Détection insensible au mouvement des frontières de plans franches et graduelles dans les séquences vidéo numériques

S. Lawrence [1] M.-F. Auclair-Fortier[1,2] D. Ziou[1] A. Beghdadi[2]

[1] Lab. de Moivre - DMI - U. de Sherbrooke
[2] L2TI -Institut Galilée - U. Paris 13

2500 Blv Université
Sherbrooke, Canada, J1K 2R1
auclair@dmi.usherb.ca

Résumé

Dans cet article, nous présentons une nouvelle méthode pour identifier les frontières de plans dans les vidéos numériques, qu'elles soient franches ou graduelles. Puisque ces frontières correspondent à des transitions dans le domaine temporel, nous proposons une méthode basée sur les dérivées partielles de la vidéo. Grâce à une caractérisation du mouvement apparent, notre méthode est capable de réduire les effets du mouvement afin d'améliorer les résultats de la détection. Pour accroître la performance de l'algorithme, nous considérons une technique d'échantillonnage spatial des pixels. Nos résultats expérimentaux montrent que notre méthode réussit mieux que d'autres méthodes populaires.

Mots Clef

Détection des frontières de plans vidéo, Détection des coupures franches, Détection des transitions graduelles, Dérivées partielles, Flot optique

1 Introduction

Au cours de la dernière décennie, le nombre de documents vidéos numériques a augmenté d'une façon telle que ce média est maintenant une composante dominante des données multimédias. Afin de retrouver facilement l'information recherchée, une description adéquate du contenu de la vidéo s'impose. L'indexation textuelle ne suffit plus pour remplir cette tâche. Plusieurs auteurs [4, 5, 8, 11, 12] croient qu'une étape essentielle à la gestion efficace des séquences vidéos et la compression MPEG est la détection des frontières de plans.

Nous définissons le problème de segmentation vidéo comme la détection automatique des frontières qui séparent les plans. Une vidéo peut être divisée en plus petites composantes appellées plans qui sont des suites d'images qui proviennent d'une seule opération d'enregistrement d'une unique caméra et qui représentent chacun une action continue dans l'espace et le temps. Au cours du montage, les plans peuvent être combinés

FIG. 1 – *Types de frontières de plans : coupure franche, fondu enchaîné et volet.*

de plusieurs façons pour produire la vidéo finale. Les frontières diffèrent en fonction de la technique utilisée pour combiner deux plans. Le type le plus simple est la coupure franche et résulte d'une simple juxtaposition de deux plans. D'autres types de frontières telles les fondus et les volets résultent de l'addition d'images entre les plans pour produire des effets de transitions plus graduels. Un fondu en ouverture fait progressivement apparaître un plan à partir d'une image noire. À l'inverse, un fondu en fermeture fait progressivement disparaître un plan vers une image constante. Un fondu enchaîné est une combinaison d'un fondu en ouverture et d'un fondu en fermeture. Un volet est une transition graduelle dans laquelle une image semble en pousser une autre en dehors de l'écran. Des exemples de quelques types de frontières sont donnés dans la figure 1.

Même si les coupures franches représentent la majorité des frontières de plans, nous croyons que détecter les frontières graduelles est aussi important. Par exemple, dans un échantillon de données vidéo composé de commerciaux, 25% des frontières de plans étaient graduelles [2]. La détection des coupes franches est un problème bien établi dans les travaux précédents mais celle des transitions graduelles demeure encore un défi parce que celles-ci s'effectuent sur plusieurs trames et donc les différences entre deux trames sont très petites. Conséquemment, la détection de ces coupures devrait s'effectuer sur plusieurs trames. Le fait que le

mouvement cause le même type de changement que les transitions graduelles contribue aussi fortement à la difficulté de les détecter. Pour cette raison, les pixels ayant une forte composante de mouvement devraient être éliminés de la détection.

Pour tenter de surmonter ces difficultés, nous proposons de trouver les frontières en identifiant les maxima locaux d'une mesure de différence sur plusieurs images, basée sur les dérivées partielles temporelles. Nous seuillons ces maxima pour en éliminer les doublons, ceux dus au bruit. De plus, nous enlevons les pixels ayant une forte composante de mouvement. Pour ce faire, nous examinons l'équation du flot optique de laquelle nous dérivons un seuil pour éliminer ces pixels du calcul de la mesure de différence. Enfin, pour augmenter la performance, un échantillonnage est utilisé avec peu d'effets sur les résultats. La section 2 présente un aperçu des travaux dans ce domaine. La section 3 présente notre méthode basée sur les discontinuités temporelles et l'équation du flot optique. La section 4 présente des résultats comparatifs avec deux autres méthodes.

