Logiques de descriptions pour l'analyse structurelle de film

Description Logics for Structural Analysis of Film

Jean Carrive[1], François Pachet[2], Rémi Ronfard[1]

[1]Institut National de l'Audiovisuel (INA)
[2]SONY CSL-Paris & Lip6 (Paris 6)

Institut National de l'Audiovisuel
Direction de la Recherche
4, avenue de l'Europe – 94 366 Bry sur Marne

{jean, remi}@ina.fr, pachet@csl.sony.fr

Résumé

Dans le contexte de l'annotation de documents filmiques assistée par ordinateur, nous posons le problème de l'analyse filmique automatique, et identifions 3 problèmes de base : classification de plans, regroupement temporel et pilotage d'algorithmes d'extraction. Ces trois processus mettent en œuvre des connaissances provenant de plusieurs domaines d'expertise : documentalistes, spécialistes d'analyse du signal et professionnels de l'audiovisuel.

Nous proposons d'utiliser le formalisme des logiques de descriptions comme paradigme principal de représentation pour représenter ces divers types de connaissances dans un environnement intégré. Nous proposons dans ce cadre un mécanisme de regroupement temporel fondé sur une restriction du formalisme de Allen qui permet de limiter les problèmes de complexité.

Mots Clef

Logiques de descriptions, indexation, raisonnement temporel, audiovisuel

Introduction

La numérisation systématique des documents filmiques ainsi que la production de documents accessibles au grand public ont connu ces dernières années une croissance exponentielle [1]. Le problème de l'indexation de grands volumes de documents filmiques devient ainsi une préoccupation importante à la fois des industriels de l'audiovisuel et des centres documentaires comme l'INA (Institut National de l'Audiovisuel).

Ce problème concerne des chercheurs de domaines très divers : bases de données, analyse d'image, traitement du signal sonore, recherche d'information, etc. Un schéma général d'indexation peut être vu comme étant composé de plusieurs étapes : analyse automatique (extraction de primitives et segmentation spatio-temporelle de bas niveau), annotation et découpage manuels, stockage en base de données. Les techniques de segmentation temporelle actuellement disponibles ne permettent pas d'extraire des unités temporelles suffisamment longues et pertinentes pour qu'il soit concevable de les annoter manuellement. Nous proposons une méthode de regroupement des unités temporelles de bas niveau, extraites automatiquement, en unités temporelles de plus haut niveau. Le but de cette méthode est d'offrir au documentaliste chargé de l'annotation un découpage du document en unités pertinentes – appelées *séquences* –, dont la cohésion permette qu'elles soient annotées comme un tout.

Cette extraction nécessite une ingénierie des connaissances adaptée qui permette à des experts des domaines concernés (documentalistes, professionnels de l'audiovisuel et experts en analyse du signal) de spécifier simplement leurs connaissances du domaine. Cette étude se place dans le contexte du projet européen DIVAN (Distributed Video Archives Network) auquel participent l'INA, l'IRISA, la Rai, etc.

Nous allons décrire précisément le problème de l'indexation dans ce contexte dans la première section, en identifiant trois problèmes de base. Nous identifions ensuite les logiques de descriptions comme un « bon » formalisme pour représenter le contenu des documents filmiques (radio, télévision) ainsi que les requêtes portant sur ces documents. Dans cet article, nous identifions le problème de l'analyse des documents à partir d'observations extraites par des automates. Nous proposons d'enrichir les logiques de descriptions avec un système de raisonnement temporel. Nous faisons le choix d'une représentation par intervalles permettant de modéliser des actions simultanées (par exemple événements visuels et sonores) et d'assurer le regroupement d'unités temporelles avec une complexité raisonnable. Nous considérons pour le moment que toutes les actions sont celles du monteur ou du réalisateur. Dans l'avenir, nous espérons modéliser également les personnages et leurs actions, ce qui pose des problèmes difficiles, à peine ébauchés dans le domaine de la vision par ordinateur (voir tout de même [2] et [3]).

Enfin, après une courte discussion, nous concluons sur l'état courant de notre implémentation et sur des perspectives d'extension.

1. Analyse filmique

L'analyse, dans le sens dans lequel nous l'entendons, consiste à reconstituer la structure temporelle d'un document à partir d'informations « primitives » obtenues le plus souvent par des algorithmes d'analyse du signal. Certaines de ces informations primitives peuvent être obtenues *a priori*, de manière systématique, pour tous les documents (découpage en plan, histogrammes de couleurs, etc.). D'autres informations nécessitent la mise en œuvre d'algorithmes plus complexes et ne peuvent être obtenues que sur demande explicite, assorties d'informations contextuelles nécessaires au bon fonctionnement de l'algorithme (reconnaissance de caractères ou de visages dans une image).

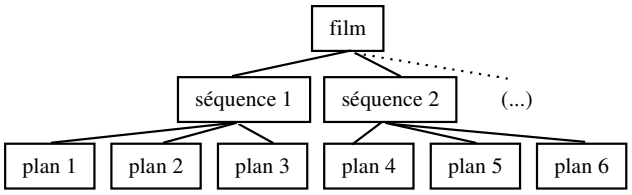

figure 1 : découpage d'un film

Dans la conception traditionnelle du film, représentée aujourd'hui par le cinéma hollywoodien, la structure temporelle du film se présente comme un arbre où les plans, unités élémentaires, sont regroupés en séquences [4] (voir figure 1). Cette analyse pose trois problèmes de fond : 1) la représentation des primitives d'extraction et leur organisation dans une taxonomie de plans, 2) le regroupement automatisé de ces plans en séquences, et 3) le pilotage explicite d'algorithmes complémentaires d'extraction permettant de raffiner l'analyse. Ce schéma général est représenté figure 2.

