Champs de Trajectoires : Suivi de Pixels Efficace et Cohérent sur Signal Vidéo

M. Tournadre[12] N. Stoiber[1] C. Soladié[2] P.-Y. Richard[2]

[1]Dynamixyz
Take-Two Interactive

[2]AIMAC
CentraleSupélec

{marc.tournadre, nicolas.stoiber}@take2games.com
{catherine.soladie, pierre-yves.richard}@centralesupelec.fr

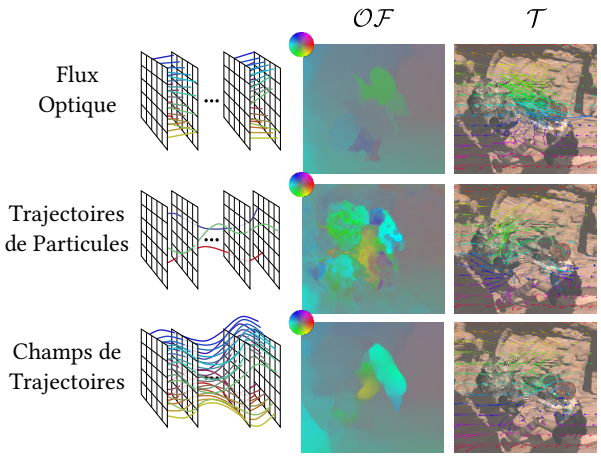

FIGURE 1 – Champs de trajectoires : les méthodes usuelles de *flux optique* (ici, RAFT [1]) sont naturellement cohérentes spatialement mais pas temporellement, celles de *trajectoires* (ici, PIPs [2]) l'opposé. Nous proposons ici une méthode holistique bas-niveau qui traite l'intégralité de la vidéo, ce qui donne des résultats plus cohérents, et plus rapidement.

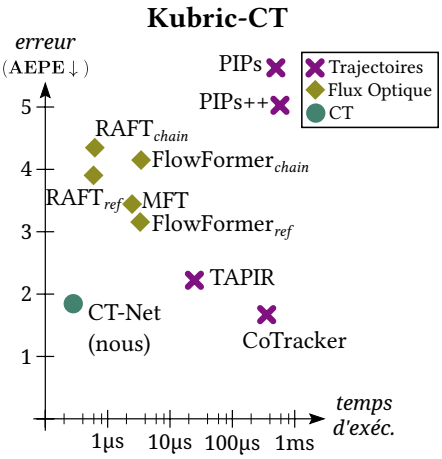

FIGURE 2 – Comparaison des différentes méthodes sur la tâche de Champs de Trajectoires, sur l'ensemble de validation de Kubric [3] dense. Le temps d'exécution est par trame et par pixel (échelle logarithmique).

Résumé

Inspiré des méthodes récentes de flux optique et de trajectoires, nous proposons une nouvelle approche holistique bas-niveau, nommée Champs de Trajectoires (CT), qui consiste à traquer tous les pixels d'une vidéo depuis une trame de référence, permettant une compréhension globale du mouvement d'une scène. Pour la résoudre, nous présentons CT-Net, une architecture résiduelle qui traite efficacement tous les pixels par l'intermédiaire de centroïdes saillants. Au regard des méthodes existantes, la nôtre est plus rapide et plus cohérente. Le code est accessible à `https://github.com/MTournadre/DTFNet`.

Mots clefs

Flux optique, Trajectoires, Suivi vidéo, Analyse bas-niveau

1 Introduction

La tâche de *flux optique* est essentielle en vision depuis des décennies [1], elle consiste à estimer le mouvement de chaque pixel d'une image à une autre. Lorsque appliqué à 2 trames consécutives, ce flux traduit un mouvement instantané, et peut être exploité de bien des manières : reconstruction 3D, reconnaissance d'action, compression vidéo. Par construction, ces méthodes (variationnelles et neuronales) offrent un résultat lisse et cohérent spatialement. Malgré certaines approches pour l'étendre à plus de trames, il ne permet pas de gérer efficacement le mouvement à long terme, parfois sujet aux occlusions.

