

Conversion d'une vidéo à fréquence d'images élevée (High Frame Rate) vers une fréquence d'images inférieure (Low Frame Rate)

Étudiants : ARCH Anthony, DAUVERGNE Pierre-Henri, VIGOUR Jérémy

Tuteur ESIR : LE MEUR Olivier

Tuteur b<>com : AUBIE Jean-Yves

Résumé—Le monde de la vidéo a vu naître de nombreux formats durant ces dernières années. Les technologies ont rapidement évolué et nous permettent de visionner des vidéos en HD, Full HD et même Ultra HD. De la même manière, les caméras permettent de filmer des séquences vidéo HFR — avec un rythme d'image de l'ordre de 100 images par seconde — plus fluides que des vidéos LFR — 50 images par secondes. Malheureusement, tous les écrans ne sont pas en mesure d'afficher des vidéos enregistrées à de telles fréquences. Nous travaillons en partenariat avec b<>com pour effectuer la conversion d'une fréquence élevée vers une fréquence adaptée, afin d'obtenir des vidéos compatibles pour tous les écrans. Une vidéo HFR contient des images plus nettes qu'une vidéo LFR, supprimer une image sur deux provoquerait donc des saccades désagréables à l'œil. Nous proposons dans notre solution une approche prenant en compte l'effet de flou provoqué par des objets en mouvement. Cet effet est naturellement présent dans les vidéos LFR et est appelé flou de mouvement ou flou cinétique. Nous détectons le mouvement dans la vidéo via le calcul du flot optique, puis nous appliquons un traitement pour reproduire le flou cinétique selon le mouvement.

Mots clés—Haute fréquence d'image (High Frame Rate, HFR), faible fréquence d'image (Low Frame Rate, LFR), image par seconde, flou cinétique, flot optique, champ dense de vecteurs.

I. INTRODUCTION

LES entreprises évoluant dans le domaine du traitement d'image et de la compression vidéo cherchent à développer de nouveaux formats permettant un gain de qualité. Pour cela, deux paramètres peuvent être améliorés : la qualité spatiale (résolution d'image, compression) et temporelle (haute fréquence d'image). Ces deux paramètres sont l'objet de travaux au sein de l'institut de recherche technologique b<>com qui a pour objectif de fournir une meilleure sensation d'immersion des spectateurs dans les contenus audiovisuels.

Cet institut de recherche possède un laboratoire étudiant le réalisme des contenus. Leur but est de développer des outils visant à faciliter la prise en main de ces futurs formats par les professionnels du secteur (cinéma, télévision, jeux, publicité, etc.) et, à terme, par le public.

Parallèlement, les chercheurs de b<>com travaillent sur la possibilité de rendre ces formats lisibles sur des moniteurs non compatibles. C'est notamment le cas des vidéos enregistrées à une fréquence d'images élevées, typiquement 100 ou 120 images par seconde (ips), non lisibles sur des écrans qui ne

peuvent afficher que 50 ou 60 ips maximum. Dans cette optique, nous sommes amenés à proposer une solution permettant de réduire de moitié la fréquence d'image de vidéos HFR sans détériorer la fluidité des mouvements pour le spectateur.

En HFR, l'intervalle de temps entre 2 images consécutives est très court, les déplacements des objets entre les 2 images le sont donc également. La succession des images donne alors une impression de mouvement très proche de la réalité. En LFR, cet intervalle de temps étant plus important, la fluidité est obtenue par un rendu plus flou des objets en mouvements (figure 1). Il s'agit du flou cinétique, provoqué par le temps de pose plus long de la caméra lors de la capture d'une image.

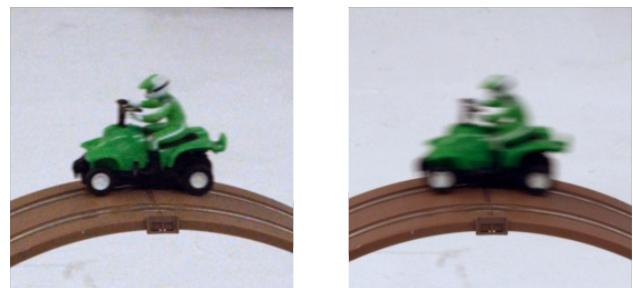


FIGURE 1. Comparaison entre une image HFR et son équivalent LFR.