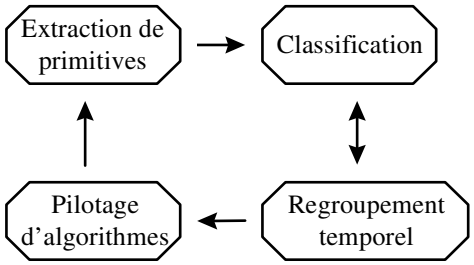

figure 2 : schéma général d'analyse

Après avoir présenté l'état actuel de la modélisation des documents filmiques à l'INA, nous allons présenter dans les sections suivantes chacun de ces problèmes.

1.1 Modèles de documents filmiques

Certains éléments de base du langage filmique sont représentés sous forme de taxonomie *explicite* ; d'autre part, il existe une taxonomie *implicite* des types de documents, qui sont représentés sous la forme d'un ensemble de « fiches collection ».

1.1.1 Taxonomie des événements filmiques

Les connaissances générales du domaine pouvant être formalisées recouvrent les éléments propres à la production : prise de vue et montage essentiellement. Les différents cadrages, par exemple – gros plan, plan moyen, plan serré, etc. – peuvent être décrits par une hiérarchie. De même, les mouvements de caméra traditionnels se prêtent au même type de description : caméra fixe (sans mouvement), les divers types de zooms (zoom avant, zoom arrière), de travellings (latéral, avant, arrière), etc. Une taxonomie des événements filmiques (TEF) existe ; il est possible d'en donner une représentation formelle partielle. La figure 3 donne une vision simplifiée des mouvements de caméra classiques.

Un des éléments importants du domaine est le *plan*. Le plan est généralement défini comme la plus petite unité syntaxique du film, et ne comporte ni coupure de caméra ni raccord [5], bien qu'on puisse le subdiviser en unités plus petites, surtout lorsque des mouvements de caméra complexes le composent. Les propriétés communément admises du plan sont la durée (plan long ou plan court), le type de transition avec le plan qui le précède et celui qui le suit : transition de type *cut* (coupe franche) ou de type graduel (fondus, volets, etc.), le nombre de personnages, le cadrage des personnages (gros plan, plan moyen, etc.). Une formalisation des propriétés générales d'un plan a été proposée dans [5].

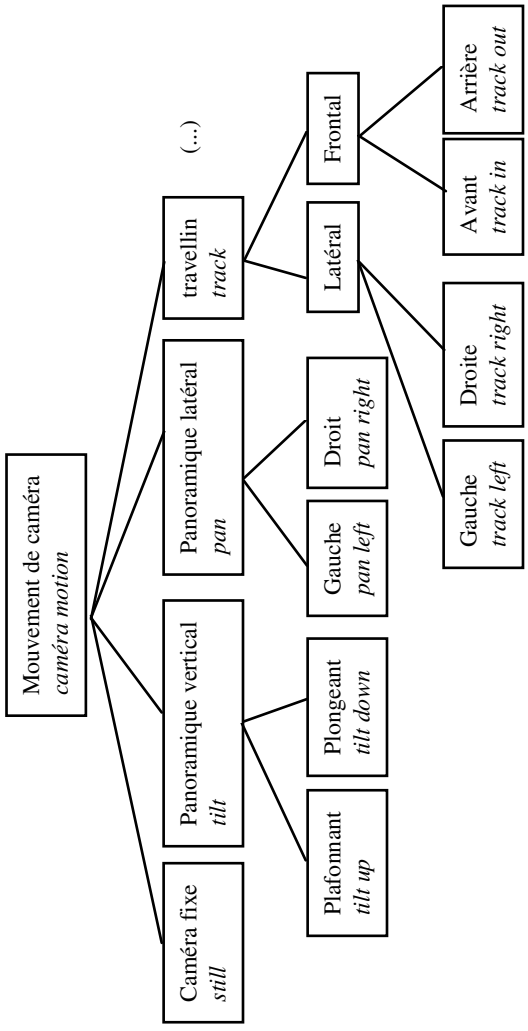

figure 3 : mouvements de caméra dans la TEF

1.1.2 Taxonomie des types de documents

Des plans consécutifs peuvent être regroupés en *séquences*, qui constituent des unités sémantiques. Les

émissions de télévision suivent très souvent un canevas relativement précis. C'est le cas pour les journaux télévisés, qui ont été jusqu'ici les plus étudiés, mais c'est également le cas de beaucoup d'autres émissions : magazines de reportages, de variétés, feuilletons ou sit-coms. Des modèles d'émission peuvent alors être établis. C'est le cas à l'INA, où de tels modèles, appelés « fiches collection », servent à indexer manuellement un grand nombre d'émissions. Une fiche collection est donnée à titre d'exemple par la figure 4.

Projection Privée
Produit par S. Bleckmans
Réalisé par E. David
Présenté par L. Weil

Magazine à rubriques hebdomadaire consacré à l'actualité cinématographique (...).

Chaîne de 1ère diffusion Métropole Télévision
Date de début 17/10/89
(...)

Genre : cinéma - **Forme** : magazine - **Descripteurs** : cinéma, film (...)

Résumé
Principe de l'émission : Ce magazine, présenté par L. Weil, est composé de présentation de sujets promotionnels sur les films qui vont sortir (...)
Déroulement chronologique : Durant le premier plateau, *xxx* présente l'émission (sommaire en images). Il lance ensuite le premier sujet (...)

Dispositifs
Générique : début sur fond noir, incrustation du titre
Dispositifs plateau : monocaméra, le siège du présentateur est (...)
Construction générale : plans généraux du plateau, plans taille du présentateur, passage d'un plan à l'autre par zoom avant
Construction des éléments : Sur les premières images, des sujets, le titre est incrusté sur un large bandeau noir (...)

etc.

figure 4 : exemple de fiche collection (source : Inathèque de France)

1.2 Agrégation temporelle de plans en séquences

Annoter un document au niveau du plan n'est pas réaliste : les plans sont en général courts, et un document peut comporter plusieurs centaines, voire plusieurs milliers de plans. Le regroupement automatique de plans en séquences constitue donc un axe de recherche important qui permettrait de faciliter l'indexation et l'annotation d'un plus grand nombre de documents.