2 Travaux existants

Plusieurs méthodes ont été proposées pour la segmentation automatique des vidéos. La plupart d'entre-elles définissent une mesure de différence, locale ou globale, entre des trames consécutives en vue d'identifier les frontières de plans. Ces méthodes sont développées pour des vidéos compressées ou non. Des revues plus détaillées sont disponibles dans [2, 4].

Les méthodes peuvent se distinguer par quatre caractéristiques. 1) Les données et le pré-traitement : dans le cas des vidéos non-compressées, les algorithmes sont appliqués directement sur l'information pixel, soit niveau de gris [9, 10], soit couleur [5, 8, 12]. Certains pré-traitements sont aussi appliqués tels, une détection de contours [11], le calcul du flot optique [3] ou un échantillonnage spatial et/ou temporel [1] pour réduire le temps de calcul ou l'effet du mouvement. Notre algorithme effectue un échantillonnage spatial, mais n'utilise aucun autre pré-traitement, ce qui est un avantage en temps de calcul. 2) La mesure de différence utilisée pour estimer le changement entre deux trames : les plus simples mesures sont basées sur la différence des intensités entre des trames consécutives [1, 9, 10, 12]. Une autre classe d'algorithme est basée sur les différences d'histogrammes de trames consécutives [8, 9]. Certains utilisent une méthode basée sur des modèles de frontières [5]. Zabih *et al.* [11] présentent une méthode basée sur le suivi de contours binaires entre trames consécutives. Cheong [3] calcule, sur le nombre de pixels ayant une forte composante de mouvement, le pourcentage de ceux qui violent la contrainte de lissage du flot optique. Notre méthode est un mélange de méthodes basées sur le flot optique, sans toutefois le calculer explicitement, et les différences d'intensités. 3) Le seuillage : une fois la mesure de différence calculée, la plupart des algorithmes comprennent une étape de seuillage pour distinguer les frontières du mouvement et du bruit. Ces seuils sont simples [8, 9, 11] ou plus complexes [1, 3, 12]. 4) Les types de frontières détectées : plusieurs auteurs [8, 9, 10] se concentrent sur les coupures franches et ignorent les transitions graduelles. Notre méthode se situe plutôt dans celles qui tentent de détecter tous les types de frontières [5, 11].

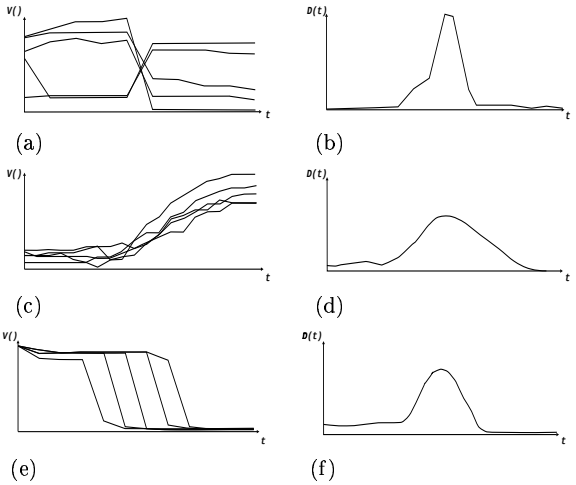

FIG. 2 – *a), c), e) Profils temporels typiques (5 pixels) pour une coupure franche, un fondu enchaîné et un volet respectivement. b), d), f) $D(t)$ associés.*

3 Approche proposée

Notre objectif est de développer une méthode capable de détecter de façon adéquate les coupures franches et les transitions graduelles. Considérons une séquence vidéo comme une fonction à trois variables discrètes $V(x,y,t)$. Notre méthode de détection est basée sur les dérivées partielles de premier ordre de cette fonction. Notre hypothèse initiale est que les frontières de plan correspondent aux discontinuités spatiales. Les figures 2(a), 2(c) et 2(e) montrent des profils temporels pour quelques types de frontières, confirmant notre hypothèse.

