

Video Frame Interpolation

John Robinson

Contenu de cette présentation

- Comprendre le problème
- Revue d'articles
 - Deep Bayesian Video Frame Interpolation
 - Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
 - IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation
 - Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation
 - Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation
- Que retenir de ces recherches ?
- Conclusion
- Prochaine étape

Interpolation d'images

Le problème

En se basant sur une série d'images

$$\mathcal{I} = \{I_{-(k-1)}, \dots, I_0, I_1, \dots I_{k-1}\}$$

Construire un modèle \mathcal{F} capable de générer une image intermédiaire.

$$I_t = \mathcal{F}(\mathcal{I}, t), \quad 0 < t < 1$$

k paramétrise le modèle et le dataset,

$$\mathcal{D} = \left\{ \bigcup_{l=1}^k I_{i \pm l}, I_i \right\}_{i=k}^{N-k}$$

- $k = 1$, triplets
- $k = 2$, quintuplets
- $k = 3$, septuplets

Le deep learning nous permet d'approcher ce problème de regression.

Regression "pure"

Le modèle \mathcal{F} tente de capturer la relation directe entre l'output I_t et les images adjacentes dans le dataset

Cette formulation offre peu de flexibilité, $t = 0.5$

Optic Flow

On considère ici une étape intermédiaire, celle de l'**optic flow**, qui caractérise le mouvement apparent de la scène.

La première étape consiste en l'estimation d'un certain nombre de optic flows, typiquement 2.

$$\phi = \{F_{i \rightarrow t}\}_i^K$$

où

$$F_{i \rightarrow t} \approx g(\mathcal{I}, t)$$

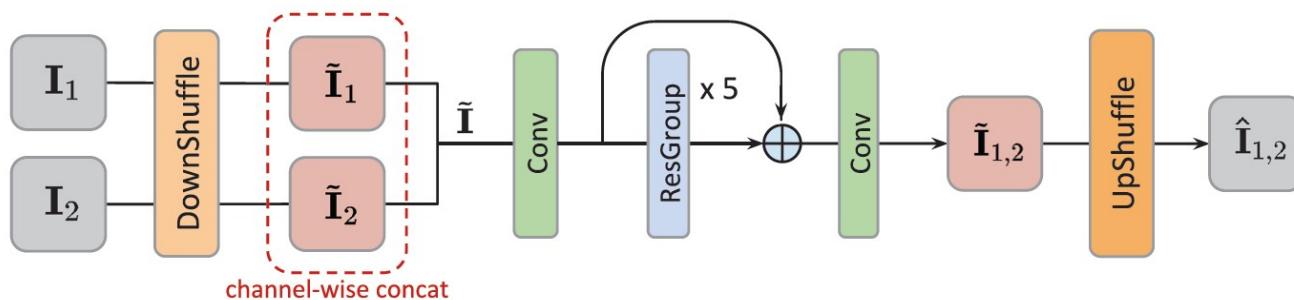
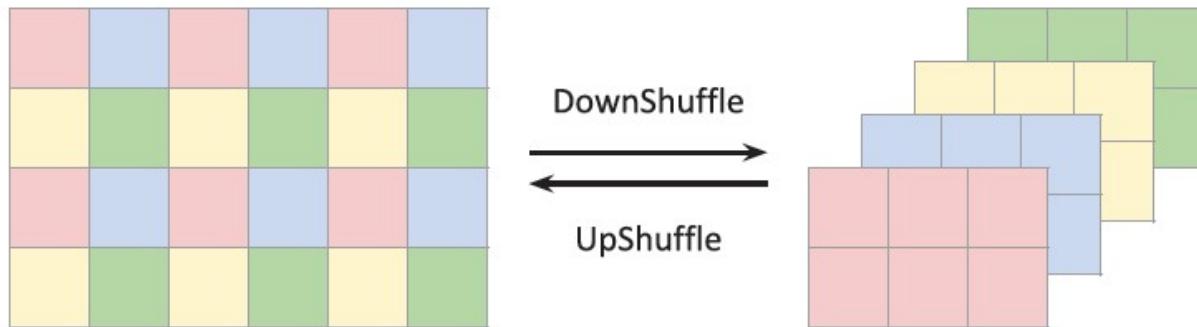
Le modèle \mathcal{F} interpole donc en fonction des images et des flows.

$$I_t = \mathcal{F}(\mathcal{I}, \phi)$$

Cette approche permet d'être **arbitraire** quant à t . Cependant, approximer l'optic flow en ne se basant que sur les images reste **imprécis** (problème d'occlusion, etc...).

Solution Actuelle, Modèle

Le modèle actuel exécute une régression "pure" et se base sur CAIN



Le modèle est entraîné à reconstruire l'image $I_{0.5}^{GT}$.

Solution Actuelle, Challenges

Difficultés avec la vitesse



Et les paternes répétitifs



Cette solution est donc perfectible.

Métriques et Evaluation

De nombreuses métriques telles que le PSNR et la SSIM

- Peak Signal-to-Noise Ratio (dB)

$$PSNR(I_1, I_2) = 10 \log_{10} \left(\frac{MAX^2(I_1)}{MSE(I_1, I_2)} \right)$$

Compare la qualité de I_2 par rapport à I_1

- Structural SIMilarity (entre 0 et 1)

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2\mu_y^2 + c_1)(\sigma_x^2\sigma_y^2 + c_1)}$$

Compare la structure de l'image x à celle de l'image y .

Ces métriques n'expliquent pas tout les aspects de la qualité d'une image,
l'analyse qualitative reste donc de vigueur.

Revue d'articles

Revue d'articles

- IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation
- Deep Bayesian Video Frame Interpolation
- Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
- Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation
- Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation

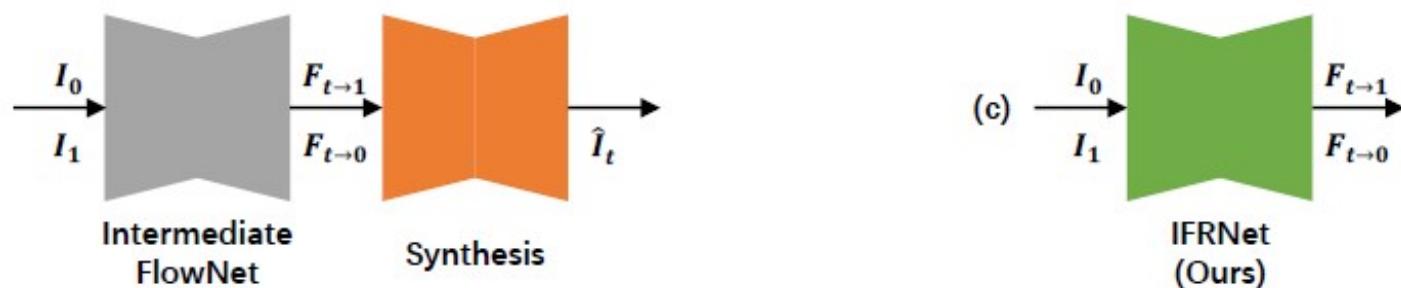
IFRNet (Mai 2022)

Cet article propose une approche encodeur-decodeur à plusieurs niveaux. Les images I_0 et I_1 sont encodées en une pyramide de features $\phi_0^{\{1,\dots,4\}}$ et $\phi_1^{\{1,\dots,4\}}$.

A chaque niveau k , le décodeur est chargé d'estimer les features interpolées $\hat{\phi}_t^k$ et $F_{t \rightarrow 0}^k$ et $F_{t \rightarrow 1}^k$.

Ces flows permettent de raffiner les features encodés ϕ_0^k , ϕ_1^k en $\tilde{\phi}_0^k$, $\tilde{\phi}_1^k$ afin de les décoder ensuite vers le prochain niveau $k - 1$.

Cette approche utilise les flows de manière plus holistique.



IFRNet, la méthode

L'encodeur construit une pyramide features.

$$\phi_{0,1}^{\{1,\dots,4\}} = \mathcal{E}(I_0, I_1)$$

Le premier décodeur \mathcal{D}^4 produit les premiers flows et features interpolés.

$$F_{t \rightarrow 0}^3, F_{t \rightarrow 1}^3, \hat{\phi}_t^3 = \mathcal{D}^4(\phi_0^4, \phi_1^4, T)$$

Les décodeurs intermédiaires $\mathcal{D}^k, k = 2, 3$ raffinent les flows et les features.

$$F_{t \rightarrow 0}^{k-1}, F_{t \rightarrow 1}^{k-1}, \hat{\phi}_t^{k-1} = \mathcal{D}^k(F_{t \rightarrow 0}^k, F_{t \rightarrow 1}^k, \hat{\phi}_t^k, \tilde{\phi}_0^k, \tilde{\phi}_1^k)$$

Le dernier décodeur \mathcal{D}^1 calcule les flows ainsi que M et R

$$F_{t \rightarrow 0}, F_{t \rightarrow 1}, M, R = \mathcal{D}^1(F_{t \rightarrow 0}^1, F_{t \rightarrow 1}^1, \hat{\phi}_t^1, \tilde{\phi}_0^1, \tilde{\phi}_1^1)$$

IFRNet, la méthode

L'output \hat{I}_t est ensuite calculée via un block warp-merge-add

$$\hat{I}_t = M \odot \tilde{I}_0 + (1 - M) \odot \tilde{I}_1 + R$$

où \odot est le produit d'élément à élément et

$$\tilde{I}_i = \text{BackwardWarping}(I_i, F_{t \rightarrow i})$$

M est un masque ajustant la fusion de \tilde{I}_0 et \tilde{I}_1 . R est une image résiduelle compensant les erreurs du warping.

IFRNet, entrainement

Le modèle est entraîné pour optimiser 3 loss

- \mathcal{L}_r pénalisant la reconstruction de l'image.
- \mathcal{L}_d pénalisant l'estimation des flows.
- \mathcal{L}_g pénalisant les changements de structure à l'échelle des features.

on considère alors l'optimisation jointe

$$\mathcal{L} = \mathcal{L}_r + \lambda \mathcal{L}_d + \eta \mathcal{L}_g$$

avec $\lambda = \eta = 0.01$.