Comme nous l'avons vu dans la section précédente, certaines caractéristiques d'un document filmique, film ou vidéo, peuvent être extraites du document de manière algorithmique. C'est le cas par exemple des mouvements de caméra (plans fixes, zooms, travellings), des transitions de plan (cuts, fondus) [6], de la présence de musique, etc. Cependant, ces caractéristiques se trouvent d'une part être en nombre limité et d'autre part présentent généralement un contenu sémantique faible. Il se pose alors le problème de les combiner entre elles afin de faire apparaître des entités de plus haut niveau, en rapport plus étroit avec l'idée que l'on peut se faire des éléments constitutifs du document. Le principe d'analyse dont nous avons besoin consiste à construire ces entités pertinentes en assemblant des caractéristiques primitives. L'analyse doit donc être générative.

1.2.1 Règles structurelles générales

Certaines règles de composition peuvent s'appliquer à un grand nombre de genres différents de films ou de documents vidéo. Il s'agit essentiellement de règles de montage qui, schématiquement, expriment la manière d'assembler les plans entre eux afin de ne pas rompre la continuité du discours : continuité des positions, des directions, des regards, des éclairages, etc. Le montage est parfois considéré comme l'essence même de l'art cinématographique [4].

L'étude menée dans [7] formalise quelques règles de montage ne nécessitant pas de connaissances préalables sur le document traité : les objets intervenants dans les prémisses des règles sont directement observables dans le flux vidéo et elles s'appliquent de manière générale pour tout type de document. Ces règles sont essentiellement fondées sur l'observation de l'alternance de différents types de transitions de plan (« cuts » et transitions graduelles), sur la similarité des plans entre eux, avec des mesures simples sur les histogrammes de couleurs, sur la présence ou l'absence de musique, sur la durée relative des plans, etc. La règle illustrée par la figure 5 signifie que lorsqu'on se trouve en présence d'au moins quatre plans séparés par des « cuts » suivis d'une transition graduelle, elle-même suivie à nouveau de quatre plans séparé par des « cuts », alors on peut estimer qu'il y a une rupture de séquence au moment de la transition graduelle.

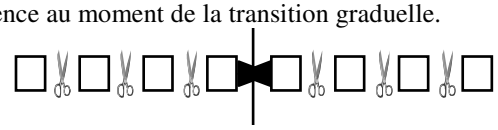

figure 5 : changement de séquence en fonction des transitions de plan, extrait de [7]

1.2.2 Connaissances spécifiques

Cependant, ces règles générales ne peuvent rendre compte de la diversité de tous les types de documents. Selon le genre d'émission (journaux télévisés, magazines de variété), la structure temporelle d'un document filmique peut être plus ou moins complexe et précise. Des modèles de documents spécifiques ont déjà été étudiés, concernant à notre connaissance principalement la structure du journal télévisé [8].

Dans les cas les plus simples, cette structure temporelle peut se formaliser comme une expression régulière. Ainsi, beaucoup de magazines alternent scènes de plateaux et scènes de reportage. La structure de ce type de documents peut alors se mettre sous la forme :

```
générique_début
(plateau reportage)*
générique_fin.
```

De la même manière, la structure de certains documents peut être décrite par une grammaire formelle, ou sous la forme d'une DTD SGML/HyTime, comme c'est le cas dans [9]. Il nous semble cependant que ces approches, si elles sont parfaitement justifiées pour un type particulier de documents, comme le journal télévisé, ne peuvent répondre au cas plus général de la description de différents types de documents.

Ces modèles de documents ne doivent pas seulement indiquer la structure temporelle respectée par les émissions appartenant à une collection donnée, ils doivent également indiquer les élément non temporels que l'on retrouve généralement d'une émission sur l'autre. Il peut s'agir du ou des présentateurs, d'éléments de décor, du logo de l'émission, des génériques, des jingles, etc. Dans la perspective d'un modèle automatisé, des descriptions numériques ou des exemples sont précieux, qui permettent une détection algorithmique des éléments récurrents : modèles phonétiques des voix des principaux intervenants, images numérisées des logos, etc.

Le modèle d'un document étant connu *a priori* à partir de simples données de catalogage, le problème se pose alors de contraindre les algorithmes d'analyse de bas niveau, en fournissant à ces algorithmes des données dépendantes du contexte.

1.3 Pilotage d'algorithmes

Nous décrirons par un exemple le type de situation dans lequel il est nécessaire de piloter des algorithmes d'extraction de primitives afin de raffiner l'analyse. Considérons un magazine de cinéma. Un tel magazine est composé principalement d'une alternance de scènes de studio et de séquences d'extraits d'un film à l'affiche. Les scènes de studio peuvent être isolées des séquences d'extraits en utilisant diverses méthodes : les scènes de reportages sont par exemple encadrées en haut et en bas par des bandes noires caractéristiques, ce qui est facile à détecter pour un algorithme d'analyse d'image. D'autre part, le modèle de document précise qu'au cours des cinq premières secondes des séquences d'extraits apparaît un panneau noir à gauche de l'écran où se trouve inscrit le titre du film dont il est question.

Il est dans ce cas possible dans un premier temps d'isoler les séquences d'extraits des scènes de studio par détection des bandes noires, et dans un second temps de n'exécuter un algorithme d'extraction de texte que sur une partie réduite du document.