Une approche plus récente d'analyse de mouvement [2] consiste à reconstruire la *trajectoire* de certains points sur la durée. Les trajectoires obtenues sont lisses temporellement, et robustes aux occlusions, gardant une trace des particules de départ. Ces méthodes fonctionnent sur un nombre réduit de particules, et les traitent généralement indépendamment, brisant leur relation spatiale.

Dans ce travail, nous nous inspirons des deux approches : nous estimons conjointement la trajectoire de tous les pixels, d'une traite, formant un Champs de Trajectoires (CT). Par une analyse holistique, à la fois temporelle et spatiale, nous garantissons un mouvement cohérent selon ces deux aspects. Pour l'estimer, nous construisons une architecture neuronale résiduelle, CT-Net, basée sur un mécanisme de *résumé par centroïdes* et des corrélations patch-à-

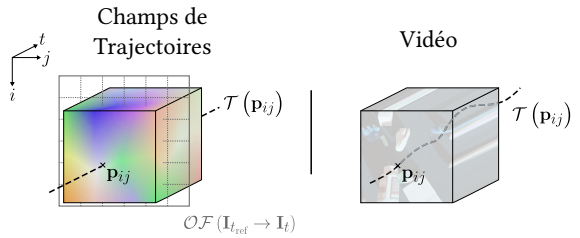

FIGURE 3 – Le Champs de Trajectoires est contenu dans un volume dense, indexable sur le temps t et l'espace (i, j). Une tranche sur t donne le *flux optique* $t_{\text{ref}} \to t$, une tranche sur (i, j) donne la trajectoire d'un point $\mathbf{p}_{ij}$. Chaque trajectoire est donc une ligne droite dans cet espace, alors que c'est une courbe dans l'espace vidéo, rendant ces deux espaces incompatibles.

patch, afin de traiter efficacement tout le signal vidéo. Pour l'entraîner, nous étendons le jeu de données Kubric [3]. Nous montrons par l'expérience que les méthodes existantes exhibent des mouvements incohérents soit spatialement soit temporellement, et sont plus lentes que la nôtre.

2 Champs de Trajectoires

Présentons d'abord la structure de l'information à reconstruire. Étant donnée une séquence vidéo d'entrée $\mathbf{I} \in \mathbb{R}^{T\times H\times W\times 3}$ de trames RGB, nous estimons le mouvement 2D de tous les pixels $\mathbf{p}_{ij}$ d'une trame de référence t_{ref}, et ce, pour tout t. Notons que t_{ref} n'a pas de contrainte, donc le CT n'est pas forcément causal. Cela donne un volume dense de mouvement $\mathbf{\Delta X}$, indexable sur (t, i, j), qui unifie les concepts existants : une tranche temporelle donne un *flux optique* $\mathcal{OF}$, et une tranche spatiale une *trajectoire* $\mathcal{T}$, comme illustré en figure 3.

On remarque que les deux espaces, *vidéo* et *mouvement*, fonctionnent différemment : dans le CT, une trajectoire est une ligne, tandis que c'est une courbe dans l'espace vidéo, ce qui les rend incompatibles. Autrement dit, la disposition spatiale du CT est celle de t_{ref} pour tout t, alors qu'elle change avec le temps sur la vidéo. Dans la suite, nous prendrons soin de bien séparer ces deux espaces.

3 CT-Net

Pour estimer le CT, nous construisons un réseau de neurones, CT-Net, qui raffine en parallèle deux espaces latents correspondant aux composantes vidéo et mouvement que nous venons d'introduire. Le but est de traiter conjointement ces deux espaces complémentaires. À l'instar de RAFT [1], la séquence d'entrée $\mathbf{I}$ est d'abord traitée par un encodeur convolutionnel $\mathcal{E}$, résultant en une composante latente $\mathbf{F}^0$. En parallèle, la composante mouvement $\mathbf{M}^0$ est initialisée à $\mathbf{0}$. Ces deux ensembles latents sont ensuite raffinés sur L couches résiduelles, chacune utilisant un mécanisme de *résumé par centroïdes*.