IFRNet, entrainement

\mathcal{L}_r est calculée pour être la somme des Charbonnier et Census loss entre \hat{I}_t et I_{GT}

$$\mathcal{L}_r = \rho(\hat{I}_t - I_t^{GT}) + \mathcal{L}_{cen}(\hat{I}_t, I_t^{GT})$$

- $\rho(x) = (x^2 + \epsilon^2)^\alpha$ avec $\alpha = 0.5$ et $\epsilon = 10^{-3}$ est la loss de Charbonnier substitue la loss $L1$ en étant plus flexible.
- \mathcal{L}_{cen} calcule la distance de Hamming entre des patches de 7×7 transformés suivant la transformée de Census. Conserve les propriétés géométriques de l'image.

IFRNet, entrainement

IFRNet distille la connaissance d'un réseau de neurone **externe** pré-entraîné pour **estimer les flows**. Ses prédictions $F_{t \rightarrow 0}^p, F_{t \rightarrow 1}^p$ servent de pseudo label pour les décodeurs.

Pour ajuster la robustesse de cette distillation, les masques P_0 et P_1 sont calculés

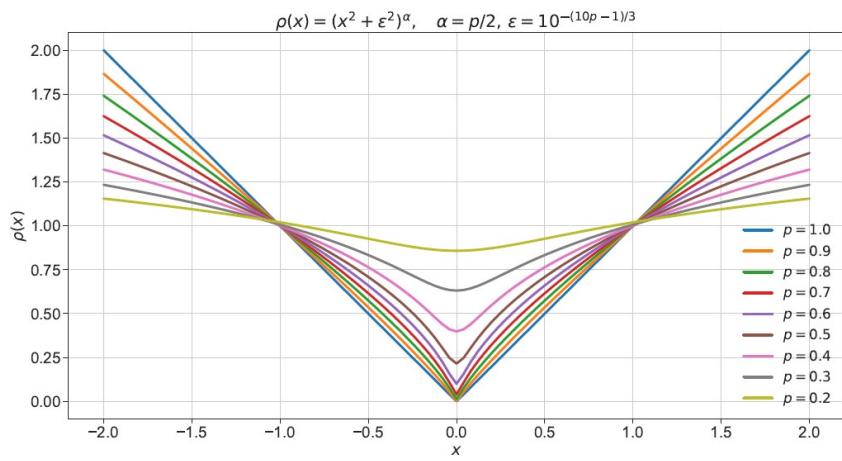
$$P_l = \exp(-\beta ||F_{t \rightarrow l} - F_{t \rightarrow l}^p||)$$

Desquels $p \in [0, 1]$ est déterminé, p ajuste les paramètres de ρ , $\alpha = p/2$ et $\alpha = 10^{-(10p-1)/3}$

La loss est ensuite calculée comme

$$\mathcal{L}_d = \sum_{k=1}^3 \sum_{l=0}^1 \rho(F_{t \rightarrow l}^{k \uparrow 2^k} - F_{t \rightarrow l}^p)$$

Cette formulation permet au modèle **d'apprendre** du réseau tiers tout en ajustant un degré de **confiance** en ce réseau grâce à β .



IFRNet, entrainement

IFRNet supervise aussi le calcul des features intermédiaires $\hat{\phi}_t^k$ en encodant les features de I_t^{GT} avec l'encodeur \mathcal{E} .

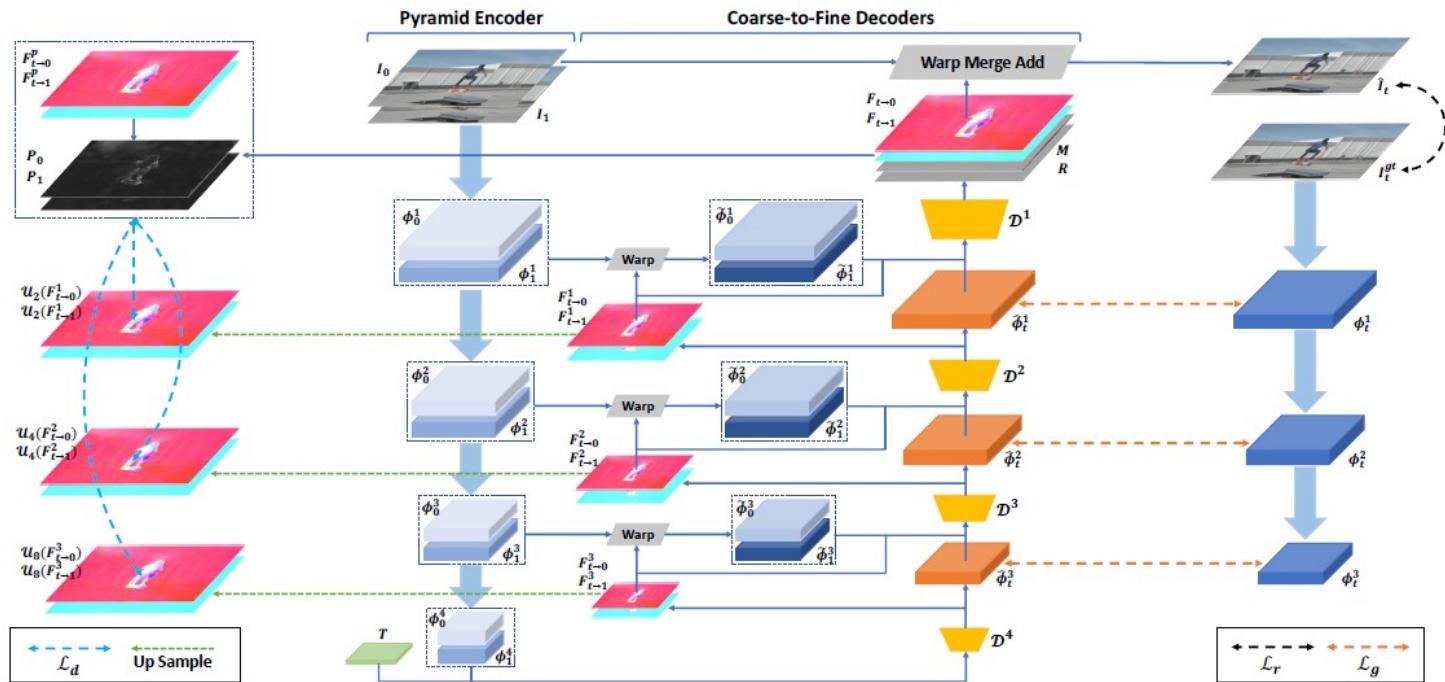
La similarité entre ces features est calculée avec la census loss sur des patches 3x3

$$\mathcal{L}_g = \sum_{k=1}^3 \mathcal{L}_{cen}(\hat{\phi}_t^k, \phi_t^k)$$

avec

$$\phi_t^{\{1, \dots, 3\}} = \mathcal{E}(I_t^{GT})$$

IFRNet, récapitulatif



IFRNet, récapitulatif

Qualitativement, IFRNet montre de bons résultats

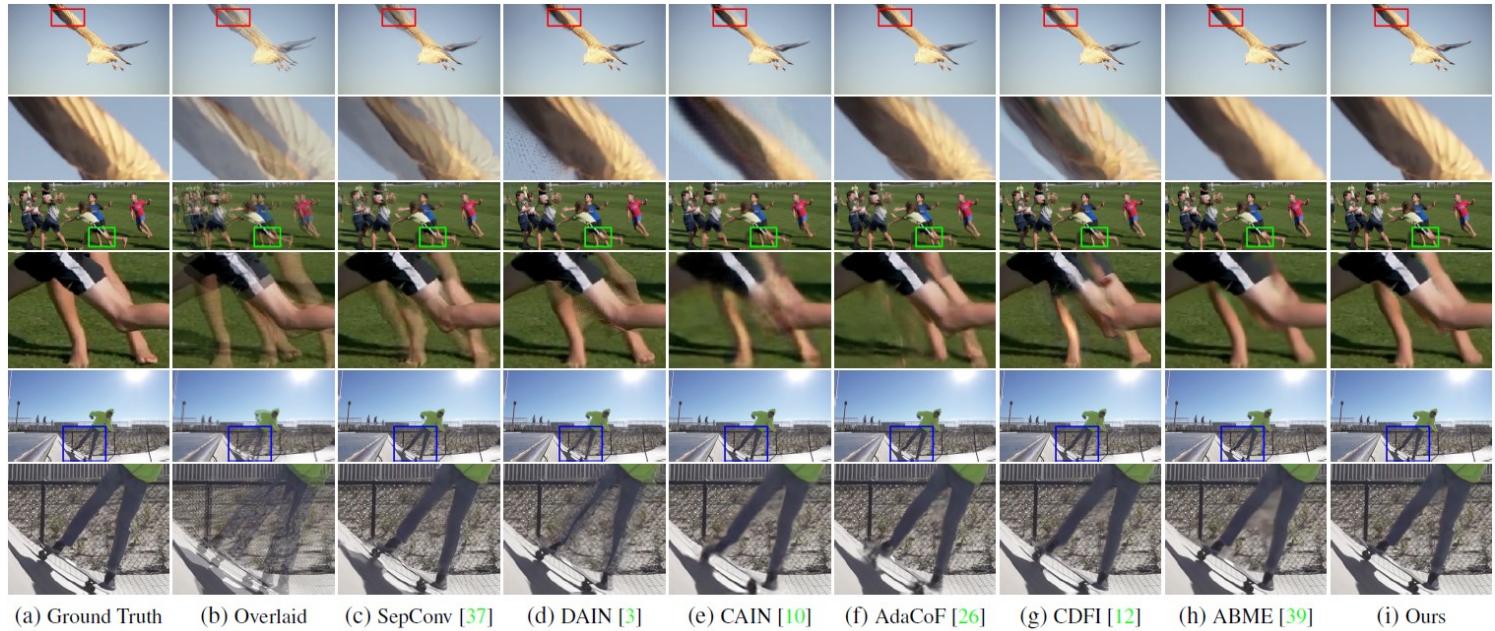


Figure 6. Qualitative comparison of different VFI methods on SNU-FILM (Hard) dataset. Proposed IFRNet algorithm can synthesize fast moving objects with sharp boundary while maintaining distinct contextual details. Zoom in for best view.

Revue d'articles

- IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation
- Deep Bayesian Video Frame Interpolation
- Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
- Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation
- Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation

DBVFI (Oct 2022)

L'approche suggérée est de considérer l'interpolation comme un problème de maximisation. L'image interpolée I_t^* maximise une distribution d'images conditionnée par les données

$$I_t^* = \operatorname{argmax}_{I_t} P(I_t | I_0, I_1, F_{0 \rightarrow t}, F_{1 \rightarrow t})$$

Ce modèle ensuite relaxé puis consolidé en considérant les possibles erreurs d'estimation de $F_{0 \rightarrow t}$ et $F_{1 \rightarrow t}$. L'interpolation est alors itérative s'apparent à une descente de gradient tirant part des réseaux de neurones.