2. Logiques de descriptions : un formalisme pour l'analyse

Le problème de l'analyse filmique étant posé dans son ensemble, se pose alors celui du cadre technologique dans lequel l'exprimer en vue d'une ingénierie des connaissances outillée et efficace. Le formalisme des logiques de descriptions est particulièrement bien adapté à la représentation et l'exploitation de taxonomies naturelles, et nous avons déjà proposé dans [10] de le prendre comme outil conceptuel de base pour nos travaux. Les logiques de descriptions sont des langages de représentation des connaissances bien étudiés et formalisés [11]. Cependant, s'il est naturel d'utiliser les logiques de descriptions comme langage de représentation pour les objets du domaine filmique, il paraît plus malaisé de s'en servir pour exprimer des connaissances à forte composante temporelle. Nous verrons plus loin que les possibilités de règles de production en chaînage avant qu'offre CLASSIC [12], système de logique de descriptions que nous avons utilisé pour l'implémentation, permet de résoudre un certain nombre de problèmes.

2.1.1 Classification et logiques de descriptions

Les logiques de descriptions forment un ensemble de langages de représentation de connaissances ; elles permettent de représenter des connaissances de manière structurée en séparant les définitions de concept (base terminologique ou *TBox*) des descriptions d'individus (base des assertions ou *ABox*). Un concept représente un un ensemble d'individus. Les *rôles* représentent des relations binaires. Descriptions de concepts et rôles sont organisés en hiérarchies par la relation de *subsomption* : le concept C subsume le concept D si les instances de D sont nécessairement instances de C. La *classification* est l'opération permettant d'attribuer une place à un concept (ou éventuellement à un rôle) dans la hiérarchie des concepts (ou des rôles). On trouvera dans [11] une introduction aux logiques de descriptions, un ouvrage de référence restant [13].

Nous proposons de poursuivre la piste indiquée par [7] en nous plaçant dans le cadre des logiques de descriptions. Nous posons ici le problème de l'agrégation temporelle décrit dans la section 1.2 dans le contexte des logiques de descriptions. Nous pensons d'autre part que dans le cadre de l'analyse de documents filmiques, seuls un nombre très restreint de règles peuvent s'appliquer en toute généralité, et qu'il convient lorsque cela est possible de formaliser des modèles de documents représentant un type particulier de documents (journal télévisé, magazine, par exemple).

2.2 Représentation de connaissances

La taxonomie TEF se représente aisément à l'aide du formalisme des logiques de descriptions. Comme il a été dit plus haut, certaines connaissances du domaine filmique peuvent être assez naturellement exprimées sous forme de hiérarchies : types de cadrage, mouvements de caméra, types de transition de plans, types d'éclairage (intérieur/extérieur, par exemple), nombre de personnages, etc. Tous ces éléments s'expriment de manière naturelle dans un langage de description comme CLASSIC.

Il est important de faire la distinction entre les aspects temporels d'un plan (ses dates de début et de fin) et sa description non temporelle, c'est-à-dire ses propriétés (ses *rôles*, pour reprendre la terminologie des logiques de descriptions).

Par exemple, conformément à la taxonomie TEF, on peut représenter le fait qu'un plan à 3 personnages dont le principal mouvement de caméra est un travelling avant (track-in), défini par l'expression CLASSIC suivante :

```
(define-concept 'SHOT-1 '(and
    (exactly 1 camera-motion)
    (fills camera-motion track-in)
    (exactly 1 character-num)
    (fills character-num 1)))
```

est plus spécifique que le concept de plan ayant entre 1 et 5 personnages et dont le principal mouvement de caméra est un travelling, défini par :

```
(define-concept 'SHOT-2 '(and
    (exactly 1 camera-motion)
    (fills camera-motion tranck)
    (exactly 1 character-num)
    (all character-num (min 1) (max 5)))
```

On obtient ainsi une opérationalisation de la taxonomie TEF, qui forme une ontologie explicite des types de plans.

2.3 Raisonnement temporel

Afin d'exprimer les règles reflétant la structure des documents, il convient de se doter d'un modèle temporel permettant d'exprimer des contraintes sur les occurrences d'événements. Il sera tout d'abord question du choix de ce modèle, puis nous nous intéresserons à un schéma de règles de regroupement temporel.

2.3.1 Un modèle temporel

Le choix du modèle de raisonnement temporel utilisé est important. Il s'agit en général d'un compromis entre expressivité et efficacité.

2.3.1.1 Logique de Allen restreinte

Pour exprimer la structure général d'un document filmique, il faut pouvoir exprimer des contraintes sur les occurrences d'événements temporels : un plan en intérieur suivi d'un plan en extérieur, une musique commençant pendant le dernier plan d'une séquence, etc.

Dans le domaine de l'analyse de la vidéo, [14] propose une théorie – PNF calculus[1] –, fondé sur la logique temporelle d'intervalles de Allen [15] et assortie d'un algorithme en temps polynomial pour la reconnaissance de structures temporelles. S'il est à noter que c'est l'une des rares tentatives d'exploiter le raisonnement temporel pour la représentation de la vidéo, il faut également remarquer que le pouvoir d'expression de cette théorie est trop limité : l'exigence d'un algorithme rapide (en $O(n^2)$) conduit les auteurs à regrouper les relations temporelles dans trois classes correspondant aux notions intuitives de *passé*, *présent* et *future*, ce qui constitue une perte d'information importante.

Nous proposons d'utiliser l'algèbre d'intervalles proposée par [16] – *Pointizable Interval Algebra* –, qui consiste à transformer les contraintes sur des disjonctions de relations de Allen (figure 6) entre intervalles temporels en conjonctions de contraintes sur les bornes de ces intervalles. Seul un sous-ensemble de l'algèbre proposé par Allen est ainsi représentable. Par exemple, la relation temporelle :

A {*before* ∨ *meets* ∨ *overlaps*} B

s'exprime par la conjonction de contraintes :

début(A) < début(B)
fin(A) < fin(B)[2]

mais la relation

A {*before* ∨ *after*} B

n'a pas d'équivalent.