La composante mouvement $\mathbf{M}^l$ peut être convertie à tout moment en mouvement explicite $\mathbf{\Delta X}^l$ par une unique tête $\mathcal{H}_m$, un simple MLP :

$$\mathbf{\Delta X}^l = \mathcal{H}_m\left(\mathbf{M}^l\right) \qquad \forall l \in [1, L] \tag{1}$$

3.1 Résumé par Centroides

Suite à cette initialisation, nous raffinons, sur chaque couche l, les composantes image et mouvement de chaque pixel. Procéder par pixel serait calculatoirement trop lourd, donc nous mettons en place un *résumé par centroïdes* pour alléger le traitement. Pour cela, nous construisons un ensemble réduit de N_T jetons apprenables $\mathbf{T}^l$ (tokens). Dans la formulation traditionnelle des Transformers [4], ces jetons jouent le rôle des requêtes $\mathbf{Q}$ (query), mises en correspondance avec les clés $\mathbf{K}$ (key), projection des composantes images $\mathbf{F}^l$:

$$\mathbf{Q} = \mathbf{T}^l \tag{2}$$

$$\mathbf{K} = \mathbf{W}_K^l \mathbf{F}^{\mathbf{l}} \tag{3}$$

$$\mathbf{A} = softmax\left(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{D_T}}\right) \tag{4}$$

Nous réinterprétons ces poids d'attention $\mathbf{A} \in \mathbb{R}^{N_T\times THW}$ comme étant des *clusters d'attention* de pixels, dont on peut extraire des *centroïdes* $\mathbf{x} = \text{argmax}_{hw}\left(\mathbf{A}_{t_{\text{ref}}}\right)$. Ils sont alors au nombre de $N_T \ll THW$, réduisant la complexité algorithmique des Transformers à un problème linéaire, au lieu de quadratique. Ceci est illustré en figure 4.

3.2 Raffinement des Centroïdes

Notre nouvelle formulation permet de localiser chaque centroïde, donc d'échantillonner dans $\mathbf{M}$ leur composante $\mathbf{m}$. Contrairement aux Transformers classiques, cette composante (valeur) reste inchangée, ce qui permet d'appliquer $\mathcal{H}_m$ et donc de reconstituer sa trajectoire :

$$\mathbf{\Delta x}_t = \mathcal{H}_m(\mathbf{m}_t) \qquad \forall t \in [1, T] \tag{5}$$

On extraie alors des patchs de $\mathbf{F}$ le long de ces trajectoires, à diverses résolutions, pour former des produits de corrélations patch-à-patch $\mathbf{C}$. Concaténé aux composantes $\mathbf{f}$ et $\mathbf{m}$, $\mathbf{C}$ est traité par un réseau convolutionnel temporel, donnant les incréments $\mathbf{\Delta f}$ et $\mathbf{\Delta m}$.

3.3 Attention Réciproque

Le traitement décrit ci-dessus s'effectue par centroïde, ce qui est très léger, mais parcimonieux. Pour l'appliquer à tous les pixels, nous conservons les affinités pixel-centroïde établies par $\mathbf{A}$, et construisons son *attention réciproque* $\mathbf{A}^\intercal$:

$$\mathbf{A}^\intercal = softmax\left(\frac{\mathbf{K}\mathbf{Q}^\intercal}{\sqrt{D_T}}\right) \tag{6}$$

Conformément à l'analyse faite en section 2, les différentes composantes s'étendent aux pixels via des attentions différentes : $\mathbf{A}_t^\intercal$ pour $\mathbf{F}$, $\mathbf{A}_{t_{\text{ref}}}^\intercal$ pour $\mathbf{M}$. Suite à quoi, de simples réseaux convolutionnels spatiaux et temporels, rajoutent une analyse locale des différentes composantes, complémentaire au mécanisme global d'attention :

$$\mathbf{M}_t^{l+1} = \text{FFN}_M\left(\mathbf{M}_t^l + \mathbf{A}_{t_{\text{ref}}}^\intercal \mathbf{\Delta m}_t^l\right) \tag{7}$$

$$\mathbf{F}_t^{l+1} = \text{FFN}_F\left(\mathbf{F}_t^l + \mathbf{A}_t^\intercal \mathbf{\Delta f}_t^l\right) \tag{8}$$

