathcal{D}_{train}$ est constitué de 823 séquences [7]. L'ensemble de test $\mathcal{D}_{test}$ est quant à lui constitué de 884 séquences. Il y a en moyenne 500 images par séquence vidéo (durée moyenne : 20 secondes). Les images sont de résolution variable : 640x352, 576x312, 548x226, etc. La présence de douze types d'actions humaines a été indiquée

5. http://www.gnu.org/software/gsl/
6. http://www.irisa.fr/vista/actions/hollywood2/
7. Le sous-ensemble *Training subset (automatic)* n'a pas été utilisé.

pour chaque séquence V (Marszałek *et al.*, 2009) : pour chaque type d'action $\mathcal{A}$, un label binaire $\delta(\mathcal{A},V)$ a été affecté à V.

(a) (b) (c)

Figure 2. Exemples d'actions annotées dans l'ensemble "HOLLYWOOD2 human action"

Notez que la méthode a été pensée pour traiter des flux vidéo continus, c'est-à-dire sans changements de plan. Or, dans HOLLYWOOD2, certaines séquences contiennent plusieurs plans. Cependant, si une sous-séquence est à cheval sur deux ou plusieurs plans, les chances qu'elle contienne une action d'intérêt entière sont faibles, en supposant que la taille des sous-séquences a été bien choisie. Donc, l'algorithme n'est pas censé générer une alerte. Et il a peu de chance d'en générer car, de par le(s) changement(s) de plan, le flot optique ne ressemble pas à celui observé lors d'une action d'intérêt.

6.2. Evaluation quantitative

La méthode a été évaluée en termes d'A_z, l'aire sous la courbe ROC (*Receiver Operating Characteristic*). Pour chaque type d'action et chaque exemple d'organisation des sous-séquences (voir figure 1), un système de poids λ a été appris sur l'ensemble d'apprentissage. L'A_z a ensuite été calculée sur l'ensemble de test. Pour cela, nous avons recherché, pour chaque sous-séquence d'une séquence vidéo $V \in \mathcal{D}_{test}$, les k=5 sous-séquences les plus proches dans l'ensemble d'apprentissage (à l'aide de l'algorithme *k-d tree*). Ensuite, la probabilité que $\delta(\mathcal{A},V) = vrai$ a été estimée par le pourcentage de sous-séquences, parmi les $k(n_V - n)$ sélectionnées, qui sont issues de séquences $U \in \mathcal{D}_{train}$ telles que $\delta(\mathcal{A},U) = vrai$. La courbe ROC est construite en faisant varier un seuil sur ce pourcentage.

6.3. Méthode de référence

Nous avons comparé notre méthode à *Video Google* de Sivic, Zisserman (2003). Dans *Video Google*, les sous-séquences vidéo sont simplement constituées d'une image : l'image courante. Une représentation par sac de mots est utilisée pour caractériser chaque image. Dans notre implémentation de *Video Google*, les descripteurs d'images SURF (*Speeded Up Robust Features*) de Bay *et al.* (2008) ont été utilisés.

7. Résultats

La méthode proposée a été évaluée quantitativement en suivant le protocole décrit en section 6.2. Compte tenu des annotations disponibles pour l'évaluation, cette expérience ne permet d'évaluer que la capacité de la méthode proposée à détecter la présence d'une action dans une séquence. Elle ne permet pas d'évaluer sa capacité à localiser précisément les actions au sein des séquences.

Les résultats obtenus pour chaque action sont présentés dans le tableau 1. Les courbes ROC obtenues pour certaines actions sont présentées dans la figure 3. Mise à part l'action *Kiss*, la méthode proposée s'est montrée plus performante que *Video Google* en termes d'aire sous la courbe ROC. Concernant les actions pour lesquelles les deux méthodes ont des performances proches, telles que *FightPerson* ou *Kiss*, nous voyons sur la figure 3 que la méthode proposée permet d'être très spécifique (*i.e.* d'avoir une précision élevée) et, à l'inverse, *Video Google* permet d'être très sensible (*i.e.* d'avoir un rappel élevé). La méthode proposée permet d'avoir le meilleur compromis sensibilité/spécificité (*i.e.* le meilleur compromis rappel/précision).