DBVFI, la méthode

Le modèle est relaxé, les paires d'images et de flows $I_0, F_{0 \rightarrow t}$ et $I_1, F_{1 \rightarrow t}$ sont indépendantes.

$$P(I_t | I_0, I_1, F_{0 \rightarrow t}, F_{1 \rightarrow t}) = \prod_{i \in \{0,1\}} P(I_t | I_i, F_{i \rightarrow t})$$

Comme l'estimation des flows est basée sur un **framerate bas**, on présume donc une erreur $\Delta F_{i \rightarrow t}$ comme étant une variable latente du modèle. En intégrant pour toute les erreurs possibles,

$$P(I_t | I_i, F_{i \rightarrow t}) = \int_{\Delta F_{i \rightarrow t}} P(I_t | I_i, F_{i \rightarrow t}, \Delta F_{i \rightarrow t}) P(\Delta F_{i \rightarrow t} | I_i, F_{i \rightarrow t}) d\Delta F_{i \rightarrow t}$$

Cette intégrale est **incalculable**, on accepte alors une approximation avec

$$\hat{\Delta F}_{i \rightarrow t} = \operatorname{argmax}_{\Delta F_{i \rightarrow t}} P(\Delta F_{i \rightarrow t} | I_i, F_{i \rightarrow t})$$

$$P(I_t | I_i, F_{i \rightarrow t}) \approx P(I_t | I_i, F_{i \rightarrow t}, \hat{\Delta F}_{i \rightarrow t}) P(\hat{\Delta F}_{i \rightarrow t} | I_i, F_{i \rightarrow t})$$

DBVFI, la méthode

Avec ces changements, prendre le logarithme négatif donne l'expression d'une loss

$$\mathcal{L} = - \sum_{i \in \{0,1\}} (\log P(I_t | I_i, F_{i \rightarrow t})) + \log P(\Delta F_{i \rightarrow t} | I_i, F_{i \rightarrow t})$$

Permettant une descente de gradient sur les images et les erreurs

$$I_t^{(k+1)} = I_t^{(k)} - \lambda_I \frac{\partial \mathcal{L}}{\partial I_t}$$

$$\Delta \hat{F}_{i \rightarrow t}^{(k+1)} = \Delta \hat{F}_{i \rightarrow t}^{(k)} - \lambda_F \frac{\partial \mathcal{L}}{\partial \Delta \hat{F}_{i \rightarrow t}}$$

Les modules Flow/Image Gradient estiment ces gradients.

- $\frac{\partial \mathcal{L}}{\partial I_t}$ est formulé explicitement se basant sur le warping de $I_t^{(k)}$ par le flow $F_{i \rightarrow t} + \Delta \hat{F}_{i \rightarrow t}^{(k)}$
- $\frac{\partial \mathcal{L}}{\partial \Delta \hat{F}_{i \rightarrow t}}$ est formulé implicitement avec un réseau de neurones.

DBVFI, la méthode

Afin de réduire le nombre d'updates nécessaire, la méthode proposée approche l'optimisation en estimant l'update à apporter avec un réseau de neurones

$$I_t^{(k+1)} = I_t^{(k)} + \mathcal{G}_I \left(\left\{ \frac{\partial \mathcal{L}}{\partial I_t} \right\}, I_t^{(k)} \{F_{i \rightarrow t}\}, \{\Delta \hat{F}_{i \rightarrow t}\} \right)$$

$\{\cdot\}$ indique l'ensemble des évaluations pour chaque image I_0 et I_1 .

$$\Delta \hat{F}_{i \rightarrow t}^{(k+1)} = \Delta \hat{F}_{i \rightarrow t}^{(k)} + \mathcal{G}_F \left(\frac{\partial \mathcal{L}}{\partial \Delta \hat{F}_{i \rightarrow t}} I_i, \Delta \hat{F}_{i \rightarrow t}^{(k)}, F_{i \rightarrow t} \right)$$

Considérant que \mathcal{G}_I et \mathcal{G}_F partagent certains inputs, ces deux réseaux de neurones sont implémentés avec un CNN commun

DBVFI, l'entraînement

Toute chose confondue, exécuter une étape d'optimisation implique l'utilisation de 2 réseaux de neurones. Entrainer ce modèle consiste à réaliser K étapes d'optimisation,

$$I_t^{(1)}, \dots, I_t^{(K)}$$

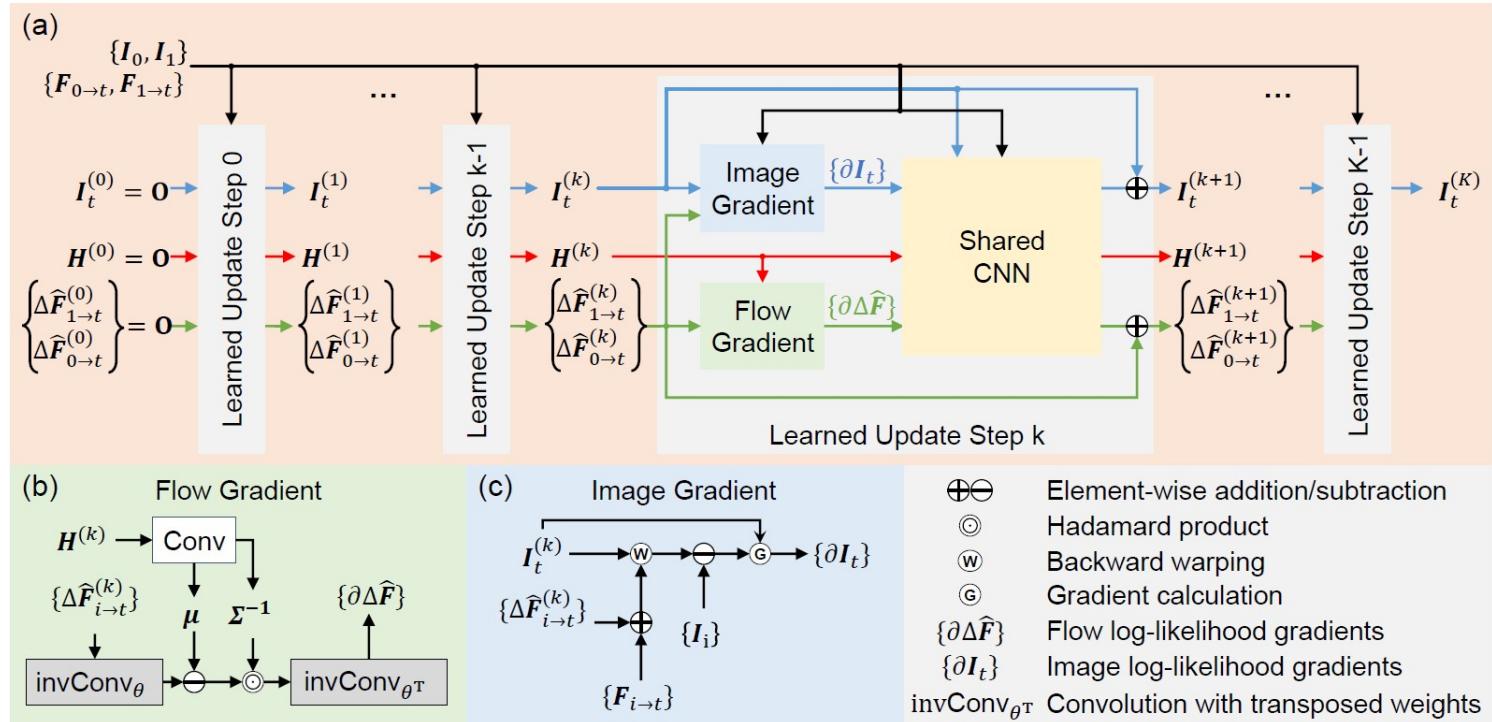
Et d'optimiser les paramètres de ses réseaux en considérant la reconstruction de l'image

$$\mathcal{L}_r = \sum_{k=1}^K \alpha_k \|I_t^{GT} - I_t^{(k)}\|_1$$

Les α_k sont déterminés empiriquement et augmentent avec les itérations

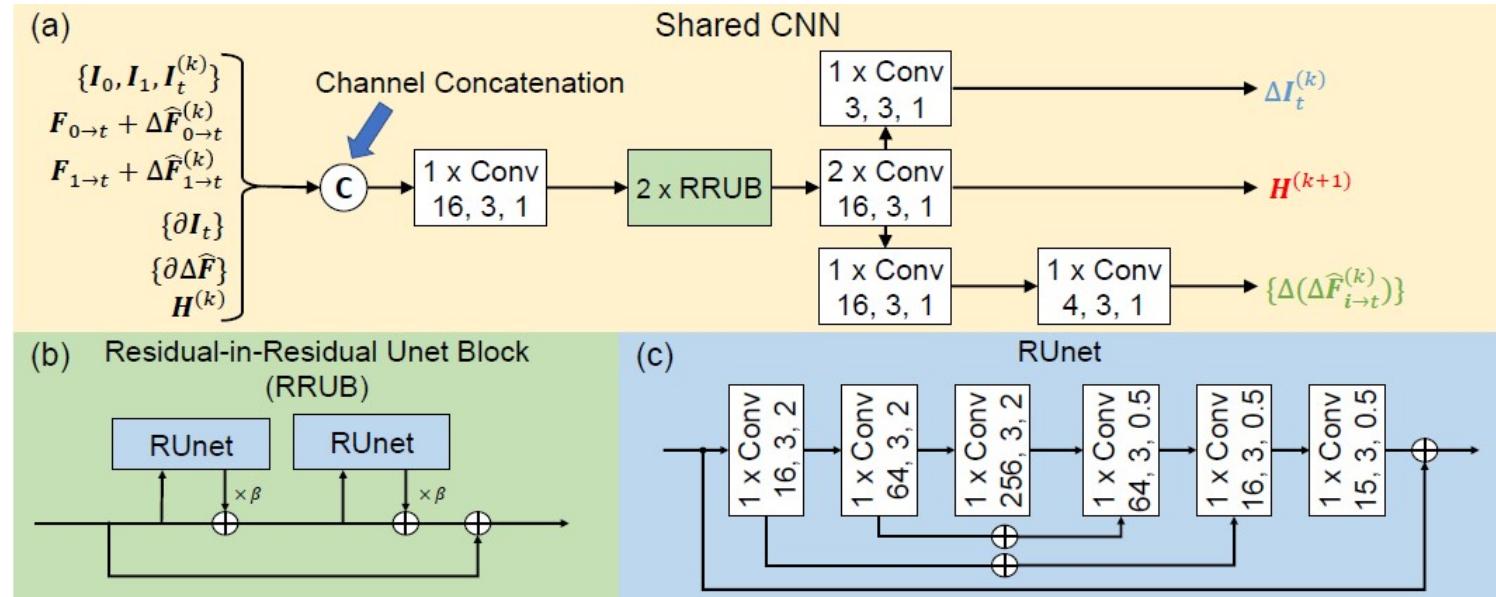
DBVFI, récapitulatif

Le modèle fonctionne selon ce pipeline.