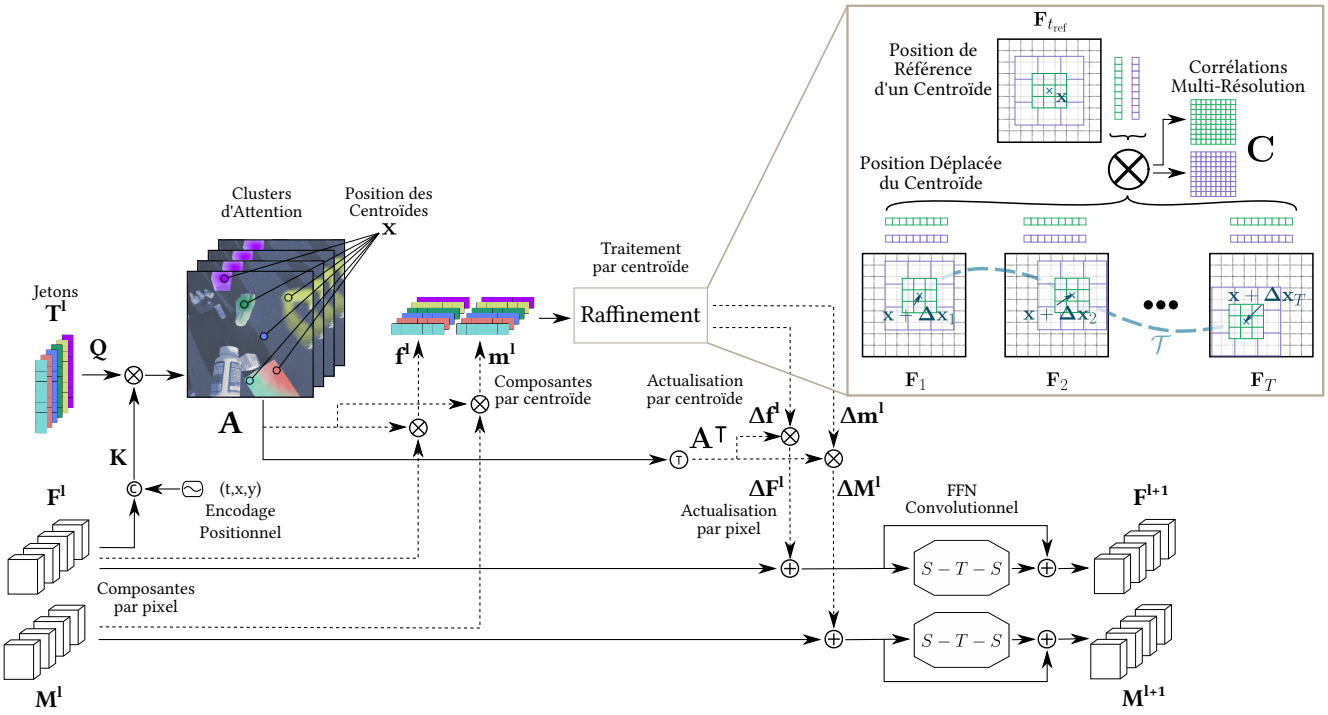

FIGURE 4 – Notre méthode de compression utilise des jetons $\mathbf{T}^l$ afin d'extraire des centroides depuis les clusters d'attention. Ces centroides permettent le traitement éparse des composantes image et mouvement, réduisant considérablement la complexité. On raffine le mouvement de chaque centroïde par des corrélations patch-à-patch, traitées dans un simple réseau convolutionnel temporel. La même relation est utilisée pour compresser et décompresser l'information, par construction d'une attention réciproque $\mathbf{A}^\intercal$. Enfin, un petit réseau convolutionnel spatialement et temporellement est appliqué, complémentaire au mécanisme global d'attention.

Voilà qui clot la couche l de CT-Net. En pratique, CT-Net contient $L = 8$ couches résiduelles consécutives.

Notons que ce mécanisme de compression/décompression est applicable à bien d'autres tâches, à condition qu'elles soient suffisamment diffuses dans l'espace.

