Localisation et identification de personnes dans les séquences vidéo

Stéphane Marchand-Maillet et Bernard Mérialdo
Institut EURÉCOM – Sophia-Antipolis
{marchand,merialdo}@eurecom.fr

Résumé: Cet article présente une étude dans le contexte de l'indexation de séquences vidéo. Nous nous attachons à localiser et identifier des personnes dans ce type de données. La localisation est basée sur l'utilisation de modèles de Markov cachés (HMM) uni- et bidimensionnels. L'identification exploite l'aspect temporel des données vidéos pour compenser la difficulté induite par la grande variabilité d'exposition et de contexte dans lesquels l'identification doit être effectuée.

1 Introduction

L'abondance des documents vidéos a rendu nécessaire le développement de techniques d'indexation automatiques. Dans ce travail, nous étudions l'analyse de scènes comprenant des personnes (fiction, journaux télévisés, etc). Une telle analyse est essentielle pour atteindre une description complète du document en question. Par exemple, la localisation de séquences montrant le présentateur d'un journal télévisé permettra de pouvoir segmenter ce document en parties traitant de différents sujets. Pour effectuer la localisation d'une personne dans une image, nous allons nous attacher à localiser son visage dans l'image. Nous utilisons actuellement la base de données de séquences vidéo fournie par l'INA (GT - Indexation Multimédia).

D'une manière générale, l'enchaînement des tâches que nous nous proposons de détailler dans ce travail est le suivant:
1. Détection des images d'une séquence MPEG contenant des personnes.
2. Localisation du visage de la personne dans une telle image.
3. Identification de cette personne.

Dans notre contexte, la détection d'images contenant des personnes est combinée avec la localisation de la personne dans l'image. La Section 2 présente la technique utilisée pour la localisation basée sur les modèles de Markov cachés (HMM). Les résultats de la phase de localisation sont utilisés en Section 3 dans un processus de classification en vue d'identifier la personne présente dans l'image. Finalement, la Section 4 tire les conclusions de ce travail préliminaire.

2 Localisation basée sur les modèles de Markov cachés

Le principe général des modèles de Markov cachés permet d'associer une structure (dite "cachée") à une séquence d'observations. Ces modèles sont surtout utilisés pour la reconnaissance de parole car leur flexibilité permet de modéliser la déformation

(temporelle) de ce type de données. D'une façon similaire, les modèles de Markov ont été utilisés pour l'analyse de caractères manuscrits car ils offrent une grande robustesse contre les déformations (spatiales) des caractères.

Nous proposons d'utiliser les modèles de Markov cachés pour la localisation de visages dans les séquences vidéos. L'image est assimilée à une séquence d'observations dont l'exemple les plus simple est la couleur de chaque pixel. Dans notre modèle unidimensionnel, chaque ligne est associée à un modèle, indépendemment des autres lignes. Dans le modèle (pseudo-) bidimensionnel, chaque ligne est associée à un modèle unidimensionnel (horizontal) et la séquence de ces modèles est associée à un modèle (vertical). Ceci permet de prendre en compte à moindres frais la structure bidimensionnelle de l'image.

2.1 Modèle unidimensionnel

Nous présentons cette étude nous basant sur un exemple simple. Deux types de pixels sont discernés dans une image contenant un visage: les pixels de type Fond et les pixels de type Visage. Le visage étant une partie connexe, chaque ligne sera formée soit d'une séquence de pixels Fond soit d'une alternance de pixels Fond-Visage-Fond. Ceci détermine la structure des deux modèles de Markov λ_1 et λ_2 que l'on va utiliser pour modéliser la génération des lignes de l'image.

Une ligne de pixels O est vue comme une séquence d'observations dans le temps ($O = \{o_1, o_2, \cdots, o_t, \cdots, o_t\}$ où o_t représente l'information contenue dans un pixel). Un modèle de Markov se compose d'un ensemble d'états s_i représentant des entités capables de générer des observations o_t à chaque instant t avec une certaine probabilité d'émission $b_i(o_t)$. Le modèle de Markov caché va associer à une séquence d'observations O donnée une ou plusieurs séquences (cachées) d'états,