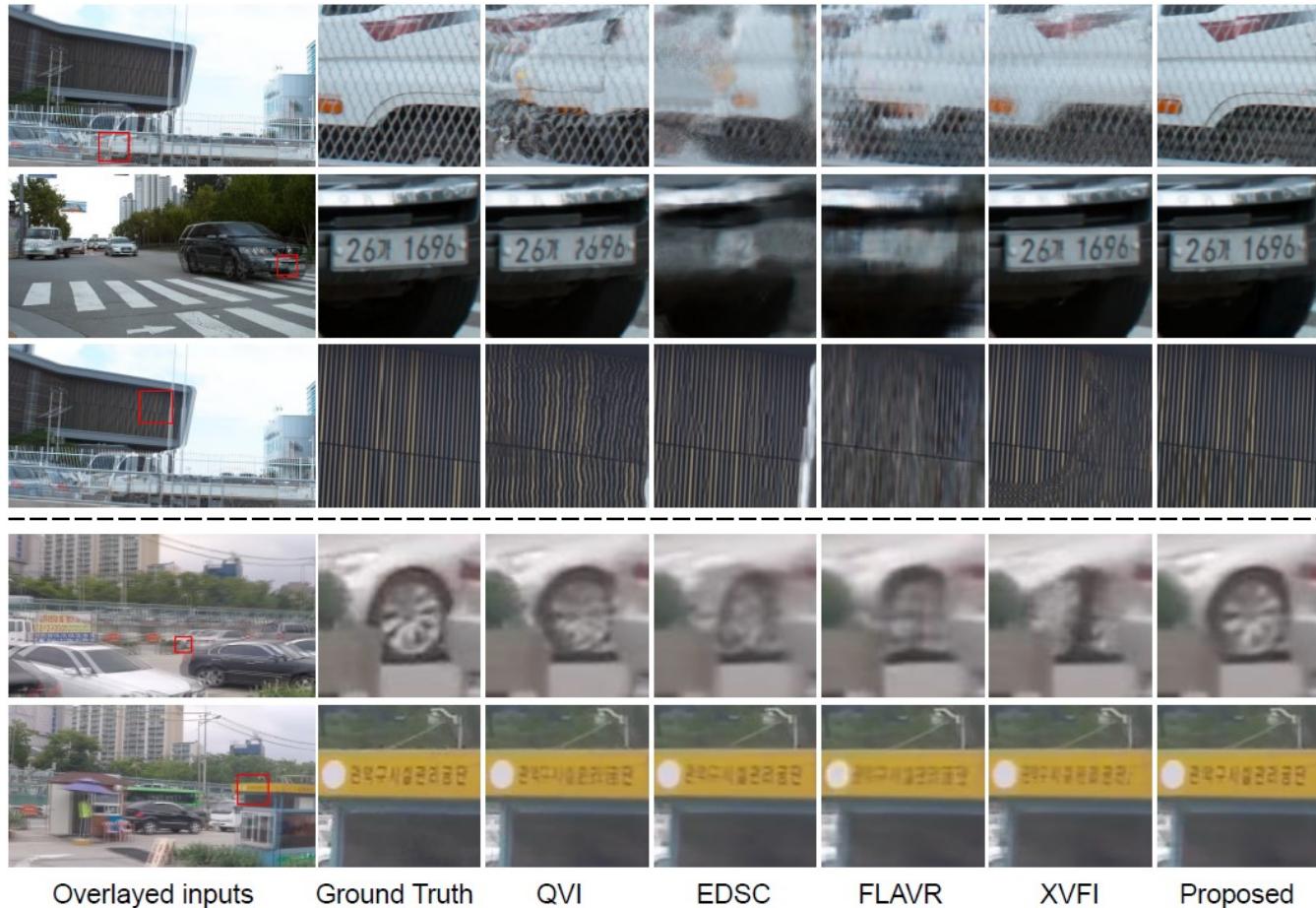
DBVFI, récapitulatif

Le réseau de neurones englobant \mathcal{G}_I et \mathcal{G}_F a cette structure.



DBVFI, récapitulatif

Qualitativement, les résultats sont satisfaisants, notamment au niveaux des structures répétitives



Overlaid inputs

Ground Truth

QVI

EDSC

FLAVR

XVFI

Proposed

Revue d'articles

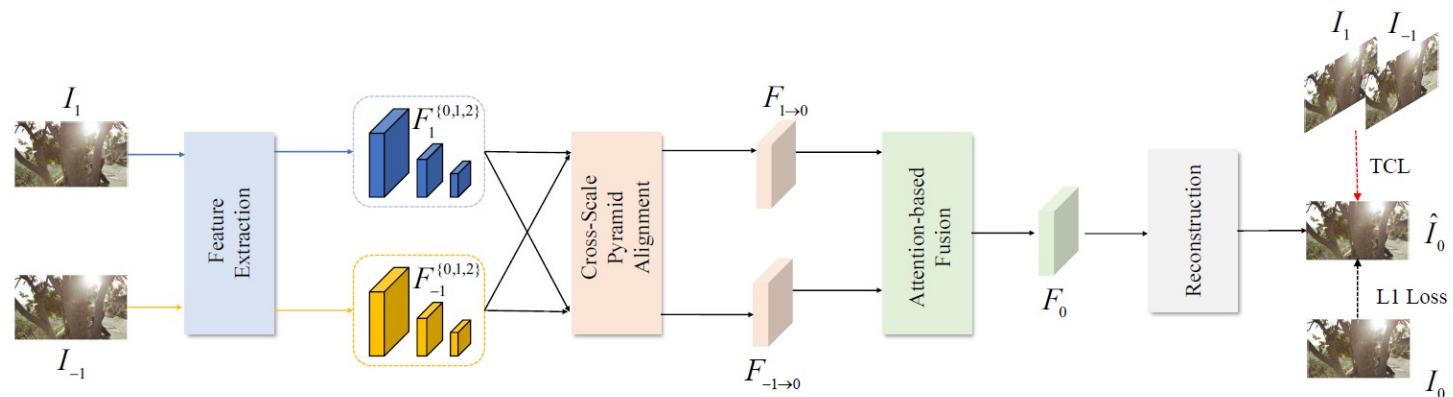
- IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation
- Deep Bayesian Video Frame Interpolation
- Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
- Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation
- Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation

Exploring Motion Ambiguity and Alignment (Mars 2022)

L'approche proposée se base également sur une décomposition des deux images en **pyramide de features**

$$\phi_0^{\{0,1,2\}} \text{ et } \phi_1^{\{0,1,2\}}$$

Ces features sont ensuite concaténés et alignés à travers leurs différents niveaux. **Cross Scale Pyramid Alignment**.



Exploring Motion Ambiguity and Alignment, la méthode

L'alignement de la pyramide se fait d'un niveau au niveau inférieur.

Au sommet, ϕ_0^2 et ϕ_1^2 sont alignés puis fusionnés

$$\phi_{0 \rightarrow 0.5}^2 = \text{Align}(\phi_0^2, \phi_1^2)$$

$$\tilde{\phi}_{0 \rightarrow 0.5}^1 = \text{Fuse}(\phi_{0 \rightarrow 0.5}^{2 \uparrow 2}, \phi_0^1)$$

La prochaine étape aligne ce résultat et fusionne avec **tout** les features antérieurs.

$$\phi_{0 \rightarrow 0.5}^1 = \text{Align}(\tilde{\phi}_0^1, \phi_1^1)$$

$$\tilde{\phi}_{0 \rightarrow 0.5}^0 = \text{Fuse}(\phi_{0 \rightarrow 0.5}^{2 \uparrow 4}, \phi_{0 \rightarrow 0.5}^{1 \uparrow 2}, \phi_0^0)$$

Le feature final $\phi_{0 \rightarrow 0.5}^0$

$$\phi_{0 \rightarrow 0.5}^0 = \text{Align}(\tilde{\phi}_{0 \rightarrow 0.5}^0, \phi_1^0)$$

Ce procédé est répété pour le calcul de $\phi_{1 \rightarrow 0.5}^0$

Exploring Motion Ambiguity and Alignment, la méthode

Le module **CSF** implémente Fuse comme une concaténation suivie d'une convolution.

Le module **AB** implémente Align en calculant un masque d'**offset** $O_{k \rightarrow 0.5}^l$ et de **weight** $W_{k \rightarrow 0.5}^l$.

Le feature aligné est calculé par une convolution

$$\phi_{k \rightarrow 0.5}^l(x) = \sum_i \tilde{\phi}_{k \rightarrow 0.5}^l(x + O_{k \rightarrow 0.5,i}^l(x)) * W_{k \rightarrow 0.5,i}^l(x)$$

i indique l' i ème élément du champ réceptif de cette convolution.