Tableau 1. Evaluation des performances (A_z) sur l'ensemble de test

organisation (voir figure 1)	(a)	(b)	(c)	*Video Google*
AnswerPhone	0.786	**0.793**	0.765	0.635
DriveCar	0.849	**0.860**	0.854	0.667
Eat	0.807	0.826	**0.831**	0.578
FightPerson	0.865	0.855	**0.880**	0.821
GetOutCar	**0.746**	0.713	0.734	0.730
HandShake	0.743	0.765	**0.779**	0.533
HugPerson	**0.667**	0.651	0.642	0.618
Kiss	0.825	0.829	0.836	**0.853**
Run	0.849	0.852	**0.853**	0.734
SitDown	0.768	**0.774**	0.739	0.699
SitUp	**0.769**	0.731	0.753	0.707
StandUp	**0.762**	0.752	0.739	0.555

Nous voyons dans le tableau 1 que l'organisation optimale des intervalles élémentaires d'images varie selon le type d'action étudiée, même si l'influence de l'organisation est modeste (voir figure 3). Notez que le choix de l'organisation en intervalles élémentaires d'images, parmi les trois solutions proposées dans la figure 1, est empirique et que d'autres organisations pourraient bien être plus adaptées.

Concernant les temps de traitement, nous avons analysé séparément les temps de caractérisation d'une sous-séquence T_c et les temps de recherche de sous-séquences similaires T_r. Lors de la caractérisation d'une sous-séquence, l'étape la plus longue est l'étape de caractérisation de l'image courante. Le temps de traitement des autres étapes (caractérisation des intervalles élémentaires d'images et caractérisation de la sous-séquence) est négligeable. Le temps de caractérisation d'une image, et donc T_c,

dépendent essentiellement du nombre de pixels par image. En moyenne, dans HOLLYWOOD2, les temps de caractérisation sont de T_c = 28 ms. Les temps de recherche, quant à eux, varient logarithmiquement avec le nombre de sous-séquences dans la base de référence, c'est-à-dire avec le nombre total d'heures de vidéo dans l'ensemble de référence. Dans HOLLYWOOD2, il y a en moyenne près de 500 sous-séquences par séquence vidéo (section 6.1), soit environ 400 000 sous-séquences dans l'ensemble de référence ($\mathcal{D}_{train}$). Les temps de recherche sont de T_r = 9 ms. Au final, avec la méthode proposée, les temps de traitement ($T_c + T_r$ = 37 ms) ont été inférieurs à 40 ms (= 25 images par seconde).

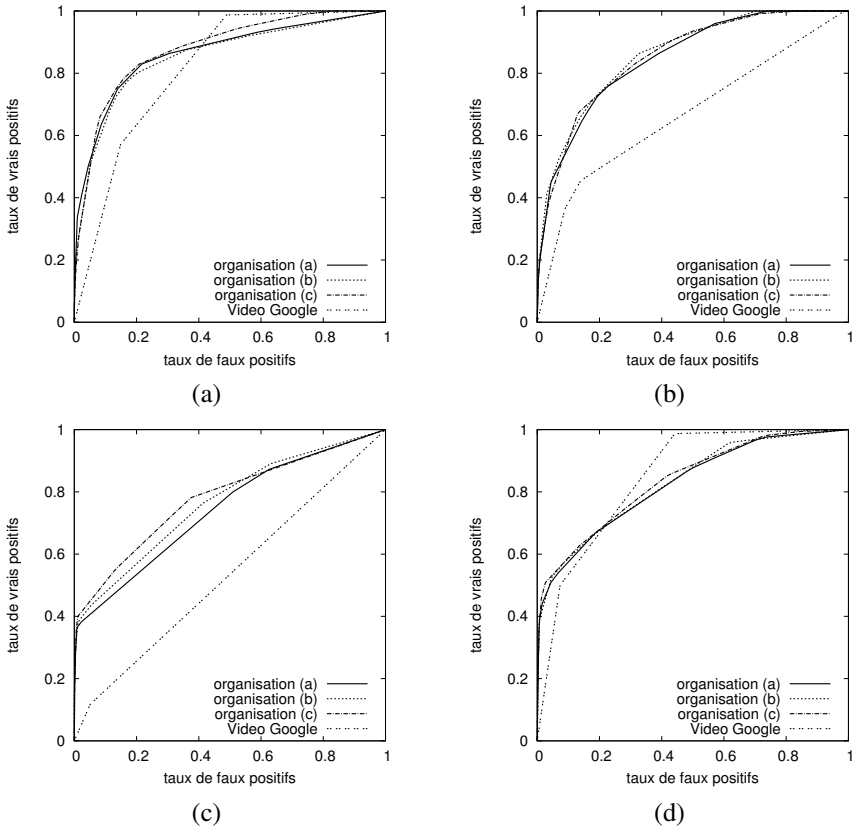

Figure 3. Courbes ROC obtenues pour différentes actions. Une courbe est présentée pour chaque organisation des sous-séquences en intervalles élémentaires d'images (voir figure 1) ainsi que pour la méthode de référence Video Google