Nous pouvons remarquer qu'une coupure ou un volet produit un contour de type marche dans la fonction $V(x_i,y_j,t)$ ce qui produit un maximum dans $|V_t(x_i,y_j,t)|$. Lors d'un fondu, l'intensité du pixel change lentement et produit un contour flou de type marche (sur plusieurs trames). Si la dérivée temporelle est calculée sur un assez large intervalle, le contour produit un maximum dans $|V_t(x_i,y_j,t)|$.

Du paragraphe précédent, il est clair que la mesure de différence doit être une fonction de $|V_t(x_i,y_j,t)|$. Elle doit aussi être globale et tenir compte d'un large nombre de pixels pour réduire le bruit et l'effet du mouvement. Nous proposons la mesure de différence suivante :

$$D(t) = \sum_{(x,y)} |V_t(x,y,t)|. \qquad (1)$$

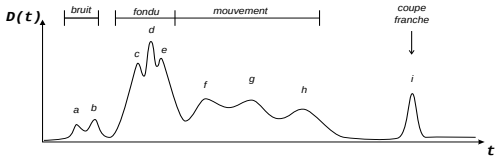

FIG. 3 – *Situations de seuillage.*

Le calcul des dérivées temporelles $V_t(x,y,t)$ est réalisé par une convolution de $V(x,y,t)$ avec un masque Gaussien, où l'échelle du filtre Gaussien permet de considérer plus de deux trames et donc est mieux adapté à l'identification des transitions graduelles. Les figures 2(b), 2(d) et 2(f) montrent les $D(t)$ associés à certains types courants de frontières de plans.

Nous proposons de localiser les frontières de plans par l'identification des maxima locaux dans la mesure de différence $D(t)$. Cependant, ce ne sont pas tous les maxima locaux dans $D(t)$ qui sont dus aux frontières de plans (figure 3). Premièrement, pour éviter les détections multiples (ex. étiquettes c, d et e), seulement les maxima locaux sur une taille de fenêtre fixée sont gardés(ex. d est gardé). Deuxièmement, deux seuils sont appliqués. τ_1 a pour but d'enlever les petits pics dus au bruit (ex. a et b) et est un pourcentage du maximum global de $D(t)$. τ_2 vise à éliminer les maxima locaux correspondant aux pics de relative basse amplitude (ex. f, g et h), qui ne sont pas causés par une frontière de plan, mais plutôt par le mouvement. Cette amplitude relative est définie comme la moyenne des différences entre le pic et les minima précédents et suivants. τ_2 est un pourcentage de la valeur du pic. En dépit de ces règles de seuillage, quelques maxima dus au mouvement demeurent (figure 4, pics a et b). La section suivante présente une nouvelle approche pour réduire ces effets du mouvement.

3.1 Réduction des effets du mouvement

Le mouvement affecte la détection des frontières de plans dans les régions où l'intensité des pixels change à cause du mouvement plutôt qu'à cause du montage. $D(t)$ n'est donc pas suffisante pour enlever les effets du mouvement en dépit des règles de seuillage. Nous proposons donc d'identifier les zones où le mouvement cause des problèmes, et d'estimer $D(t)$ seulement dans les endroits qui ne contiennent pas de flot optique. Dans ces régions, les dérivées spatiales peuvent être utilisées pour prévenir ces effets. De l'équation du flot optique ($I_x u + I_y v + I_t = 0$, où (u,v) est le vecteur de mouvement) [6], il est clair que s'il y a mouvement, au moins une des dérivées partielles spatiales doit être différente de zéro. Nous proposons donc d'évaluer les deux dérivées partielles $V_x(x,y,t)$ et $V_y(x,y,t)$ et éliminer du calcul de $D(t)$ les pixels où une des deux dérivées est importante. Cela assure que les valeurs importantes de $|V_t(x,y,t)|$ résultantes du mouvement, ne sont pas prises en compte dans le calcul de $D(t)$. Les pixels à forte composante de mouvement sont trouvés en seuillant l'angle entre le vecteur (V_x,V_y,V_t) et le vecteur $(0,0,V_t)$. L'éq. 1 devient donc :

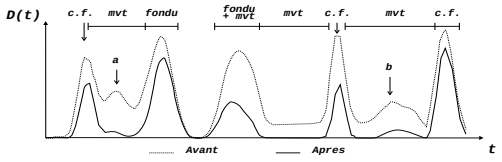

FIG. 4 – *Réduction des effets du mouvement.*

$$D(t) = \sum_{(x,y)} \begin{cases} |V_t| & \text{si } \tan\theta < \tan\tau_{mvt} \\ 0 & \text{ailleurs} \end{cases}, \quad (2)$$

où τ_{mvt} est l'angle maximum toléré entre les deux vecteurs et

$$\tan\theta = \left(\sqrt{V_x^2 + V_y^2}\right) / |V_t|,$$

ce qui peut être facilement vérifié. La figure 4 montre la réduction des effets du mouvement sur la détection des frontières de plans. Deux maxima locaux (a et b) dus aux mouvements ont été presque totalement enlevés.