$$Q = \{q_1, q_2, \cdots, q_t, \cdots, q_t\} \quad ; \quad q_t = s_i$$

Le calcul des paramètres du modèle de Markov caché se fait grâce à un entraînement itératif dont le but est de maximiser la probabilité de création d'une séquence d'observation O donnée par un modèle donné λ, donc de maximiser $P[O|\lambda]$. Deux types d'entraînement sont possibles, ils sont tout deux basés sur un principe de programmation dynamique. La procédure de Viterbi assure l'optimisation en se basant sur la plus probable séquence d'états par laquelle O a été générée. La procédure de Baum-Welsh maximise la somme des probabilités des séquences permettant de générer O. Dans les deux cas, cette optimisation permet le calcul des paramètres du modèle utilisé et donc de déterminer pour une nouvelle séquence d'observation O la probabilité d'être dans un état s_i donné pour générer une observation o_t donnée. Donc de retrouver la structure (cachée) de cette séquence d'observations. Avec les notations introduites ci-dessus, on calcule pour chaque temps t et pour chaque état s_i, $P[q_t = s_i|\lambda]$. La sélection de la séquence d'états Q^* la plus probable permet d'associer O avec sa structure cachée la plus probable.

Dans l'exemple qui nous intéresse, on définira deux états: s_1 =Fond et s_2 =Visage et deux modèles de Markov cachés correspondant à des lignes contenant uniquement des pixels Fond et des lignes suivant la séquence Fond-Visage-Fond. Ces deux modèles sont présentés en Figure 1

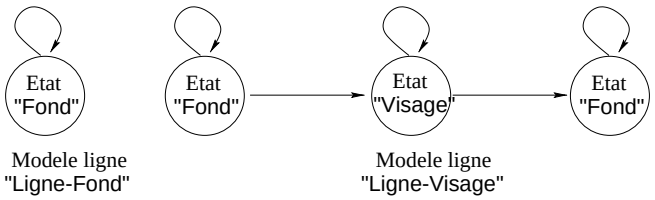

Figure 1: Structures des modèles de Markov cachés utilisés

2.2 Modèle (pseudo-)bidimensionnel

L'extension du modèle ci-dessus se fait en réduisant l'indépendance de lignes entre elles. Dans un vrai modèle bidimensionnel, la dépendance bidimensionnelle se traduirait au niveau du pixel. Toutefois, il a été montré que ce type de modèle mène à une complexité de calcul exponentielle et peut être raisonnablement bien approximé par les modèles pseudo-bidimensionnels que nous présentons ici [2, 3]. Pour introduire ces modèles, nous présentons cette extension en nous basant sur le même exemple simple.

La structure bidimensionnelle est introduite en considérant que le fait d'utiliser un modèle de Markov (1D) $S_j = \lambda_j$ pour modéliser une ligne correspond au fait d'être dans un super-état particulier ($Q_y = S_j$). Dans notre exemple, deux super-états sont utilisés. Une ligne contenant du fond uniquement sera associée au super-état S_1 =Ligne-Fond et une ligne contenant une partie de visage sera associée au super-état S_2 =Ligne-Visage. Le modèle de Markov pseudo-bidimensionnel Λ aura donc la structure décrite en Figure 2.

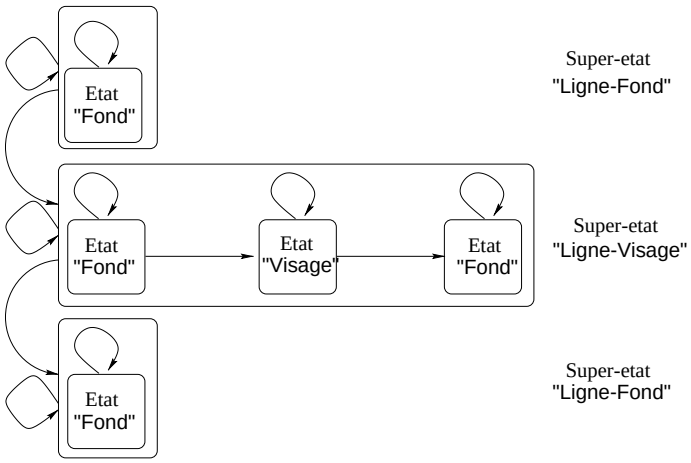

Figure 2: Structures du modèle de Markov caché pseudo-bidimensionnel utilisé.

Une procédure de deux entraînements imbriqués permet la maximisation de $P[q_{xy} = s_i^j|\Lambda]$ où s_i^j est l'état i du super-état S_j (implicitement, $Q_y = S_j$) et q_{xy} est l'état au pixel o_{xy}.