Ce modèle est moins complet que celui de Allen, mais le test de cohérence peut être effectué en temps polynomial. D'autre part, la mise en œuvre de cette algèbre est relativement simple. Plus récemment, [17] a proposé un sous-ensemble maximal calculable de la logique de Allen, la sous-classe *ORD-Horn*, qui reprend les travaux de van Beek [16] en s'affranchissant du passage des intervalles aux bornes d'intervalles. Nous n'excluons pas d'utiliser ce formalisme dans un second temps.

Relation	**Exemple**
X equals Y	
X meets Y Y met-by X	
X overlaps Y Y overlapped-by X	
X during Y Y includes X	
X starts Y Y started-by X	
X finishes Y Y finished-by X	
X before Y Y after X	

figure 6 : les 13 relations temporelles de Allen

2.3.1.2 Subsomption de relations temporelles

Nous avons choisi pour l'instant de réifier les relations temporelles. La principale raison est que nous bénéficions ainsi des mécanismes de classification de CLASSIC. Ainsi, la relation temporelle {*before* ∨ *meets* ∨ *overlaps*} subsume-t-elle la relation {*before* ∨ *meets*}, ce qui correspond bien à l'intuition : l'ensemble des segments temporels mis en relation par {*before* ∨ *meets*} l'est *a fortiori* par la relation {*before* ∨ *meets* ∨ *overlaps*}.

Du point de vue de l'implémentation, CLASSIC offre la possibilité sous certaines restrictions de définir un concept comme étant l'ensemble des instances qui satisfont un test (fonction Lisp renvoyant un booléen), et de définir explicitement des liens de subsomption entre ces tests afin que le système puisse classifier concepts et instances.

Il est donc possible de définir un concept temporel (TEMPORAL) comme ayant un début et une fin, le début précédent la fin[3] :

[1] PNF : past, now, future

[2] avec bien sûr début(A) < fin(A) et début(B)<fin(B)

[3] La syntaxe utilisée est proche de celle de CLASSIC, avec quelques facilités d'écriture pour les contraintes de

```
(define-concept 'TEMPORAL '(and
    (exactly 1 begin)
    (all begin integer)
    (exactly 1 end)
    (all end integer)
    (< begin end)))
```

De la même manière, on définit le concept de relation temporelle comme faisant intervenir deux instances de TEMPORAL.

```
(define-concept 'TEMPORAL-RELATION '(and
    (exactly 1 temporal-1)
    (all temporal-1 TEMPORAL)
    (exactly 1 temporal-2)
    (all temporal-2 TEMPORAL)))
```

La relation temporelle définie ci-dessus est la relation la plus générale. Quelles que soient deux intervalles temporels, ils sont toujours en relation !
Les 13 intervalles de base de Allen sont définis comme sous-concepts de TEMPORAL-RELATION. La relation *meets* est par exemple définie par :

```
(define-concept 'TEMPORAL-MEETS '(and
    TEMPORAL-RELATION
    (= (temporal-1 end) (temporal-2
    begin)))
```

CLASSIC ne permettant pas de définir des concepts comme disjonction de concepts, les concepts qui correspondent aux relations temporelles s'exprimant comme disjonction des relations de Allen doivent être décrites explicitement, ainsi que les relations de subsomption entre les tests correspondants.
Une remarque doit être faite quant à la manière dont est effectué le test *a* R *b* (*a* et *b* sont-ils mis en relation par R ?). Dans l'implémentation existante, une instance de TEMPORAL-RELATION est créée. Si cette instance se trouve être classifiée par CLASSIC comme instance de R, alors la relation est respectée. Il est évident que cette manière de procéder en réifiant les relations temporelles est très coûteuse, mais cela nous permet dans un premier temps d'une part d'exprimer clairement ces relations et d'autre part de profiter du mécanisme de subsomption offert par CLASSIC.

2.4 Règles de regroupement

Pour exprimer la structure temporelle des documents, nous proposons d'utiliser des règles de regroupement temporel qui agrègent des formes temporelles de bas niveau en des formes de plus haut niveau.

2.4.1 Principe général

Le principe de regroupement temporel présenté ici repose sur la possibilité offerte par CLASSIC d'associer des règles à un concept, lesquelles règles sont déclenchées à chaque instanciation de ce concept. Lorsqu'une instance *a* du concept A est créée, une règle et déclenchée. Dans notre cas, le traitement de cette règle consiste à chercher toutes les instances b_i du concept B telles que *a* R b_i, avec R instance d'un sous-concept de TEMPORAL-RELATION (R est une relation temporelle). *a* et b_i sont alors agrégées pour former une instance de plus haut niveau, de concept G. Ce type très général de règles fait intervenir quatre paramètres : les concepts A, B, R et G. Les nouvelles instances ainsi créées peuvent à leur tour être agrégés en groupes d'encore plus haut niveau.