Pour réduire la fréquence d'image, la solution intuitive consisterait à supprimer une image sur deux. Néanmoins, des travaux réalisés auparavant à b<>com ont montré que cette solution n'est pas satisfaisante car elle ne permet pas de retrouver le flou cinétique. La vidéo semble alors très hachée. De même, des traitements globaux comme une combinaison linéaire classique d'images successives n'élimine pas suffisamment les saccades.

Dans ce document, nous étudions la manière dont il est possible de simuler le flou cinétique localement sur les images. Les deux solutions que nous proposons se basent sur l'estimation du mouvement dans la vidéo originale. La première consiste à reprendre l'idée d'une combinaison de plusieurs images, cette fois par pixel. La seconde méthode est l'application de flous cinétiques sur les objets qui se déplacent.

II. MÉTHODOLOGIE

A. Modélisation du problème

Pour la suite du document, nous utilisons les notations suivantes :

$$\begin{aligned} I_n &: n^{\text{ème}} \text{ image dans la vidéo HFR} \\ I'_n &: n^{\text{ème}} \text{ image dans la vidéo finale LFR} \\ I_n(\mathbf{u}) &: \text{intensité (couleur et luminosité) du pixel } \mathbf{u} \text{ de} \\ &\text{coordonnées } \begin{pmatrix} x \\ y \end{pmatrix} \text{ dans l'image } I. \end{aligned}$$

À partir d'une vidéo enregistrée à une fréquence d'image élevée f , nous cherchons à obtenir une vidéo équivalente de fréquence f' :

$$f' = \frac{f}{2}$$

L'image I'_n correspond à l'image originale I_{2n} .

Comme expliqué dans l'introduction, la différence principale entre une vidéo LFR et une vidéo HFR d'une même scène réside dans le flou cinétique, moins visible lorsque la fréquence d'images augmente. C'est la raison pour laquelle la simple suppression d'une image sur deux sans autre traitement crée en effet de saccades dans la vidéo finale. Les travaux réalisés précédemment au sein de b<>com ont montré que des traitements globaux sur les images, comme des combinaisons linéaires d'images successives, n'éliminent pas suffisamment les saccades. Nous proposons d'étudier deux solutions basées sur des traitements locaux, en fonction des mouvements.

B. Évaluation du mouvement

Les traitements que nous proposons se basent sur l'évaluation des mouvements dans la scène. Nous cherchons à calculer le flot optique, ou champ dense de vecteurs : à chaque pixel de l'image est associé un vecteur qui exprime sa vitesse et sa direction en fonction de sa position dans l'image suivante. Ce vecteur est déterminé selon le principe de conservation de l'intensité :

$$I_{n+1}(\mathbf{u} + \mathbf{d}_u) = I_n(\mathbf{u}), \mathbf{d}_u = \begin{pmatrix} \Delta_x \\ \Delta_y \end{pmatrix}$$

L'intensité I du point \mathbf{u} ne varie pas entre l'image au temps n et l'image au temps $n + 1$. Durant cet intervalle de temps, le point se déplace selon le vecteur \mathbf{d}_u (de Δ_x pixels horizontalement et Δ_y pixels verticalement). Suivant cette hypothèse, détecter le mouvement revient à chercher pour chaque pixel p de I_n le pixel p' qui lui ressemble le plus dans I_{n+1} (en prenant en compte son voisinage). Le vecteur d_u est ensuite exprimé comme la distance horizontale et verticale en pixels entre p et p' . La figure 2 montre un sous-ensemble de vecteurs de mouvement calculés.

Pour cette étape, nous utilisons une fonction de la bibliothèque OpenCV basée sur l'algorithme de Farneback [1]. La fonction détermine notamment le flot optique à plusieurs échelles pour lisser le champ dense de vecteurs et ainsi limiter le nombre de vecteurs aberrants.