Nous résumons l'algorithme de la manière suivante. 1) Convolution de chaque trame de la vidéo avec les dérivées de premier ordre de la Gaussienne et estimation de $D(t)$ (éq. 2). 2) Localisation des maxima locaux de $D(t)$ et application des règles de seuillage. 3) Échantillonnage spatial à intervalles réguliers pour réduire le traitement car l'algorithme est gourmand en temps. Le nombre de pixels sur lesquels les dérivées sont évaluées est réduit. Chaque x*ième* pixel est gardé dans chaque direction (ratio $1:x$). L'augmentation du ratio implique une réduction de la précision. Un choix adéquat de l'échantillonnage permet de garder une bonne précision tout en faisant des gains considérables au niveau du temps de calcul.

4 Résultats expérimentaux

La performance de l'algorithme a été testée, en comparaison avec deux autres méthodes (histogrammes de régions [8] et comparaison des contours [11]. Pour chaque méthode, le nombre de frontières correctement détectées, manquées et fausses positives sont calculées. Dans le cas des détections multiples, seulement une détection est considérée correcte, les autres étant considérées fausses positives. Nous utilisons les graphes de *Rappel* vs *Précision* pour comparer les méthodes. La mesure de *Rappel* est définie comme le pourcentage des éléments désirés qui ont été trouvés, alors que la mesure de *Précision* est définie comme le pourcentage des éléments trouvés qui étaient désirés :

$$Rappel = \frac{Correct}{Correct + Manqué}, \quad Précision = \frac{Correct}{Correct + Faux}$$

Chaque algorithme a été testé avec plusieurs combinaisons de paramètres de seuil, donc plusieurs valeurs de *Précision* ont été générées pour chaque valeur de *Rappel*. Le graphe montre donc la meilleure valeur de *Précision* pour chaque valeur de *Rappel*.

La séquence vidéo utilisée pour la comparaison est extraite d'une bande annonce du film *Les Misérables* (1715 trames, 160×120 pixels/trame, 15 trames/sec., 87 frontières de plans dont 60 franches et 27 graduelles) et contient un large nombre d'objets et de mouvements. D'autres séquences vidéos ont été utilisées [7] mais ne sont pas présentées dans cet article.

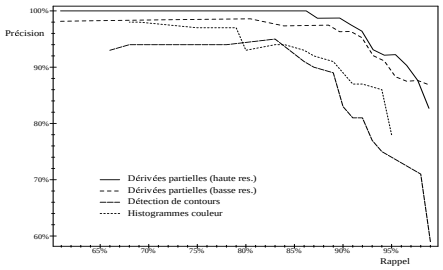

FIG. 5 – *Meilleure Précision par Rappel.*

Histogramme	60 sec. (28.6 t/s)
Suivi de contours	2000 sec. (0.86 t/s)
3D Dériv. (haute rés.)	3200 sec. (0.54 t/s)
3D Deriv. (basse rés.)	700 sec. (2.45 t/s)

TAB. 1 – *Temps de calculs.*

Pour la méthode des histogrammes locaux (64 n.g.), chaque image est divisée en 16 régions de 4×4 pixels. Les 8 régions avec les plus grandes différences sont éliminées pour réduire les effets du mouvement. La technique à deux seuils de Zhang *et al.* [12] est utilisée pour identifier les transitions graduelles. Les seuils varient entre 1000 et 40000.

Pour la méthode de comparaison des contours, l'échelle du détecteur de contours est $\sigma = 1.5$, le seuil sur le gradient est 80 et la distance r est de 6 pixels. Malheureusement, nous n'avons pas implanté l'étape d'estimation et de compensation de mouvement proposée. Le seuil sur la valeur de $max(p_{in},p_{out})$ varie entre 0.1 et 0.6.