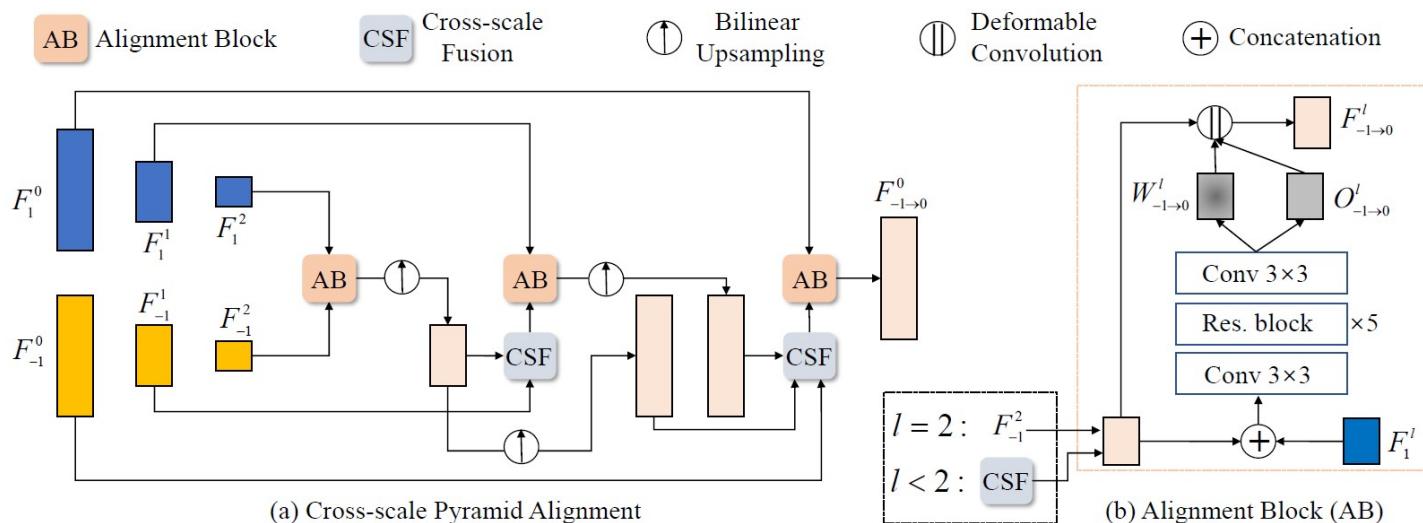
Le feature interpolé est calculé grâce à un masque d'attention.

$$\phi_{0.5} = M * \phi_{0 \rightarrow 0.5}^0 + (1 - M)\phi_{1 \rightarrow 0.5}^0$$

$$M = \sigma(\phi_{0 \rightarrow 0.5}^0 * \phi_{1 \rightarrow 0.5}^0)$$

$\hat{I}_{0.5}$ est ensuite reconstruite à partir de $\phi_{0.5}$ par un module composé de blocs résiduels et d'une convolution.

Exploring Motion Ambiguity and Alignment, la méthode



Exploring Motion Ambiguity and Alignment, entrainement

Pour l'entraînement, on pénalise le modèle

- Sur la **reconstruction** de l'image

$$\mathcal{L}_1 = \|\hat{I}_{0.5} - I_{0.5}^{GT}\|_1$$

- Sur la **cohérence des textures** de l'image interpolée comparée aux inputs.
 \mathcal{L}_{TCL}

L'interpolation est alors formulée

$$\hat{I}_{0.5} = \underset{I_{0.5}}{\operatorname{argmin}} \mathcal{L}_1(\hat{I}_{0.5}, I_{0.5}^{GT}) + \alpha \mathcal{L}_{TCL}(\hat{I}_{0.5}, I_0, I_1)$$

Exploring Motion Ambiguity and Alignment, entrainement

Comparer la texture d' $\hat{I}_{0.5}$ avec celle de I_0 et I_1 se fait par patches.

Pour un patch \hat{f}_x centré en x , on recherche le patch $f_y^{t^*}$ lui correspondant le plus.

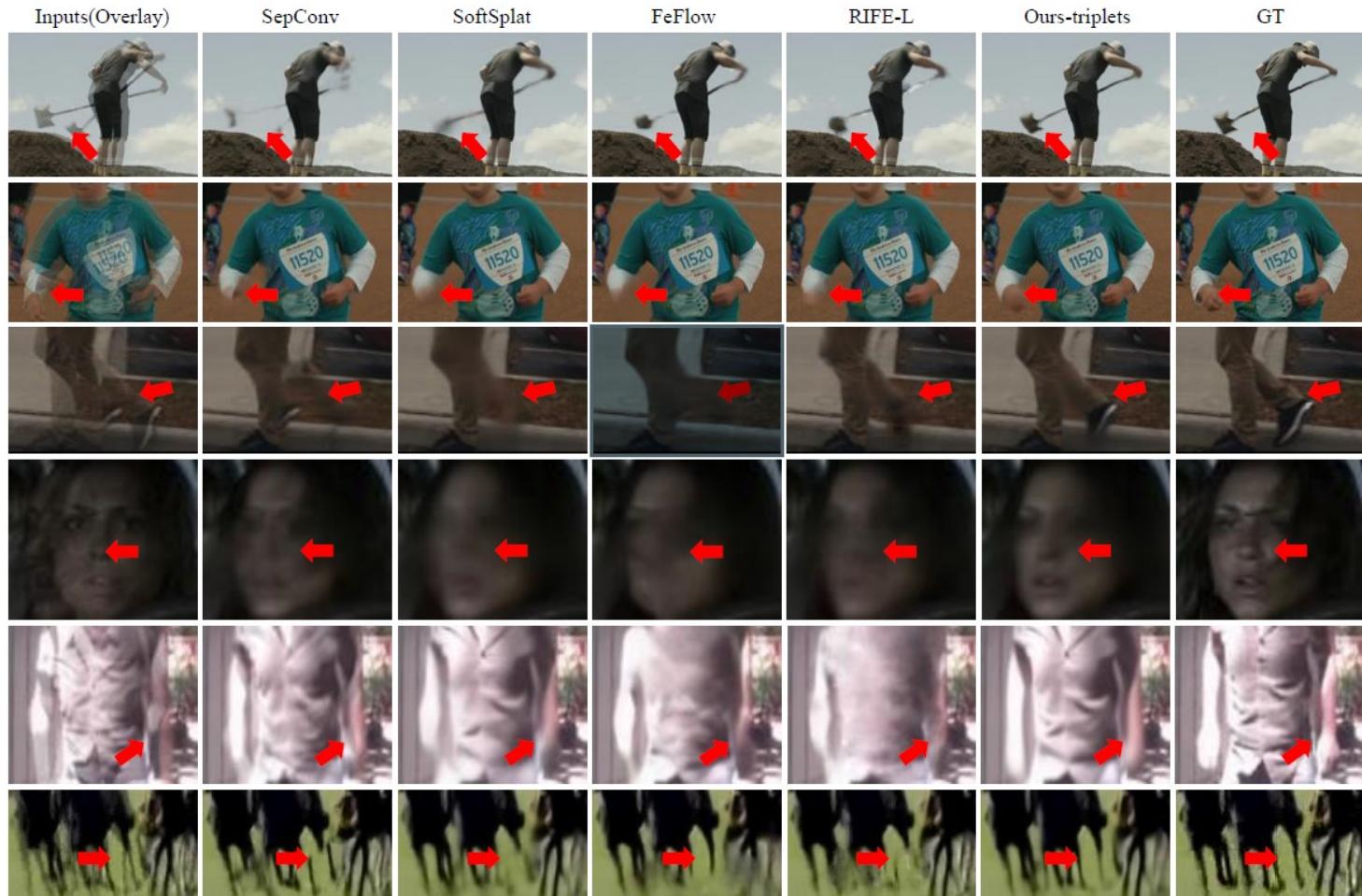
$$t^*, y^* = \underset{t \in \{0,1\}, y}{\operatorname{argmin}} \| \hat{f}_x - f_y^t \|_2$$

t^* et y^* indexent l'image et la position.

On pénalise alors la reconstruction de ce patch

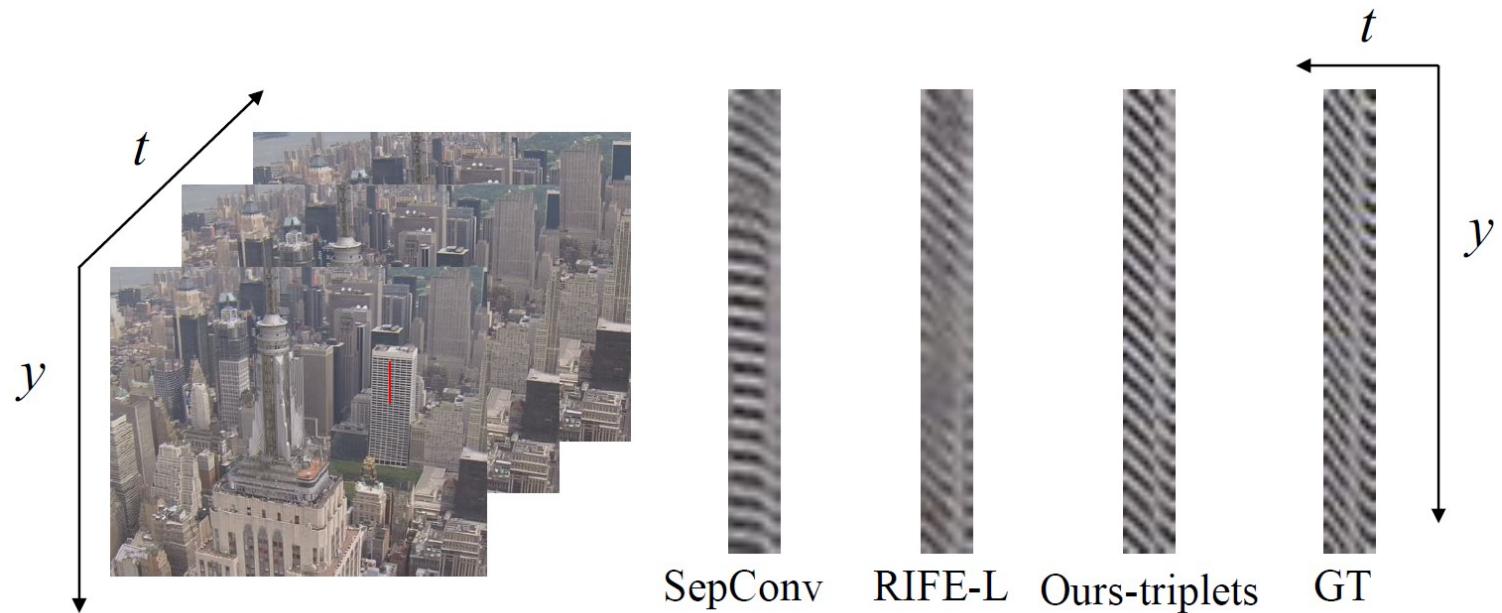
$$\mathcal{L}_{TCL}(\hat{I}_{0.5}, I_0, I_1) = \| \hat{f}_x - f_{y^*}^{t^*} \|_1$$

Exploring Motion Ambiguity and Alignment, récapitulatif



Exploring Motion Ambiguity and Alignment, récapitulatif

On note également que les structures répétitives évoluent de manière cohérente.