Une fois les mouvements détectés et exprimés pour chaque pixel, nous utilisons leurs vitesses et directions pour créer les effets de fluidité sur l'image. Nous envisageons deux



FIGURE 2. Exemple de flot optique

approches : la combinaison linéaire d'images adaptée au mouvement et l'application d'un filtre de flou directionnel.

C. Combinaison linéaire adaptée au mouvement

Dans cette première solution, nous souhaitons combiner chaque image principale I_{2n} avec ses 2 images adjacentes I_{2n-1} et I_{2n+1} en fonction des mouvements. Cette méthode nous permet dans un premier temps de vérifier la pertinence des mouvements détectés et d'obtenir une version approximative mais relativement peu coûteuse en temps de calcul.

Notre combinaison linéaire est réalisée pixel par pixel, contrairement aux combinaisons linéaires des précédents travaux effectuées directement sur l'ensemble des pixels. À chaque pixel des images voisines I_{2n-1} et I_{2n+1} est associé un poids proportionnel à la norme du vecteur vitesse de ce pixel. Le pixel u de l'image finale est ensuite calculé comme suit :

$$\begin{aligned} I'_n(\mathbf{u}) &= \omega_a \cdot I_{2n-1}(\mathbf{u}) + I_{2n}(\mathbf{u}) + \omega_b \cdot I_{2n+1}(\mathbf{u}) \\ \text{avec } \omega &= k \cdot \exp\left(\frac{-1}{\|\mathbf{d}_u\|}\right) \end{aligned}$$

Les poids ω sont compris entre 0 et 1, le scalaire k nous permettant de limiter l'apport des images voisines au profit de l'image principale. Il est nécessaire d'avoir un et un seul vecteur vitesse par pixel des images voisines. Nous devons donc chercher les mouvements depuis l'image voisine vers

l'image principale et non l'inverse, le flot optique calculé n'étant ni injectif, ni surjectif.

Intuitivement, si un objet est statique, il ne sera visible qu'à une seule position dans l'image obtenue. À l'inverse, on observera 3 traces d'un objet en mouvement, le long de sa trajectoire, comme dans la figure 3.

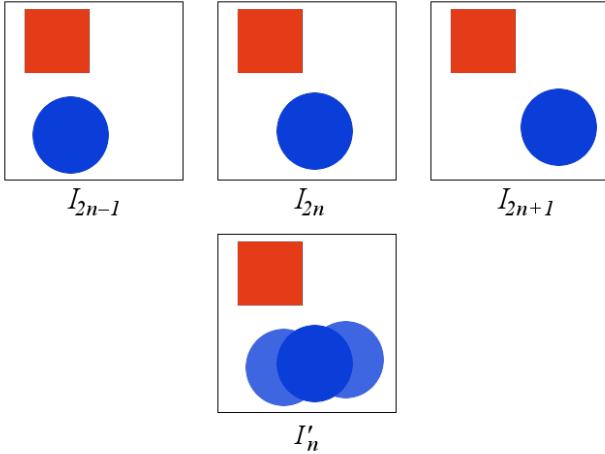


FIGURE 3. Combinaison linéaire adaptée au mouvement

Toutefois les traces obtenues sont nettes et le résultat est visuellement différent du flou cinétique naturel dans la vidéo LFR. Dans notre seconde approche, nous cherchons à reproduire le flou toujours à partir du mouvement observé dans la vidéo.

D. Ajout d'un flou directionnel

Afin de donner l'illusion de mouvements plus fluides, il faut que les traces des zones en mouvement soient plus lisses. La solution idéale se base sur l'application d'un filtre de flou dans la direction du mouvement.

La première étape consiste toujours à déterminer le flot optique. Cependant, les vecteurs de mouvement vont à présent servir à créer un filtre de flou directionnel pour chaque pixel de l'image principal. Il faut cette fois calculer le flot optique de l'image principale vers les images voisines.