2.3 Résultats

L'évaluation de la localisation se fait en segmentant l'image de test grâce au résultat de la localisation. Dans la Figure 3, on atténue la couleur des pixels dont l'état correspondant le plus probable est Fond. Pour le modèle unidimensionnel, l'image est analysée successivement dans deux orientations, les séquences d'observations sont successivement les lignes et les colonnes de pixels. Le résultat est l'ensemble de pixels correspondant à l'état Visage dans les deux orientations (intersection). Ceci permet de réduire les erreurs générées par l'indépendance de lignes dans le modèle. L'étude des valeurs $P[O|\lambda]$ dans les deux orientations montrées en Figure 3 permet aussi d'estimer une décomposition de l'image. Dans cet exemple, on peut distinguer l'influence des yeux et de la bouche (graphe des lignes) et aussi de l'oreille (graphe des colonnes).

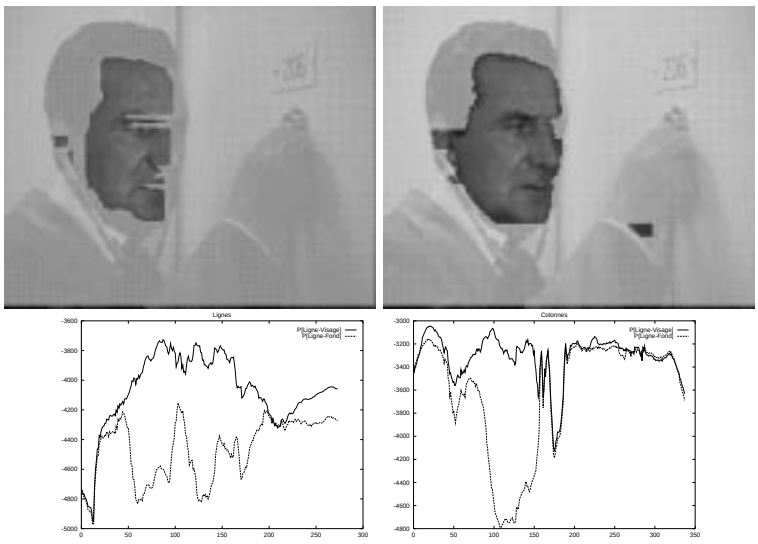

Figure 3: Haut: Résultats de localisation. Gauche: Modèle unidimensionnel. Droite: Modèle bidimensionnel. Bas: probabilités des modèles unidimensionnels. Gauche: Lignes. Droite: colonnes.

Le modèle bidimensionnel permet de formaliser cette approche en générant une dépendance entre lignes. L'analyse ne se fait donc que dans une orientation. On constate que l'efficacité de la localisation est améliorée par cette extension. Les performances de localisation pourront être améliorée en considérant plus d'information pertinente à chaque pixel (et donc décrivant un état particulier). Par exemple, on pourra prendre en compte les courbures locales, base de l'index de forme (Shape Index [1]).

Du fait de ce modèle simple, on constate aussi en général qu'à haute résolution, trop de caractéristiques du visage sont présentes. Un lissage de ces caractéristiques fines (ou une baisse de résolution) permet aussi d'améliorer cette localisation "grossière". A partir de ce modèle simple, on peut envisager une segmentation plus fine du visage en compliquant les modèles utilisés (en insérant par exemple des états Oeil, Bouche ou Nez).

3 Identification de visages dans des séquences vidéo

Le but est d'identifier automatiquement les différentes personnes apparaissant dans une vidéo, ainsi que les instants où elles apparaissent. Les techniques classiques de reconnaissance de visage s'appliquent en général à des images isolées, mais lorsqu'on traite des séquences vidéo, une information temporelle vient se rajouter qu'il est intéressant d'exploiter dans le processus de reconnaissance, puisque chaque visage apparaîtra toujours dans plusieurs images consécutives. L'opération de base n'est donc plus de comparer seulement des images de visages, mais plutôt des suites d'images.

3.1 Pré-traitement vidéo

Des traitements classiques permettent de découper une séquence vidéo en plans consécutifs, par détection des coupures séparant les plans. Chaque image d'un plan est analysée pour localiser et détecter des visages. Un simple algorithme de suivi permet d'identifier, à l'intérieur d'un même plan, comment se déplace le visage d'une personne. On peut donc automatiquement extraire d'une séquence vidéo des suites d'images de visages correspondant chacune à une seule personne. A l'intérieur d'une suite, le visage pourra avoir des orientations différentes en fonction des mouvements que cette personne effectue à l'intérieur du plan.