Revue d'articles

- IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation
- Deep Bayesian Video Frame Interpolation
- Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
- Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation
- Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation

Uncertainty Guided Spatial Pruning (Oct 2023)

Lorsqu'on interpole deux images, l'essence est de concentrer le calcul sur les zones de mouvement.

Cet article présente une méthode permettant de **déterminer ces zones**, réduisant la complexité de nos modèles



Uncertainty Guided Spatial Pruning, la méthode

La méthode proposée utilise deux réseaux de neurones

- Un réseau d'incertitude (UEN), responsable d'estimer les zones de mouvement.
- Un réseau d'interpolation (VFI), responsable de l'interpolation des images.

L'architecture du VFI est très similaire à celle d'[IFRNet](#).

Les images sont encodées en pyramide

$$\phi_{0,1}^{\{1,\dots,3\}} = \mathcal{E}(I_0, I_1)$$

Pour être ensuite décodées en [features](#) et [flows](#) sur plusieurs niveaux.

$$F_{t \rightarrow 0}^{k-1}, F_{t \rightarrow 1}^{k-1}, \hat{\phi}_t^{k-1} = \mathcal{D}^k(F_{t \rightarrow 0}^k, F_{t \rightarrow 1}^k, \hat{\phi}_t^k, \tilde{\phi}_0^k, \tilde{\phi}_1^k)$$

Uncertainty Guided Spatial Pruning, la méthode

L'entraînement du VFI est similaire, on ne distille pas la connaissance des flows d'un réseau tiers.

A chaque niveau, les décodeurs produisent additionnellement un masque d'incertitude P_k indiquant au décodeur $k - 1$ les zones sur lesquelles convoluer.

La génération de ces masques est supervisée par l'UEN.

L'entraînement du modèle se fait alors en deux parties, premièrement l'entraînement de l'UEN puis du VFI.

Ignorer les zones redondantes implique l'utilisation de sparse convolution

- Durant l'entraînement, pour permettre la propagation des gradients, le résultat d'une convolution dense est masqué.
- Durant l'inférence, la convolution est exécutée en n'appliquant le kernel que sur les zones spécifiées par le masque.

Uncertainty Guided Spatial Pruning, entraînement

L'UEN tâché d'estimer la variance de l'image interpolée est pénalisé par

$$\mathcal{L}_{su} = \exp(-U) \|I_t - f(I_0, I_1)\|_1 + 2U$$

Le VFI est lui pénalisé sur

- La **reconstruction** de l'image

$$\mathcal{L}_{rec} = \|I_t - I_t^{GT}\|_1$$

- Le degré **d'omission** contrôlé par S_t

$$\mathcal{L}_s = \left\| \frac{1}{\sum_{k=1}^3 H_k, W_k} \left(\sum_{k=1}^3 \sum_{h=1}^{H_k} \sum_{w=1}^{W_k} P_{k,h,w} \right) - S_t \right\|_1$$

- La **prédiction des masques**

$$\mathcal{L}_{ugm} = \|P_k^{u \downarrow 2^{k+1}} - P_{k+1}\|_1$$

P_{k+1} est le masque prédit par le VFI et $P_k^{u \downarrow 2^{k+1}}$ et le masque estimé par l'UEN.

Uncertainty Guided Spatial Pruning, entrainement

Additionnellement, le VFI utilise une branche auxiliaire n'omettant aucune zone à chaque niveau qui output I_t^{sc} et $\hat{\phi}_t^{sc,k}$.

On peut alors pénaliser le modèle sur la reconstruction des features et images non masqués

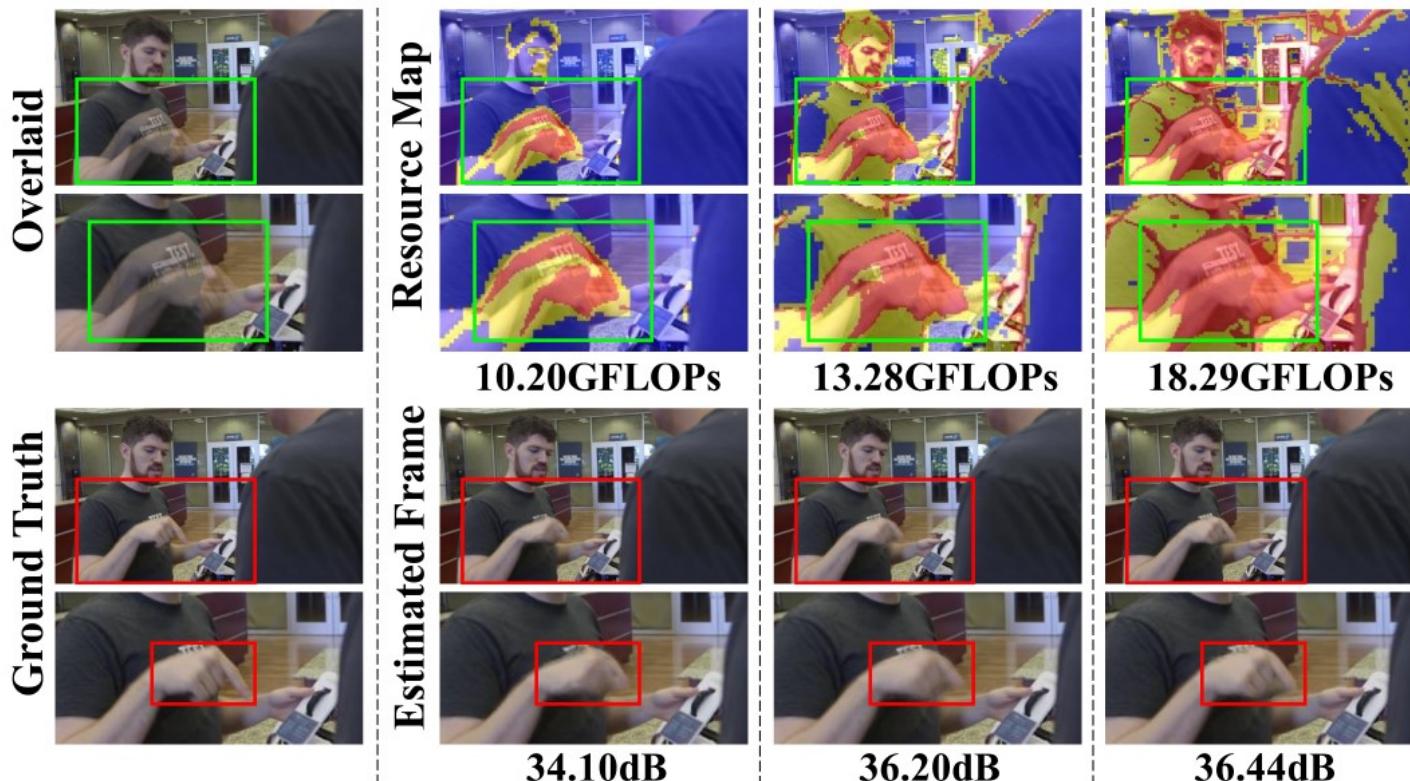
$$\mathcal{L}_{sc} = \|I_t^{sc} - I_t^{GT}\|_1 + \sum_{k=1}^3 \mathcal{L}_{cen}(\hat{\phi}_t^{sc,k}, \hat{\phi}_t^k)$$

On entraîne alors le VFI en optimisant

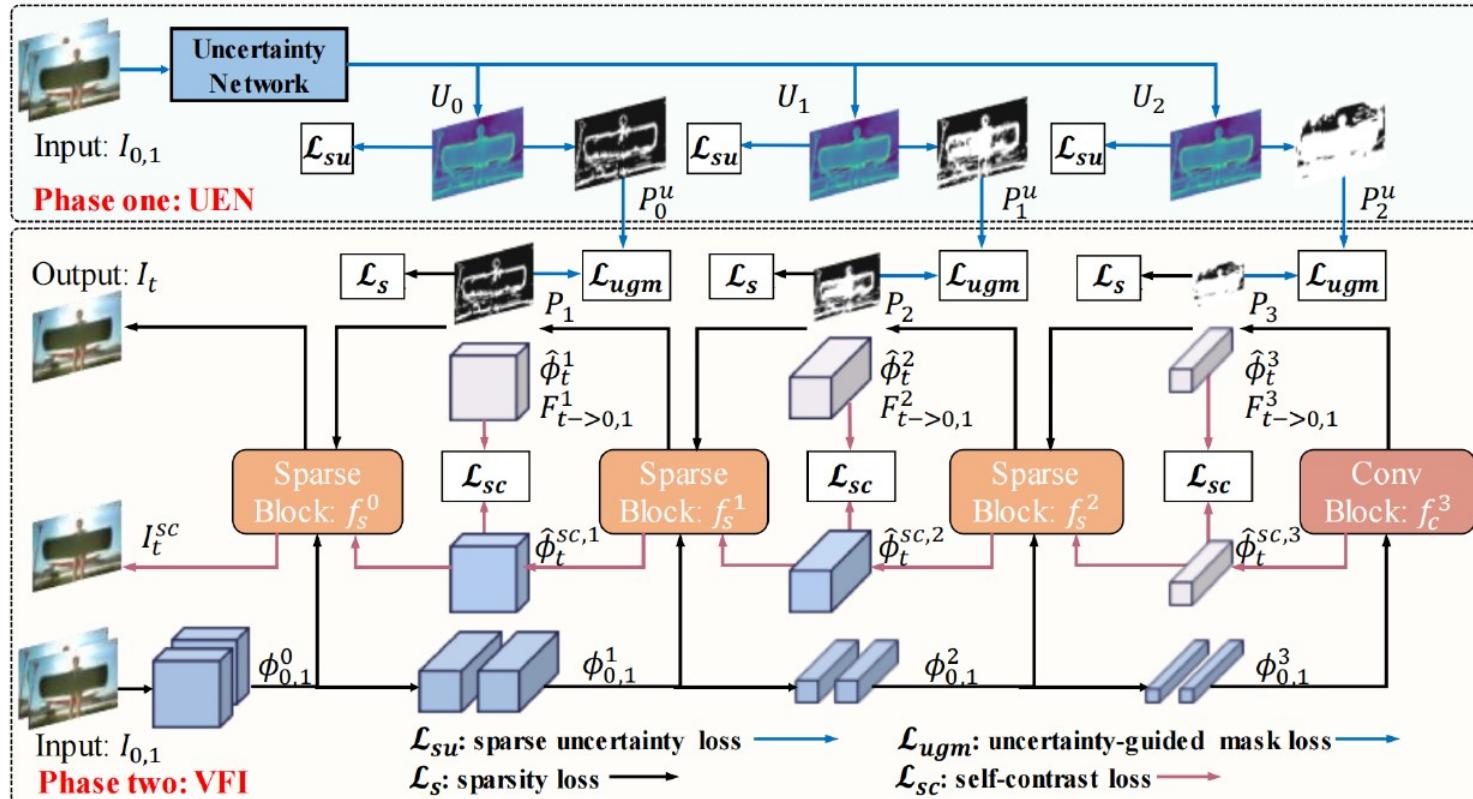
$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_s \mathcal{L}_s + \lambda_{ugm} \mathcal{L}_{ugm} + \lambda_{sc} \mathcal{L}_{sc}$$

Uncertainty Guided Spatial Pruning, recapitulatif

L'hyper-paramètre S_t permet d'ajuster l'utilisation des resources.