2.4.2 Trois types de règles

Dans un premier temps, deux types de règles ont été identifiés qui permettent de regrouper des éléments temporels : les règles qui agrègent deux instances de concepts différents en une instance d'un concept de plus haut niveau, et les règles qui agrègent N instances du même concept en une instance d'un concept de plus haut niveau.
Dans le premier cas, on cherche à regrouper des formes qui se complètent. Ce sera le cas par exemple du champ-contre champ : admettons que l'on ait pu segmenter un document selon le schéma de la figure 7. Les cases représentent les plans. Les plans notés A et B sont des plans d'un type quelconque. Les plans notés I (resp. F) signalent la présence du personnage I (resp. F). Il paraît naturel ici de regrouper les plans sur le schéma :

A* (I F)* B*

La règle de regroupement qui s'applique ici regroupe un plan de type I suivi d'un plan (relation temporelle *meets*) de type F en un groupe de deux plans I-F. Le premier type de règles peut se décrire sous la forme :

$$C_1 \, R \, C_2 \rightarrow G \qquad (1)$$

avec :
C_1, C_2 : concepts héritant de TEMPORAL
R : concept héritant de TEMPORAL-RELATION
G : concept héritant de TWO-TEMP-GRP, groupe de deux instances de TEMPORAL. Ce concept est défini comme suit :

```
(define-concept 'TWO-TEMP-GRP '(and
    TEMPORAL
    (exactly 1 first-temporal)
    (all first-temporal TEMPORAL)
    (exactly 1 second-temporal)
    (all first-temporal TEMPORAL)))
```

Dans l'exemple cité, la règle de regroupement serait donc :

```
I meets F → I-F
```

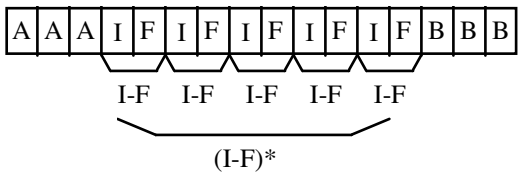

figure 7 : champ-contre champ

Une fois regroupés les plans I et les plans F en groupes I-F , on voudrait pouvoir regrouper ces derniers en une séquence d'instances de I-F. Un deuxième type de règles intervient ici qui agrège des instances de la même classe en une instance d'une classe de plus haut niveau :

$$C \, R \rightarrow G \qquad (2)$$

avec :
C : concept héritant de TEMPORAL

---

cardinalité et les tests Lisp. Ces derniers sont notés en italique.

R : concept héritant de TEMPORAL-RELATION
G : concept héritant de TEMPORAL-SEQ, séquence d'instances de TEMPORAL, défini par :

```
(define-concept 'TEMPORAL-SEQ '(and
    TEMPORAL
    (at-least 2 element)
    (all element TEMPORAL)))
```

Ce type de règles permet d'instancier une séquence temporelle dont les éléments appartiennent à la même classe et sont deux à deux mis en relation par une relation temporelle donnée.
Il est apparu qu'à ces deux types de règles d'agrégation devaient s'ajouter des règles permettant d'exprimer des ruptures temporelles ou des transitions. C'est le cas de la règle de transition entre séquences présentée plus haut (voir figure 5). Ces règles permettraient d'identifier des structures du type ABA' comme indiquant que B est un élément de transition. Dans l'exemple décrit, A et A' représentent une suite de plans séparés par des « cuts », et B représentent une transition graduelle. La règle tirée de [7] et illustrée par la figure 8 indique que lorsque deux plans consécutifs présentent des mesures colorimétriques voisines au sens d'une certaine distance, il y a de fortes chances qu'il y ait une rupture de séquences entre ces deux plans. Cette structure est de la forme ABB'C. Il est possible qu'un type spécifique de règles soit utile pour pouvoir effectuer des regroupements représentant ce genre de structure. Cependant, il faut noter que des structures de type AB A' ou ABB'C peuvent être exprimées avec une combinaison de règles de type (1). Dans le premier cas, on peut agréger A et B en A-B, puis A-B et A' en A-B-A', dans le deuxième on peut également décomposer la règle, ce qui nuit bien sûr à la clarté des règles, et ce qui conduit à créer des concepts intermédiaires n'ayant pas nécessairement de sens. Cependant, il doit être possible d'automatiser la décomposition de ces règles et ainsi de « cacher » ces concepts intermédiaires.

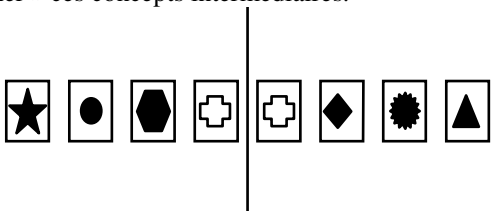

figure 8 : changement de séquence en fonction de similarités de plan

2.4.3 Principes d'exécution et monotonie

Les règles de regroupement étant exprimées de la manière exposée plus haut, il se pose alors un problème de monotonie lié à l'ordre dans lequel sont créées les instances. Une règle de type A R B → G est déclenchée à chaque création d'une instance *a* du concept A. L'action effectuée par la règle consiste à chercher toutes les instances b_i du concept B mis en relation par R avec *a*, puis à agréger *a* et b_i en une instance du concept G. Ce principe est illustré par l'algorithme suivant :

```
Création d'une instance a ⊂[4] A          (i)
Déclenchement de la règle                 (ii)
Recherche des b_i ⊂ B tq a R b_i          (iii)
Agrégation de a et b_i                    (iv)
```

Les règles étant déclenchées par l'instanciation de concept, on constate que l'ordre dans lequel les instances sont créées est important : deux instances ne peuvent être agrégées que si elles existent déjà au moment du déclenchement de la règle. Nous proposons ici une manière de résoudre ce problème pour chacun des deux types de règles exposés plus haut.
Pour les règles de type (1) il convient d'engendrer pour chaque règle la règle symétrique quant à la relation temporelle. Ainsi, pour chaque règle de type :

$$A\ R\ B \rightarrow G$$

doit-on également créer la règle :

$$B\ \overline{R}\ A \rightarrow G$$

avec $\overline{R}$ la relation temporelle inverse de R, définie par :