3.4 Données

Pour entraîner notre réseau, nous souhaitons bénéficier de données denses et long terme, pour que CT-Net en apprenne la cohérence spatio-temporelle. A notre connaissance, aucun jeu de données actuel n'a de telle vérité terrain. Pour y remédier, nous étendons un travail existant, Kubric [3], afin de re-générer sa vérité terrain sur tous les pixels. Kubric est un jeu de données synthétique, composé de scènes d'objets rigides tombant et s'entrechoquant, avec un mouvement de caméra linéaire. Il contient 10k séquences vidéo de 24 trames de résolution 256x256.

Les annotations fournies contiennent déjà les coordonnées objet de chaque pixel, ainsi que les transformations rigides des objets sur le temps, ce qui permet de reconstituer la trajectoire 3D de tous les pixels d'une trame de référence, t_{ref}, choisie aléatoirement. Une fois projetée, nous obtenons la vérité terrain 2D.

On applique une augmentation de données analogue à RAFT [1] : perturbation de couleur, étirement de l'image, zones d'occlusion.

3.5 Entraînement

CT-Net est entraîné bout-à-bout à résoudre le mouvement 2D de tous les pixels de référence au court du temps. Cette fonction de coût est supervisée à chaque couche intermédiaire par une fonction Huber, avec un poids $\gamma \in [0, 1]$ déclinant avec la profondeur l :

$$\mathcal{L} = \frac{1}{THW} \sum_{l=1}^{L} \gamma^{L-l} \text{Huber}\big(\|\mathbf{\Delta X} - \mathbf{\Delta X}^*\|_2^1\big) \qquad (9)$$

L'entraînement se déroule sur 100k itérations, sur des séquences de 8 trames à 256x256. Sur 4x GPUs A100, 48Go, il dure 40h.

4 Résultats

On se compare à l'état de l'art de flux optique et de trajectoires. Les méthodes de trajectoires peuvent suivre chaque pixel indépendamment, ce qui donne le CT $\mathbf{\Delta X}$. Pour les méthodes de flux optique, deux stratégies sont envisagées : chaîner les flux instantanés $t \to t+1$, nommée X_{chain}, qui peut perdre trace de la particule de départ lors d'occlusions, et dériver sur la durée ; et estimer chaque flux $t_{\text{ref}} \to t$, nommée X_{ref}, qui doit résoudre des déplacements plus larges, mais gère les occlusions.

Les méthodes considérées sont RAFT [1], FlowFormer [6], et MFT [7] pour le flux optique, ainsi que PIPs [2], PIPs++ [8], TAPIR [9] et CoTracker [10] pour