La taille et les poids du filtre de flou sont calculés en fonction de la norme et de l'angle du vecteur de mouvement. Plus le mouvement est important, plus le filtre prendra en compte un nombre élevé de pixels voisins. L'application du flou est similaire à un produit de convolution entre l'image et un filtre, à ceci près qu'ici le filtre change selon le pixel. La figure 4 montre le résultat théorique de cette méthode.

Un pixel final est obtenu via la relation suivante :

$$I'_n(u) = \frac{1}{\sum_{i=0}^K \sum_{j=0}^L C_u(i, j)} \times \sum_{i=0}^K \sum_{j=0}^L C_u(i, j) \cdot I_{2n}(x + i - \frac{K}{2}, y + j - \frac{L}{2})$$

C_u est le filtre de convolution de taille $K \times L$ associé au pixel u , centré en u .

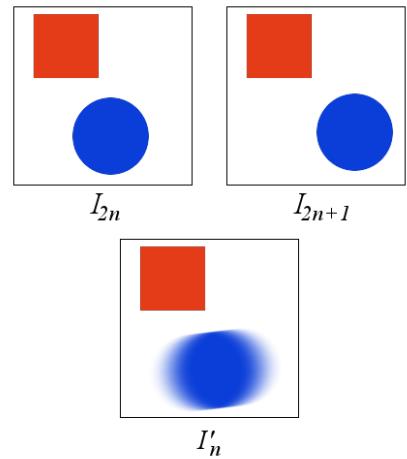


FIGURE 4. Flou directionnel en fonction du mouvement.

$\binom{i}{j}$ correspondent aux coordonnées relatives au filtre.

III. RÉSULTATS ET DISCUSSION

La figure 5 montre les images originales en HFR et LFR ainsi que les résultats des deux méthodes. Les images de gauche sont tirées d'une séquence où tous les éléments sont en mouvement. Les images de droite sont extraites d'une vidéo où le fond et la plupart des objets sont statiques.

Les résultats obtenus via nos deux méthodes permettent de réduire l'effet de saccade et de retrouver une fluidité suffisante pour le confort visuel. Notre étude montre qu'il est préférable d'adopter une approche locale basée sur le flot optique, plutôt qu'une technique classique de combinaison linéaire d'images.

D'un point de vue qualité visuelle, les tests montrent que l'ajout du flou directionnel produit naturellement un résultat plus proche de l'image LFR originale que la combinaison linéaire adaptée au mouvement. Dans certains extraits des vidéos, il est difficile de discerner la séquence LFR originale de la séquence LFR reconstruite avec le flou cinétique.

Nous pouvons remarquer que le flou directionnel pour la séquence « chevaux » est manquant autour des pattes avant. Phénomène également présent dans l'autre méthode mais moins visible, c'est l'une des principales limitations de l'algorithme de détection de flot optique utilisé. En effet, il tend à harmoniser le flot optique calculé pour supprimer localement les vecteurs aberrants. En contrepartie, il ne peut détecter les frontières nettes lorsque des objets se déplacent selon des directions différentes se chevauchent. Au lieu d'obtenir deux régions distinctes de vecteurs, l'algorithme détecte une seule région dont les vecteurs varient spatialement, comme dans la figure 6. Toutefois, ce défaut apparaît ponctuellement et non systématiquement sur une suite d'images successives. Il est en fait difficilement décelable lors de la lecture de la vidéo et n'est pas gênant pour le spectateur.