$$x\ R\ y \Leftrightarrow y\ \overline{R}\ x$$

Par exemple, la relation inverse de la relation temporelle {*before* ∨ *meets* ∨ *overlaps*} est {*after* ∨ *met-by* ∨ *overlapped-by*}.
Pour les règles de type (2), la procédure est un peu plus complexe. Le principe des règles inverses doit également être appliqué mais n'est cependant pas suffisant.
L'étape `(iv)` de l'algorithme consiste à agréger *a* et *a'*, instances du même concept A.
Pour la règle de type (2) A R → G regroupant N instances de A en une instance de G, une manière naïve de procéder consiste, lors du déclenchement de la règle par la création d'une instance *a* du concept A, à chercher une instance *g* du concept G telle que *a* soit en relation temporelle R avec l'une des valeurs du rôle `element` de *g*, puis à ajouter *a* comme valeur du rôle `element` à *g*. Si une telle instance *g* n'est pas trouvée, on cherche une instance *a'* de A telle que *a* et *a'* sont en relation temporelle R. Si un tel *a'* existe, *a* et *a'* sont alors agrégés en une nouvelle instance de G.

```
chercher g ⊂ G tq a' ∈[5] g
si g existe alors
    ajouter a comme valeur du rôle
    element à g
sinon
    chercher a' tq a R a'
    si a' existe
    créer g ⊂ G avec a et a' pour
    valeurs du rôle element
```

Sur l'exemple illustré par la figure 9, les plans grisés représentent des plans de la même classe L, par exemple une série de plans de plateau. Lorsque deux plans se chevauchent, la transition entre ces deux plans est graduelle. C'est le cas entre les plans 1 et 2, 2 et 3, 4 et 5, 5 et 6, alors qu'il y a un « cut » entre les plans 3 et 4.
Pour regrouper les plans de type L, on écrit la règle de type (2) :

$$L\ \{meets \vee overlaps\} \rightarrow GL$$

[4] ⊂ signifie « instance du concept »
[5] ∈ signifie ici « est valeur du rôle `temporal-element` de »

avec GL le concept définissant un groupe de plans L, concept héritant de TEMPORAL-SEQ.
On constate que si les plans sont créés dans l'ordre de leur numéro, l'algorithme indiqué agrège bien les plans en une seule instance de GL. Toutefois, si l'ordre de création est 1-6-2-3-5-4, les plans seront regroupés en deux instances de GL : 1-2-3-4 et 4-5-6. On constate ici que le plan n°4 peut être agrégé à deux groupes de plans distincts. Il convient dans ce cas d'agréger à nouveaux ces deux groupes.
L'étape (iv) de l'algorithme, pour les règles de type (2), doit donc prendre en compte le fait que lorsqu'une instance peut s'agréger à plusieurs groupes, ces groupes doivent être agrégés entre eux. Cela peut s'exprimer de la manière suivante :

```
extraire liste l des gi ⊂ G tq a' ∈ gi
si |l| = 0
    créer g ⊂ G et ajouter a et a' à g
si |l| = 1
    ajouter a à g1
si |l|>1
    ajouter a à g1
    agréger gi, 2≤i≤|l|, à g1
```

Agréger plusieurs instances de TEMPORAL-SEQ, comme il est indiqué à la dernière ligne de l'algorithme ci-dessus, peut ne pas être une opération triviale si d'autres rôles que `element` sont pourvus. Nous ne traiterons pas ce problème ici.

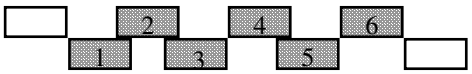

figure 9 : groupement de plans similaires

Enfin, un problème de monotonie se pose si les instances intervenant dans les règles de regroupement venaient à être modifiées. En effet, des propriétés peuvent être ajoutées aux instances ; ce serait le cas par exemple si plusieurs algorithmes de détection, ou bien une intervention manuelle, venaient successivement raffiner une description des plans.

2.5 Classification / Analyse

L'analyse telle qu'elle est présentée ici repose en dernier recours sur les algorithmes d'analyse. Il reste donc à préciser la manière dont vont interagir ces algorithmes et les processus de raisonnement présentés plus haut. La manière la plus simple consiste à exécuter tous les algorithmes d'analyse, puis à utiliser les résultats pour raisonner. Cependant, nous pouvons faire interagir plus étroitement analyse et raisonnement. Deux modes d'interaction entre analyse et raisonnement ont été identifiés. Le premier consiste pour un algorithme à créer des instances « primitives ». C'est le cas d'un algorithme de segmentation en plan, par exemple. Dans le second mode d'interaction, un algorithme spécialise une instance, par exemple en lui ajoutant des attributs. Cette instance peut alors à nouveau être classifiée. Ce second mode d'interaction est le plus complexe, car il fait intervenir des instances existantes. En utilisant ici encore le mécanisme de règles proposé par CLASSIC, il est possible de lier étroitement analyse de bas niveau et raisonnement.
Dans l'exemple du magazine de cinéma cité plus haut, une première étape consiste à exécuter un algorithme de segmentation en plans du document (premier mode d'interaction) des instances de la classe SHOT, concept primitif héritant de TEMPORAL. Dans une seconde étape, les plans d'une séquence d'extraits sont définis ainsi :

```
(define-concept 'TRAILER-SHOT '(and
    'SHOT
    (black-strip? begin-time end-time)))
```

black-strip? étant une fonction Lisp à deux paramètres calculant la présence de bandes noires dans le document entre deux dates. L'algorithme est donc déclenché automatiquement par le processus de classification (second mode d'interaction).
Les plans instances de TRAILER-SHOT sont alors regroupés par la règle de type (2) :
TRAILER-SHOT {*meets* ∨ *overlaps*} → TS-SEQ
avec TS-SEQ concept héritant de TEMPORAL-SEQ. Enfin, un algorithme d'extraction de texte peut être exécuté sur le début de chacune des séquences d'extraits instances de TS-SEQ.
Ainsi, un schéma général se dégage où alternent des phases de classification et de regroupement avec des phases d'analyse de bas niveau.