3.2 Identification

Deux variantes peuvent se présenter :

- Si l'on dispose d'une base des visages connus, il suffit de comparer chaque suite à cette base pour identifier les différentes personnes présentes. L'opération de base consiste alors à comparer une suite de visages à un modèle de visage provenant de la base.
- Si l'on ne dispose pas d'une telle base, il faut comparer les séquences entre elles pour déterminer celles qui représentent la même personne. On a alors besoin de définir une distance entre séquences. Nous intéressons à la deuxième approche, pour laquelle nous utilisons une classification hiérarchique ascendante. Au départ, chaque suite est placée dans une classe différente. A chaque itération de la classification, on fusionne les deux classes les plus semblables, jusqu'à ce qu'un critère de fin soit satisfait.

Ce procédé fait intervenir deux éléments-clés :

- Une mesure de similarité entre ensembles de suites de visages. C'est le point le plus délicat, pour lequel une approche naturelle consiste à utiliser une version ensembliste de mesures utilisées en reconnaissance de visages.
- Le critère de fin permettant de terminer la classification, lorsque (idéalement) les suites correspondant aux mêmes personnes ont toutes été classifiées ensemble.

Enfin, un troisième point est de mesurer la qualité de l'identification, lorsque celle-ci n'est pas parfaite.

3.3 Expérimentations

Dans cette partie, nous décrivons quelques premières expériences mettant en oeuvre la méthodologie d'identification précédemment exposée. Nous utilisons la séquence vidéo AIM1MB08 de la base de données distribuée par l'INA dans le cadre du Groupe de Travail "Indexation Multimédia". Cette séquence représente un épisode de 50 Min de la fiction "Chapeau Melon et Bottes de cuir", codé au format MPEG1. Les traitements de détection et de localisation de visages produisent 71 suites de visages, dont la longueur varie de 2 à 30 images, pour un total de 640 visages représentant 13 personnes différentes.

A partir d'une mesure de similarité entre visages $d(v_1, v_2)$, nous avons considéré deux mesures possibles entre ensembles de visages :

$$D_1(S_1, S_2) = \min\{d(v_1, v_2) | v_1 \in S_1, v_2 \in S_2\}$$

$$D_2(S_1, S_2) = \text{avg}\{d(v_1, v_2) | v_1 \in S_1, v_2 \in S_2\}$$

Idéalement, on souhaite utiliser une mesure $d(v_1, v_2)$ telle que deux visages correspondant à la même personne soient très proches, même s'ils correspondent à des conditions d'éclairage ou d'orientation différentes, alors que deux visages de deux personnes différentes devraient correspondre à une valeur élevée. La mesure D_1 correspond à faire l'hypothèse que parmi les variations présentes dans S_1 et S_2, celles qui sont les plus proches sont les plus significatives pour savoir si S_1 et S_2 représentent la même personne. Par exemple, si S_1 et S_2 contiennent de nombreuses orientations différentes, on peut penser que le minimum sera atteint lorsqu'on compare des orientations très proches, s'il en existe. La mesure D_2 consiste simplement à lisser les indications fournies par les comparaisons de visages. La Figure 4 montre un exemple de matrice des distances de visages, où l'on peut remarquer le long de la diagonale les blocs correspondant aux images de la même suite.

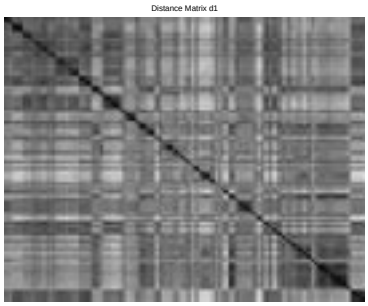

Figure 4: Matrice de distance inter-visages

Nous avons expérimenté plusieurs mesures simples pour la distance $d(.,.)$:

1. La norme L^1 entre deux visages,

2. La norme L^1 entre deux visages sans tenir compte du fond,

3. La norme L^1 du meilleur alignement (ligne et colonne) entre les deux images (obtenu par programmation dynamique dans le but de réduire la dépendance de l'orientation).

Pour chacune de ces mesures, nous avons réalisé une classification hiérarchique complète, jusqu'à ce que toutes les suites soient regroupées dans la même classe. L'évaluation d'une classification s'effectue en comptant le nombre minimum de modifications nécessaire pour obtenir la classification parfaite (où chaque modification consiste à mettre une séquence dans une autre classe). Le graphe obtenu pour les trois mesures considérées est présenté dans la Figure 5. On peut remarquer que l'alignement améliore la qualité de la classification, mais que dans tous les cas, le minimum n'est pas obtenu pour un nombre de classes égal au nombre de personnes (mais sensiblement supérieur). Sur la partie droite de la figure se trouve le graphe des valeurs de la mesure de similarité des classes les plus proches à chaque itération. On fusionne des classes qui sont de moins en moins semblables, mais malheureusement, le graphe ne semble pas donner d'information significative sur l'instant où l'on fusionne des séquences correspondant à des personnes différentes. Le problème du critère d'arrêt de la classification est donc difficile.