D'un point de vue complexité, la combinaison linéaire adaptée au mouvement a l'avantage d'être moins coûteuse en temps de calcul. En effet, elle consiste en un parcours des images pour calculer les poids des pixels et pour les

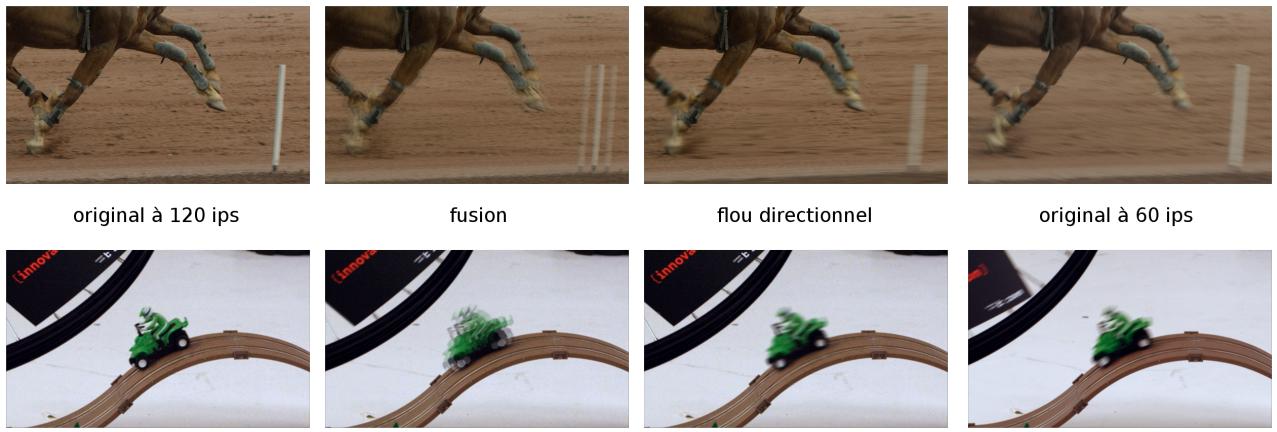


FIGURE 5. Images originales 120 et 60 ips et résultats.

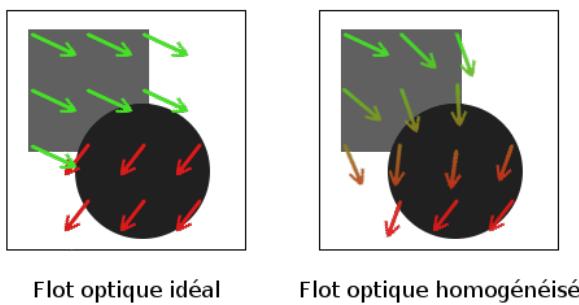


FIGURE 6. Limites de l'algorithme de calcul du flot optique.

combiner dans l'image résultat. En considérant une image de taille $N \times N$, on a donc une complexité d'ordre $O(N^2)$. En revanche pour la seconde méthode, une convolution de l'image par un filtre de taille $K \times K$ a une complexité d'ordre $O(K^2N^2)$, soit un coût quadratique en fonction de la taille du filtre. Pour traiter des images haute définition, c'est un coût très pénalisant. En programmation, la convolution d'une image par un filtre est généralement remplacée par une multiplication matricielle de l'image et du filtre dans l'espace de Fourier. Cela permet des gains de temps d'exécution très élevés. Néanmoins, cette technique n'est utilisable que pour un seul filtre convolué à toute l'image et dans notre cas, il y a autant de filtres que de pixels dans l'image.

IV. CONCLUSION

Dans ce papier, nous proposons deux méthodes pour diviser par deux la fréquence d'images d'une vidéo HFR. Les deux solutions sont basées sur la simulation du flou cinétique à partir des mouvements dans le scène, pour conserver la fluidité de la vidéo. Pour déterminer les mouvements, nous utilisons un champ dense de vecteurs obtenu par l'algorithme de Farneback. Nos deux approches locales — combinaison linéaire adaptée au mouvement et application d'un flou directionnel — fournissent des résultats plus convaincants que des approches globales qui ne tiennent pas compte des mouvements. La

technique du flou directionnel permet de récupérer des images très proches visuellement des images extraites de la vidéo LFR originale. Les effets de saccade sont estompés au profit d'une grande fluidité des vidéos et d'un confort visuel accru.

RÉFÉRENCES

- [1] Gunnar Farnebäck, “Two-frame motion estimation based on polynomial expansion,” *Lecture Notes in Computer Science*, vol. 2749, pp. 363–370, 2003.