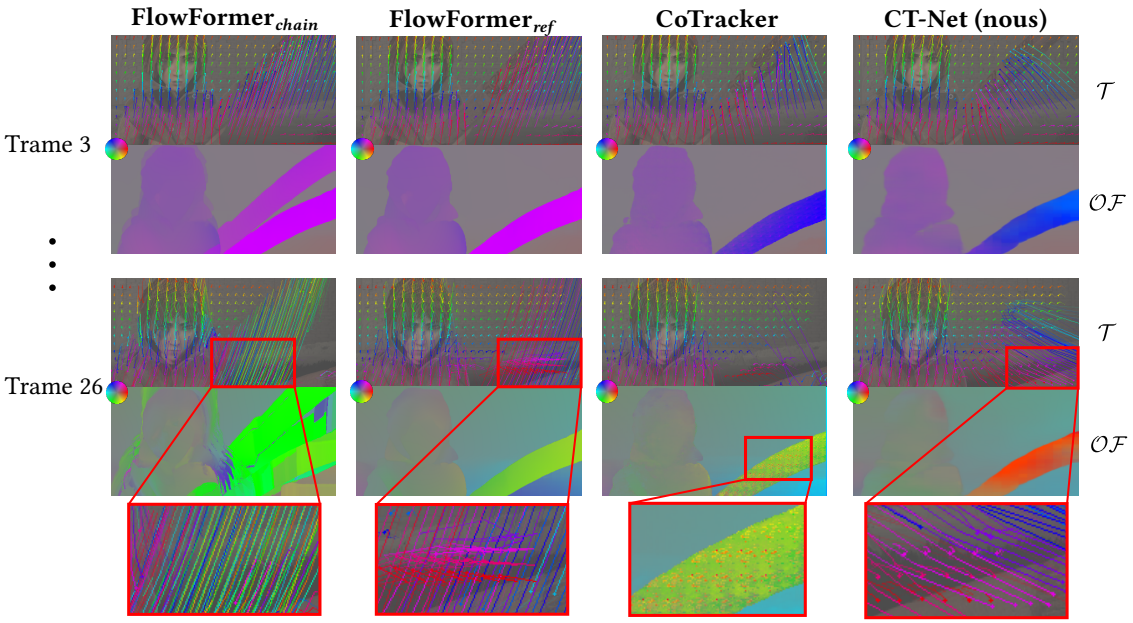

FIGURE 5 – Comparaison qualitative des différentes méthodes sur une séquence de test de Sintel [5]. Les méthodes usuelles de flux optique sont lisses spatialement mais pas temporellement, les méthodes de trajectoire l'inverse. La nôtre l'est sur les deux aspects. Comme elle est holistique, elle est la seule à avoir une compréhension globale de la séquence, et à retrouver le bâton après occlusion.

les trajectoires.

La figure 2 compare ces méthodes, en précision et temps d'exécution, sur l'ensemble de validation de Kubric dense. Notons que TAPIR, CoTraker et MFT sont également entraînés sur Kubric. Notre méthode est la plus rapide, et seulement battue en précision par CoTracker, qui est 10^3 fois plus lent. En outre, CoTracker utilise 10Go de VRAM pour traquer des grilles de 30x30 particules, quand la nôtre n'occupe que 2Go pour les 24 trames de 256x256.
Des visuels qualitatifs sont affichés en figure 5, sur une séquence de Sintel [5]. Comme on peut s'y attendre, les approches de trajectoires sont incohérentes spatialement : de larges zones censées être solidaires ont des mouvement désordonnés spatialement. A l'inverse, les méthodes de flux optique sont plus lisses spatialement, mais instables temporellement. La stratégie de chaîner les flux optiques entraîne une confusion des objets avec l'arrière-plan. CT-Net est le seul à avoir une compréhension globale, il est donc capable de retrouver le bâton au premier plan après occlusion. Bien qu'il ne soit pas le plus fin temporellement et spatialement, il est le seul à être cohérent sur les deux aspects à la fois.

5 Conclusion

Nous présentons une nouvelle tâche, Champs de Trajectoires, dont le but est l'analyse bas-niveau globale du mouvement d'une séquence. Nous construisons CT-Net, une architecture légère et efficace se focalisant sur des centroïdes grâce à un mécanisme d'attention réciproque. Nous étendons le jeu de données Kubric pour construire une donnée dense et entraîner notre réseau. Comparé aux méthodes de flux optique et de trajectoires, CT-Net est plus rapide et plus cohérent spatio-temporellement.

Annexe

Ce travail a été accepté à ACCV 2024.

Références

[1] Zachary Teed et Jia Deng. Raft : Recurrent all-pairs field transforms for optical flow. Dans *ECCV*, 2020.

[2] Adam W. Harley et al. Particle video revisited : Tracking through occlusions using point trajectories. Dans *ECCV*, 2022.

[3] Greff et al. Kubric : A Scalable Dataset Generator. Dans *CVPR*, 2022.

[4] A. Vaswani et al. Attention is all you need. Dans *Advances in neural information processing systems*, 2017.

[5] D. J. Butler et al. A naturalistic open source movie for optical flow evaluation. Dans *ECCV*, 2012.

[6] Z. Huang et al. FlowFormer : A transformer architecture for optical flow. *ECCV*, 2022.

[7] M. Neoral et al. Mft : Long-term tracking of every pixel, 2023.

[8] Y. Zheng et al. PointOdyssey : A Large-Scale Synthetic Dataset for Long-Term Point Tracking, 2023.

[9] C. Doersch et al. TAPIR : Tracking Any Point with per-frame Initialization and temporal Refinement, 2023.

[10] N. Karaev et al. Cotracker : It is better to track together. 2023.