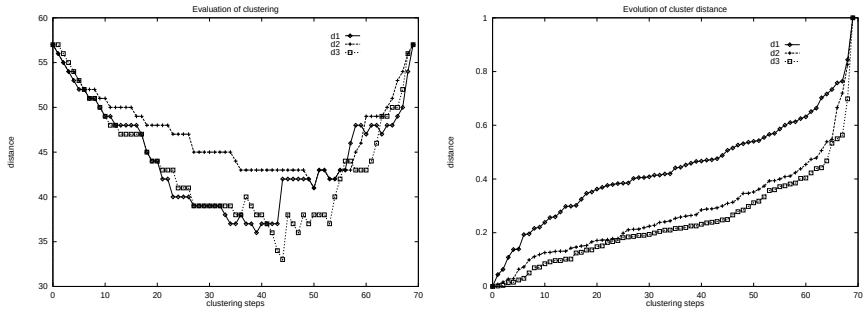

Figure 5: Evolution du processus de classification.

3.4 Évaluation de la classification

Chaque séquence de visage correspond à une personne. On recherche donc à retrouver la classification des séquences obtenue en mettant dans la même classe les séquences de la même personne, comme dans l'exemple suivant :
P_1 : S_1, S_3, S_6
P_2 : S_2, S_7
P_3 : S_4, S_5
Pour évaluer une classification, il faut comparer cette classification à la classification idéale. Malheureusement, lorsqu'on classifie les séquences de façon automatique, on obtient des classes dont on ne peut savoir a priori à quelle personne elles correspondent. La classification peut ne pas avoir le bon nombre de classes, puisque c'est le nombre de personnes différentes, qui n'est pas connu. L'exemple suivant donne un exemple de classification à comparer :

C_1 : S_3, S_4, S_6
C_2 : S_2
C_3 : S_1
C_4 : S_5, S_7
Dans cet exemple, la classe C_1 pourrait correspondre à P_1, mais il n'est pas clair de voir laquelle de C_2 ou C_4 correspond le mieux à P_2.

Pour faire une comparaison de façon rigoureuse, on considère des mouvements élémentaires consistant à faire passer une séquence d'une classe à une autre. La distance entre deux classifications est définie comme le nombre minimum de mouvements permettant de transformer la première classification en la seconde (mais le processus est symétrique). Notons que dans ce mécanisme, un mouvement élémentaire peut supprimer une classe (en enlevant son seul élément pour le mettre ailleurs), ou bien créer de nouvelles classes (en y prenant un élément quelque part pour le mettre dans cette nouvelle classe). Les noms des classes n'ont pas de signification particulière dans une classification automatique, donc deux classifications sont considérées comme identiques si elles correspondent à la même partition de l'ensemble des séquences. Un algorithme de programmation dynamique est utilisé pour calculer la distance entre deux classifications.

4 Conclusion

Ce travail se place dans le cadre de l'indexation automatique de séquences vidéo. Dans cet article, nous avons montré les potentialités des modèles de Markov cachés pour réaliser la localisation de personnes dans des images de type vidéo. L'identification est rendue difficile par les variations engendrées par ce type de données. Dans cette étude, nous nous sommes intéressés à un processus de classification entre séquences plutôt qu'une comparaison directe entre image, ceci afin d'exploiter la dimension temporelle des données video. Toutefois, cette classification est basée sur une distance et cette étude suggère que des distances plus élaborées que des distance de type L^1 sont nécessaires. A moins de réaliser une phase de normalisation envisageable grâce aux modèles de Markov qui peuvent permettre, par exemple, de supprimer l'effet du fond de l'image en réalisant une segmentation.

References

[1] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10, 2 1992.

[2] S.-S. Kuo and O. E. Agazzi. Keyword spotting in poorly printed documents using Pseudo 2-D Hidden Markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):842–848, 1994.

[3] E. Levin and R. Pieraccini. Dynamic planar warping for optical character recognition. In *Proceedings of ICASSP'92*, volume III, pages 149–152, 1992.