8. Conclusion

Nous avons présenté dans cet article une méthode originale pour rechercher des sous-séquences vidéo similaires par le contenu. L'application visée est la génération

en temps réel d'alertes lorsqu'une action d'intérêt est détectée. Dans le domaine de la vidéosurveillance, il peut s'agir de détecter des comportements suspects. Dans le domaine médical, il peut s'agir de prédire des complications au cours d'une chirurgie.

En introduisant de la flexibilité temporelle dans la caractérisation des sous-séquences, nous avons pu éviter l'utilisation de distances flexibles, telles le *Dynamic Time Warping* de Sakoe, Chiba (1978). Ce transfert de flexibilité, de la mesure de distance vers la caractérisation, a rendu possible la recherche en temps réel ($< \frac{1}{25}$ secondes) de sous-séquences vidéo similaires parmi 400 000 sous-séquences de référence. Dans un contexte de génération d'alertes, cette propriété nous semble avantageuse par rapport aux méthodes apparentées (Piriou *et al.* (2006) ; Duchenne *et al.* (2009); Xu, Chang (2008)).

La mesure de distance proposée est générale et s'adapte automatiquement à chaque problème par apprentissage. Par conséquent, la recherche est non seulement rapide, mais également précise (cf. tableau 1). Nous voyons que l'organisation optimale des intervalles élémentaires d'images varie selon le type d'action étudiée. En adaptant l'organisation des sous-séquences en intervalles élémentaires d'images, ainsi que l'échelle de temps, nous nous concentrerons à l'avenir sur l'aide à la chirurgie sous contrôle vidéo. A chaque instant, l'image capturée par la caméra de contrôle, ainsi que les $n-1$ images précédentes, pourront être comparées à des sous-séquences vidéo stockées dans des archives chirurgicales. Cela permettra d'identifier, à tout moment, des situations similaires et, au besoin, de générer des alertes ou des préconisations. La mise en place d'une telle application est difficile de par la rareté et la variété des complications : constituer un ensemble de référence représentatif est donc une tâche de longue haleine.

Bibliographie

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 *MICCAI*, vol. 13, p. 480–487.

Arya S., Mount D. M. (1993). Approximate nearest neighbor queries in fixed dimensions. In *Proc. of the ACM-SIAM symposium on discrete algorithms*, p. 271–280.

Bay H., Ess A., Tuytelaars T., Gool L. van. (2008). Surf: Speeded up robust features. *Comput Vis Image Und*, vol. 110, n° 3, p. 346–359.

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*, vol. 30, n° 9, p. 1520–1533.

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

Duchenne O., Laptev I., Sivic J., Bach F., Ponce J. (2009). Automatic annotation of human actions in video. In *ICCV'2009*, p. 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 *ICAPR'09*, p. 113–116.

Gao H. P., Yang Z. Q. (2010). Content based video retrieval using spatiotemporal salient objects. In *IPTC'10*, p. 689–692.

Gionis A., Indyk P., Motwani R. (1999). Similarity search in high dimensions via hashing. In *Proc. of the 25th very large database (VLDB) conference.*

Hoi S. C. H., Lyu M. R. (2007). A multimodal and multilevel ranking framework for content-based video retrieval. In *ICASSP'07*, vol. 4, p. 1225–1228.

Hu W., Xie D., Fu Z., Zeng W., Maybank S. (2007). Semantic-based surveillance video retrieval. *IEEE trans. on Image Processing*, vol. 16, n° 4, p. 1168–1181.

Juan K., Cuiying H. (2010). Content-based video retrieval system research. In *ICCSIT'10*, vol. 4, p. 701–704.

Lucas B. D., Kanade T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc. imaging understanding workshop*, p. 121–130.

Marszałek M., Laptev I., Schmid C. (2009). Actions in context. In *Ieee conference on computer vision & pattern recognition.*

Mises R. von. (1964). *Mathematical theory of probability and statistics* (H. Geiringer, Ed.). Academic Press, New York.

Naturel X., Gros P. (2008). Detecting repeats for video structuring. *Multimedia Tools and Applications*, vol. 38, n° 2, p. 233–252.

Patel B. V., Deorankar A. V., Meshram B. B. (2010). Content based video retrieval using entropy, edge detection, black and white color features. In *ICCET'10*, vol. 6, p. 272–276.