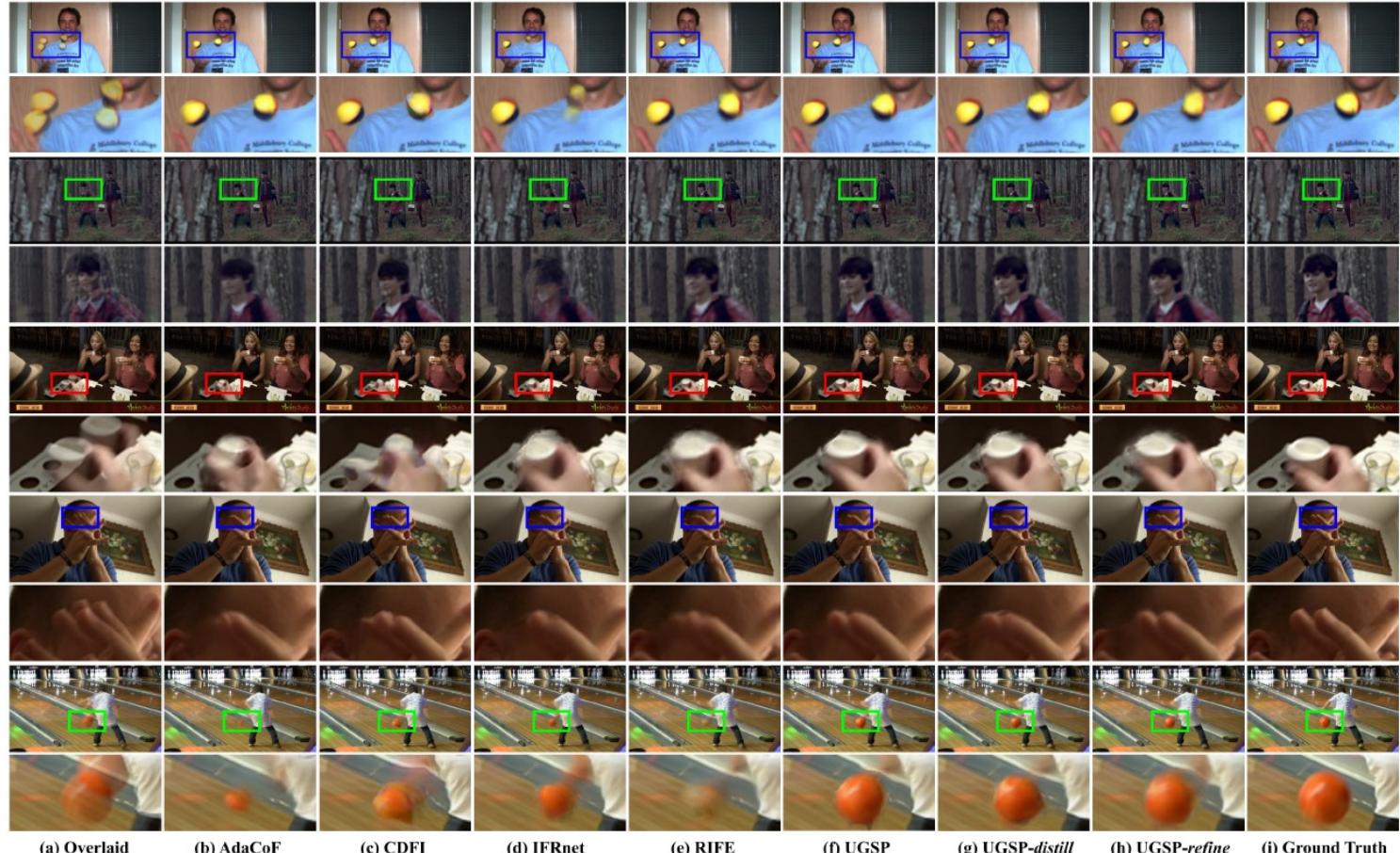


Uncertainty Guided Spatial Pruning, recapitulatif



(a) Overview of our proposed UGSP.

Uncertainty Guided Spatial Pruning, recapitulatif



Revue d'articles

- IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation
- Deep Bayesian Video Frame Interpolation
- Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation
- Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation
- Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation

Clearer Frames, Anytime (Nov 2023)

Cet article ne présente pas de modèle mais une solution envers l'ambiguité.

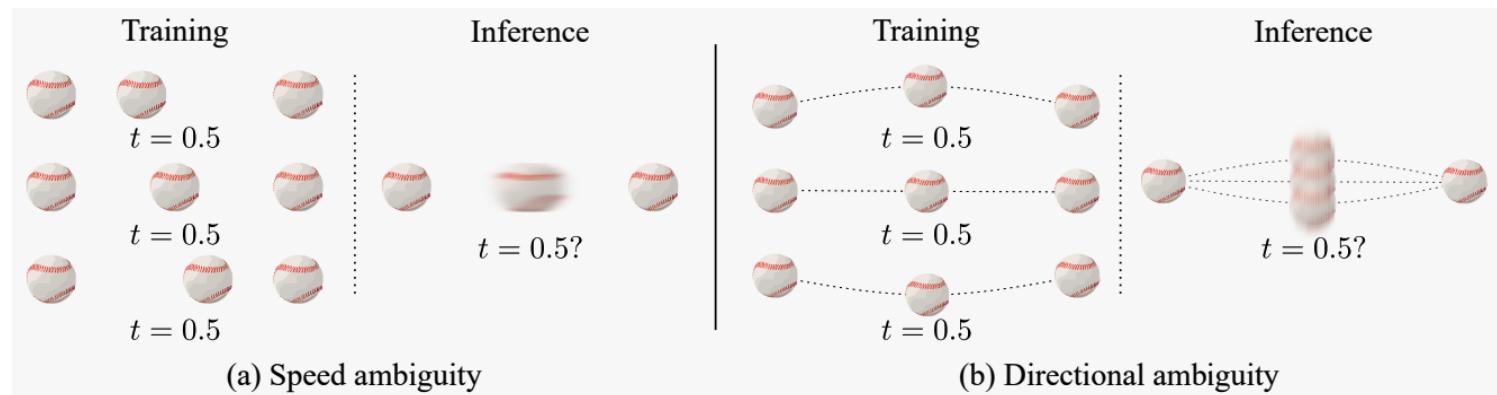
L'ambiguité en régression est un problème qui surgit lorsque la distribution des images plausibles à une **haute variance**

$$I_t^1, I_t^2, \dots, I_t^n \sim \mathcal{F}(I_0, I_1, t)$$

L'output de la régression \hat{I}_t est la moyenne de ces images

$$\hat{I}_t = \mathbb{E}_{I_t \sim \mathcal{F}(I_0, I_1, t)} \{I_t\}$$

Résultant en des images floues, la méthode proposée réduit l'ensemble d'images plausibles en modifiant les inputs.



Clearer Frames, Anytime, la méthode

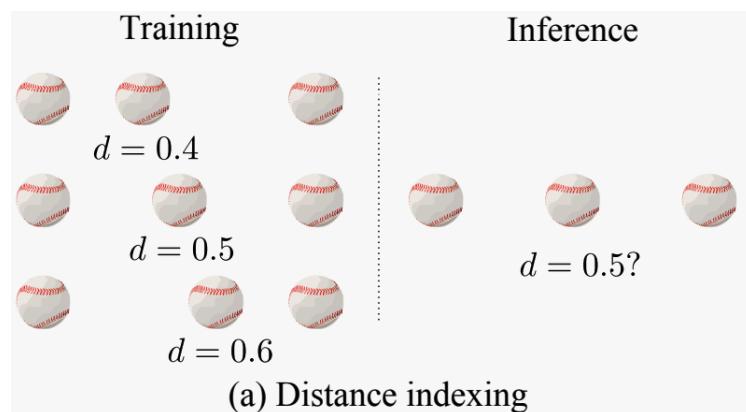
La première approche consiste à indexer l'interpolation par la **distance**, plutôt que le **temps**, résolvant l'ambiguité liée à la vitesse.

$$\mathcal{F}(I_0, I_1, D_t) \text{ plutôt que } \mathcal{F}(I_0, I_1, t)$$

Où D_t associe à chaque pixel un ratio de la distance parcourue entre I_0 , et I_1

- Durant l'**entraînement**, D_t est rigoureusement estimée en se basant sur l'optic flow
- Durant l'**inférence**, il est suffisant de fournir une map uniforme

$$D_t(x, y) = t \quad \forall x, y$$



Clearer Frames, Anytime, la méthode

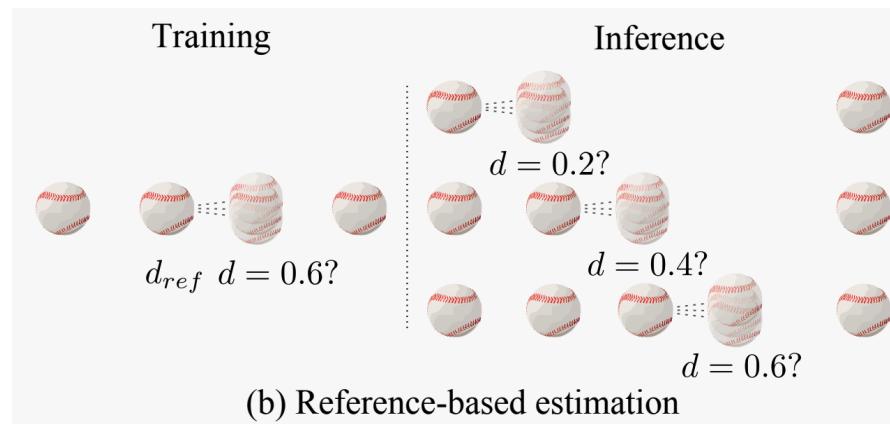
L'article attaque l'ambiguité directionnelle en proposant d'indexer l'interpolation avec une **frame de référence**.