3. Discussion : reconnaissance de plan

Les règles de regroupement présentées ci-dessus constituent d'une certaine manière des *plans*, dans le sens de la reconnaissance de plan ou de scénario. Plusieurs travaux ont déjà été conduits qui utilisent le raisonnement terminologique dans le cadre de la reconnaissance de plan : par exemple, [18] propose d'étendre la notion de subsomption aux plans, et [19] définit un langage unifié de raisonnement terminologique et temporel. Dans le même ordre d'idées que [18], [20] propose d'organiser les plans en taxonomie en intégrant à un système de raisonnement terminologique (en l'occurrence CLASSIC) des techniques connues de planification utilisant des automates d'états finis. Une telle approche présente l'avantage d'offrir une formalisation claire du type de raisonnement mis en œuvre, ce qui permet d'avoir une bonne évaluation des types de problèmes pouvant être abordés ainsi que des limites du formalisme. En particulier, les plans qui peuvent être représentés dans [20] sont restreints à des expressions régulières, ce qui est trop peu expressif pour décrire le contenu de documents filmiques. Nous envisageons cependant de nous inspirer de cette démarche en formalisant d'avantage le raisonnement temporel à l'extérieur du système de représentation.

4. Expérimentations en cours

L'implémentation du système décrit dans ce papier est en cours. Nous utilisons la version 2.3 du système CLASSIC déjà mentionné dans l'environnement CLISP. Les principaux mécanismes de regroupement temporels ont été mis en œuvre ; des tests de plus grande envergure sur un

corpus documenté doivent être engagés. notamment dans le contexte du projet DIVAN.

Conclusion

L'indexation de documents filmiques est un sujet de recherche dont les enjeux sont considérables. La nature de la diffusion étant en train de changer considérablement, il sera de plus en plus important de pouvoir accéder à des bases de documents gigantesques, posant ainsi le problème de l'annotation dans un cadre nouveau. Nous pensons que la combinaison d'expertises de champs complémentaires est nécessaire pour obtenir des systèmes efficaces et opérationnels. Ceci nécessite d'une part la formalisation des divers types de documents existants, ce qui permet de restreindre le champ de l'analyse, et d'autre part la collaboration des techniques d'analyse « bas niveau » du traitement d'image et de l'analyse de signal d'une part et des techniques de représentation et de raisonnement de plus haut niveau d'autre part. Nous proposons dans ce papier un élément de réponse fondé sur l'exploitation du formalisme des logiques de descriptions qui nous paraît prometteuse.

Bibliographie

1. Poncin, P., *Audiovisuel : vers le tout numérique.* Les dossiers de l'audiovisuel, 1997. **74**.
2. Pinhanez, C.S., Bobick, A.F. *Approximate World Models: Incorporating Qualitative and Linguistic Information into Vision Systems*. in *AAAI 96*. 1996.
3. Herzog, G. *Utilizing Interval-Based Event Representations for Incremental High-Level Scene Analysis*. in *Proc. of the 4th International Workshop on Semantics of Time, Space, and Movement and Spatio-Temporal Reasoning*. 1992. Château de Bonas, France.
4. Katz, S.D., *Film Directing Shot by Shot*. 1991: Michael Wiese Production.
5. INA, *Glossaire de l'image et du son*, 1997, Collection techniques et production audiovisuelles, Institut National de l'Audiovisuel.
6. Yeo, B.-L., Liu, B., *Rapid Scene Analysis on Compressed Video.* IEEE Transactions on Circuits and Systems for Video Technology, 1995. **5**(6): p. 533-544.
7. Aigrain, P., Joly, P., Longueville, V., *Medium Knowledge-Based Macro-Segmentation of Video into Sequences*, in *Intelligent Multimedia Information Retrieval*, A.P.M. Press, Editor. 1997.
8. Merlino, A., Morey, D., Maybury, M. *Broadcast News Navigation using Story Segmentation*. in *Proceedings of the Fifth ACM International Conference*. 1997. Seattle, Washington, USA.
9. Özsu, M.T., Szafron, D., El-Medani, G., Vittal, C., *An Object-Oriented Multimedia Database for a News-on-Demand Application.* Multimedia Systems, 1995. **3**(5-6): p. 182-1995.
10. Ronfard, R. *Shot-level description and matching of video content*. in *SPIE 97*. 1997. San Diego.
11. Napoli, A., *Une introduction aux logiques de description*, 1997, Rapport de recherche n 3314, INRIA.
12. Borgida, A., Brachman, R.J., McGuiness, D.L., Resnick, L.A. *CLASSIC: A Structural Data Model for Objects*. in *ACM SIGMOD Int. Conf. on Management of Data*. 1989.
13. Nebel, B., *Reasoning and Revision in Hybrid Representation Systems.* LNAI, 1990. **422**.
14. Pinhanez, C., Bobick, A., *PNF Calculus : A Representation and a Fast Algorithm for Recognition of Temporal Structure*, 1996, Technical Report n 389, MIT Media Lab Perceptual Computing Section.
15. Allen, J.F., *Towards a general theory of action and time.* Artificial Intelligence, 1984. **23**(2): p. 123-154.
16. van Beek, P. *Reasoning about qualitative temporal information*. in *Proceedings of AAAI'90*. 1990.
17. Nebel, B., Bückert, H.J. *Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra*. in *Proceedings of AAAI'94*. 1994. Seattle, Washington.
18. Weida, R., Litman, D. *Terminological Reasoning with Constraint Networks and an Application to Plan Recognition*. in *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*. 1992. Cambridge, Massachussetts.
19. Artale, A., Franconi, E. *A Computational Account for Description Logic of Time and Action*. in *Proc of the 4th International Conference on Principles in Knowledge Representation and Reasoning (KR94)*. 1994.
20. Devanbu, P.T., Litman, D., *Taxonomic Plan Reasoning.* Artificial Intelligence, 1996. **84**: p. 1-35.