Pearson K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, vol. 2, n° 6, p. 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*, vol. 15, n° 11, p. 3417–3430.

Quellec G., Lamard M., Cazuguel G., Cochener B., Roux C. (2010). Wavelet optimization for content-based image retrieval in medical databases. *Med Image Anal*, vol. 14, n° 2, p. 227–241.

Sakoe H., Chiba S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE trans. on Acoustics, Speech and Signal Processing*, vol. 26, n° 1, p. 43–49.

Sivic J., Zisserman A. (2003). Video google: A text retrieval approach to object matching in videos. In *Proc int conf on computer vision*, p. 1470–1477.

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

Annexe A Caractéristiques d'images utilisées

Pour décrire une image V_i, nous avons extrait des caractéristiques de texture et de couleur de V_i ainsi que des caractéristiques de mouvement du flot optique entre V_{i-1} et V_i.

Les caractéristiques de texture et de couleur ont été extraites dans la transformée en ondelettes de chaque canal de couleur de V_i, par une méthode que nous avons proposée précédemment. Cette méthode fournit une description paramétrique de la distribution des coefficients d'ondelette (Quellec *et al.*, 2010) dans chaque sous-bande de la décomposition (au nombre de 9). La distribution des coefficients d'ondelette dans une sous-bande est approchée par une distribution gaussienne généralisée définie par deux paramètres : α et β. Nous obtenons donc un vecteur de dimension 54 (3 canaux de couleur $\times$ 9 sous-bandes $\times$ 2 paramètres).

Le mouvement est quant à lui caractérisé par quatre histogrammes : un histogramme de l'amplitude des vecteurs déplacements, un histogramme de l'angle des vecteurs déplacements pondérés par leur amplitude, ainsi qu'un histogramme de l'abscisse et un histogramme de l'ordonnée des points d'intérêts, pondérés par l'amplitude des vecteurs déplacements en ces points. Les deux derniers histogrammes permettent de localiser le mouvement dans l'image. Chacun de ces histogrammes comporte huit niveaux. Au final, nous obtenons donc un vecteur de dimension 86 (54 + 4 $\times$ 8 niveaux).

Gwénolé Quellec *est chargé de recherche à l'Inserm. Il est membre du Laboratoire de Traitement de l'Information Médicale (LaTIM). Ses travaux portent sur la recherche d'images et de vidéos par le contenu dans des bases de données médicales pour l'aide au diagnostic et à la chirurgie. Il s'intéresse plus particulièrement aux applications ophtalmologiques.*

Mathieu Lamard *est ingénieur de recherche à l'Université de Bretagne Occidentale. Il est membre du Laboratoire de Traitement de l'Information Médicale (LaTIM). Ses travaux portent sur la recherche d'images et de vidéos par le contenu dans des bases de données médicales pour l'aide au diagnostic et à la chirurgie. Il s'intéresse plus particulièrement aux applications ophtalmologiques.*

Guy Cazuguel *est enseignant-chercheur, directeur d'études, à Télécom Bretagne. Il est membre du Laboratoire de Traitement de l'Information Médicale (LaTIM). Ses travaux portent sur la recherche d'images et de vidéos par le contenu dans des bases de données médicales pour l'aide au diagnostic et à la chirurgie. Il s'intéresse plus particulièrement aux applications ophtalmologiques.*

Zakarya Droueche *est doctorant à Télécom Bretagne. Il est membre du Laboratoire de Traitement de l'Information Médicale (LaTIM). Ses travaux portent sur la recherche de vidéos par le contenu dans des bases de données médicales pour l'aide à la chirurgie. Il s'intéresse plus particulièrement aux applications ophtalmologiques.*

Béatrice Cochener *est professeur (PU-PH) au CHRU de Brest. Elle est Chef du service d'Ophtalmologie et membre du Laboratoire de Traitement de l'Information Médicale (LaTIM). Elle est une spécialiste internationale de la chirurgie du segment antérieur. Ses travaux de recherche portent sur l'évaluation clinique de techniques d'imagerie et de chirurgie, ainsi que sur la recherche d'images et de vidéos par le contenu.*

Christian Roux *est professeur à Télécom Bretagne. Il est directeur scientifique de Télécom Bretagne et membre du Laboratoire de Traitement de l'Information Médicale (LaTIM), dont il est le fondateur. Ses travaux portent sur le traitement de l'information médicale, la modélisation de l'information spatiale et fonctionnelle et l'analyse d'images médicales.*