Pour notre méthode, l'échelle des Gaussiennes est $\sigma = 1.5$. Deux échantillonnages différents ont été testés. Un test à basse résolution utilise un ratio 1 : 10 et un autre à haute résolution utilise un ratio 1 : 5. La taille des fenêtres de recherche est fixée à 7. τ_{mvt} est fixé à 10 degrés, τ_1 varie entre 1% et 70% et τ_2 varie entre 10% et 80%.

La figure 4 montre un graphe de *Rappel* versus *Précision*, duquel nous pouvons constater que pour les deux résolutions, notre algorithme génère moins de faux positifs que les autres méthodes, résultant en des meilleures valeurs de *Précision* pour chaque valeur de *Rappel*. Notre méthode est moins sensible au mouvement et donc localise mieux les transitions graduelles. À haute résolution, notre algorithme ne génère aucun faux positif pour la plupart des valeurs de *Rappel*< 86%. Pour des hautes valeurs de *Rappel*, notre algorithme surpasse les autres. Ce graphe montre que notre méthode est efficace et robuste. La table 4 donne la performance en secondes et trames par secondes (tps) pour chaque méthode.

5 Conclusion

Depuis plusieurs années, l'indexation vidéo est un domaine de recherche important qui requière des méthodes robustes pour la segmentation. Nous avons présenté un nouvel algorithme destiné à détecter les coupures franches aussi bien que les transitions graduelles. En utilisant les dérivées partielles de premier ordre, notre algorithme reconnaît les frontières de plans en étant peu influencé par le mouvement. Une étape d'échantillonnage permet de réduire considérablement le temps de calcul avec un impact raisonnable sur les résultats. Les résultats présentés confirment que notre méthode fonctionne bien avec tous les types de frontières et même en présence de mouvements forts.

Références

[1] P. Aigrain and P. Joly. The Automatic Real-Time Analysis of Film Editing and Transition Effects and its Applications. *Computers & Graphics*, 18(1):93–103, 1994.

[2] J. S. Boreczky and L. A. Rowe. Comparison of Video Shot Boundary Detection Techniques. In IS&T/SPIE *Symposium on Electronic Imaging: volume 2670*, pages 170–179, San Jose, 1996.

[3] L.-F. Cheong. Scene-Based Shot Change Detection and Comparative Evaluation. *Computer Vision and Image Understanding*, 79:224–235, 2000.

[4] A. Dailianas, R. B. Allen, and P. England. Comparaison of Automatic Video Segmentation Algorithms. In *Proceedings of* SPIE *Photonics East '95: volume 2615*, pages 2–16, Philadelphia, 1995.

[5] A. Hampapur, R. Jain, and T. Weymouth. Feature based Digital Video Indexing. In IFIP *TC2/WG 2.6 Third Working Conference on Visual Database Systems*, pages 115–141, Lausanne, 1995.

[6] B. K. P. Horn. *Robot Vision.* McGraw-Hill Book Company, 1986.

[7] S. Lawrence, D. Ziou, M.-F. Auclair-Fortier, and S. Wang. Motion Insensitive Detection of Cust and Gradual Transitions in Digital Videos. Technical Report 266, D.M.I., U. de Sherbrooke, Sherbrooke, Canada, Mai 2001. Accepté pour publication dans Pattern Recognition Letters.

[8] J. C.-M. Lee and D. M.-C. Ip. A Robust Approach for Camera Break Detection in Color Video Sequence. In IAPR *Workshop on Machine Vision Application*, pages 502–505, Kawasaki, 1994.

[9] A. Nagasaka and Y. Tanaka. Automatic Video Indexing and Full: Video Search for Object Appearances. In IFIP *TC2/WG 2.6 Second Working Conference on Visual Database Systems*, pages 113–127, Budapest, 1991.

[10] B. Shahraray. Scene Change Detection and Content-based Sampling of Video Sequences. In IS&T/SPIE *Symposium on Electronic Imaging: volume 2419*, pages 2–13, San Jose, 1995.

[11] R. Zabih, J. Miller, and K. Mai. A Feature-based Algorithm for Detecting and Classifying Scene Breaks. In ACM *International Conference on Multimedia*, pages 189–200, San Francisco, 1995.

[12] H. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic Partitioning of Full-motion Video. *Multimedia Systems*, 1(1):10–28, 1993.