$$I_t = \mathcal{F}(I_0, I_1, D_t, I_{ref}, D_{ref})$$

Par exemple, en séparant l'estimation d'une frame en deux

$$I_{t/2} = \mathcal{F}(I_0, I_1, D_t, I_0, D_0)$$

$$I_t = \mathcal{F}(I_0, I_1, D_t, I_{t/2}, D_{t/2})$$

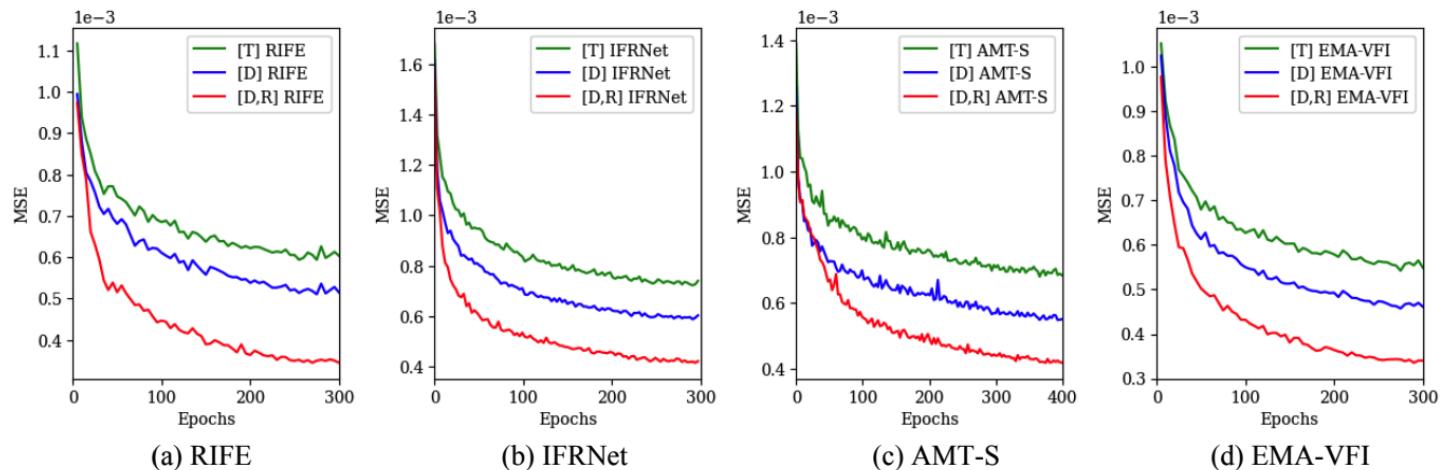


Clearer Frames, Anytime, entrainement

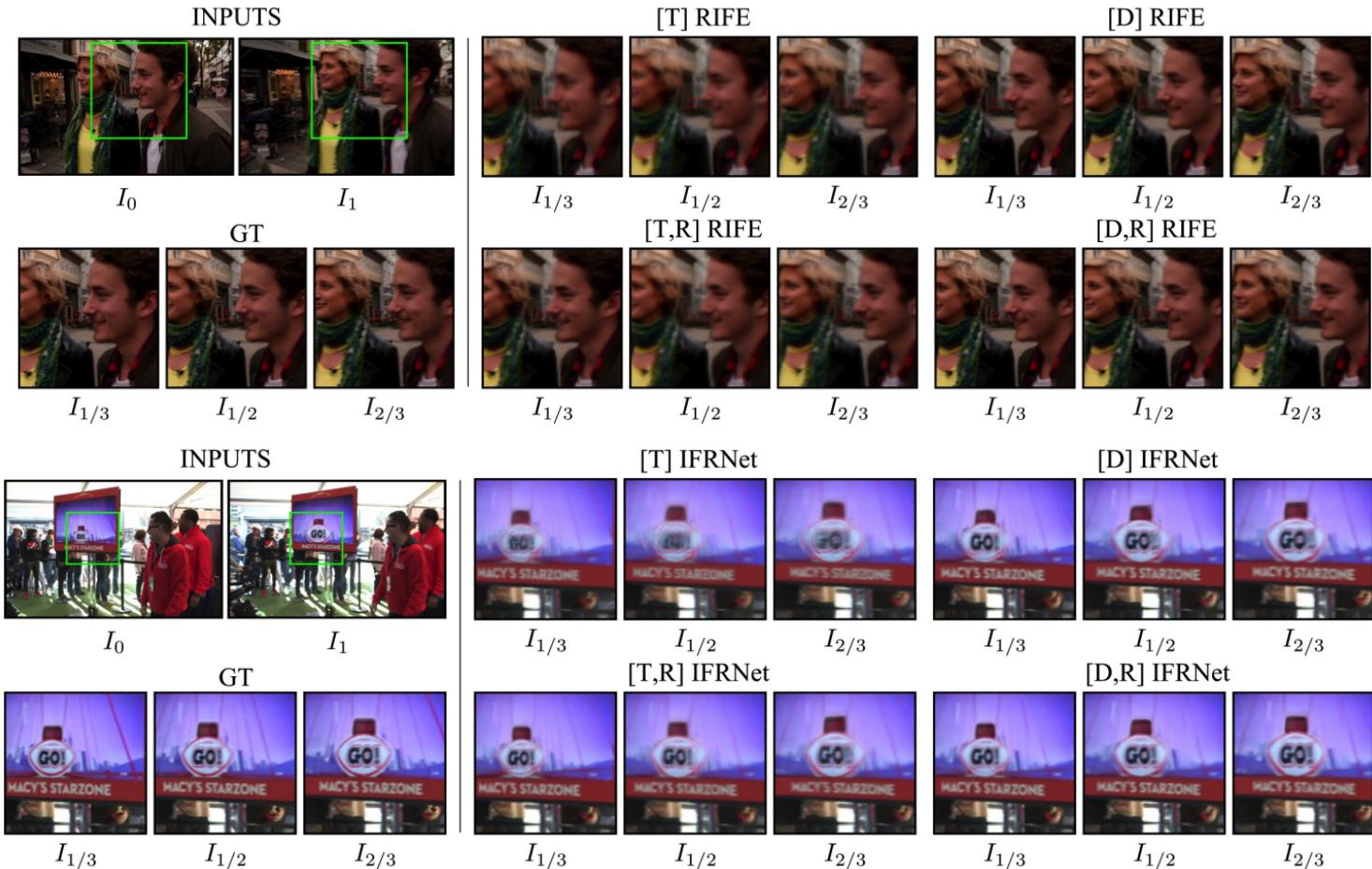
Les stratégies d'indexation sont implémentées sur des modèles où t est arbitraire.

RIFE, IFRNet, AMT, EMA-VFI

Utilise RAFT pour le calcul des distances durant l'entraînement



Clearer Frames, Anytime, résultats



Clearer Frames, Anytime, résultats



Que retenir de ces recherches ?

Récapitulatif

L'analyse des résultats **quantitatifs**.

Model	Vimeo90K	Params (M)	FLOPs (T)	Runtime (s)
CAIN	34.69/0.969	42.8	1.29	0.069
DBVFI	36.17/0.976	15.18	1.28	/
CSPA	36.76/0.980	28.9	/	0.68
IFRNet	35.80/0.9794	5	0.21	0.025
IFRNet small	35.59/0.9786	2.8	0.12	0.019
IFRNet large	36.20/0.9808	19.7	0.79	0.079
UGSP	35.62/-	/	0.016	<u>0.46 (CPU)</u>
UGSP-distill	35.65/-	/	0.015	<u>0.43 (CPU)</u>

Récapitulatif

- De nombreux articles utilisent le **flow**
- Le state of the art a **bien évolué** depuis CAIN
 - Les modèles présentés surpassent tous CAIN d' ≈ 1 dB en PSNR et 0.10 SSIM.
 - CAIN est de loin le modèle le plus **lourd**
 - Certains articles présentent des solutions **plus rapide**
- La recherche propose des solutions plus **directe**
 - Utilisation de loss plus avancées préservant la texture locale (TCL, Census loss)
 - L'utilisation de **l'incertitude** améliore la vitesse et contrôle les ressources
 - L'indexation par distance et référence peut-être une solution pour le **flou**

Prochaine étape

Prochaine étape

- Expérimenter avec les implémentations disponibles
 - <https://github.com/ltkong218/IFRNet>
 - <https://github.com/Oceanlib/DBVI>
 - <https://github.com/zzh-tech/InterpAny-Clearer>
- Considérer l'application de méthodes directes sur la solution actuelle
- Garder la recherche en fil rouge
 - Han, Xu, & al. "Video Frame Interpolation with Region-Distinguishable Priors from SAM" Dec 2023
 - Danier, Zhang, Bull "LDMVFI: Video Frame Interpolation with Latent Diffusion Model" Dec 2023
 - Zhang, Zhu, & al. "Extracting Motion and Appearance via Inter-Frame Attention for Efficient Video Frame Interpolation" Mar 2023
 - ...

References

- Liste des articles
 - Yu, Zhiyang, & al. "Deep Bayesian Video Frame Interpolation." Oct 2022.
 - Choi, Kim, & al. "Channel Attention Is All You Need for Video Frame Interpolation" 2020.
 - Zhou, Li, & al. "Exploring Motion Ambiguity and Alignment for High-Quality Video Frame Interpolation" Mar 2022
 - Kong, Jiang, & al. "IFRNet: Intermediate Feature Refine Network for Efficient Frame Interpolation" May 2022
 - Cheng, Jiang, & al. "Uncertainty-Guided Spatial Pruning Architecture for Efficient Frame Interpolation" Oct 2023
 - Zhong, Krishnan, & al. "Clearer Frames, Anytime: Resolving Velocity Ambiguity in Video Frame Interpolation" Nov 2023

- Autres références

- Hinton, Vinyals, & al. "Distilling the Knowledge in a Neural Network" Mar 2015
- Adler, Ötkem "Solving ill-posed inverse problems using iterative deep neural networks" May 2017
- Andrychowicz, Denil, & al. "Learning to learn by gradient descent by gradient descent" Nov 2016
- Wang, Dong, & al. "Exploring Sparsity in Image Super-Resolution for Efficient Inference" Apr 2021
- Xu, Siyao, & al. "Quadratic Video Interpolation" Nov 2019