VERS UNE TRANSMISSION VIDÉO SANS LATENCE PAR L'EXTRAPOLATION D'IMAGES

Melan VIJAYARATNAM[1], Marco CAGNAZZO[1,2], Giuseppe VALENZISE[3], Anthony TRIOUX[4], Michel KIEFFER[3]

[1]LTCI, Télécom ParisTech, Institut Polytechnique de Paris

[2]University of Padua, Department of Information Engineering, Italy

[3]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes

[4]UMR 8520 - IEMN, DOAE, Univ. Polytechnique Hauts-de-France, CNRS

melan.vijayaratnam@telecom-paris.fr, marco.cagnazzo@telecom-paris.fr
giuseppe.valenzise@l2s.centralesupelec.fr , Anthony.Trioux@uphf.fr
michel.kieffer@l2s.centralesupelec.fr

Résumé – Ces dernières années, plusieurs efforts ont été consacrés à la réduction des différentes sources de latence dans la transmission vidéo, notamment l'acquisition, le codage et la transmission réseau. L'objectif est d'améliorer la qualité d'expérience dans les applications nécessitant une interaction en temps réel. Néanmoins, ces efforts sont fondamentalement contraints par des limites technologiques et physiques. Dans cet article, nous étudions une approche radicalement différente qui peut réduire arbitrairement la latence globale au moyen de l'extrapolation vidéo. Nous proposons deux schémas de compensation de latence dans lesquels l'extrapolation vidéo est effectuée soit au niveau de l'encodeur, soit au niveau du décodeur. Puisqu'une perte de fidélité est le prix à payer pour compenser arbitrairement la latence, nous étudions le compromis latence-fidélité en utilisant trois schémas de prédiction vidéo récents. Nos résultats préliminaires montrent qu'en acceptant une perte de qualité, nous pouvons compenser une latence typique de 100 ms avec une perte de 8 dB en PSNR avec le meilleur extrapolateur. Cette approche est prometteuse mais suggère également que des travaux supplémentaires doivent être réalisés dans le domaine de la prédiction vidéo afin de poursuivre la transmission vidéo sans latence.

Abstract – In the past few years, several efforts have been devoted to reduce individual sources of latency in video delivery, including acquisition, coding and network transmission. The goal is to improve the quality of experience in applications requiring real-time interaction. Nevertheless, these efforts are fundamentally constrained by technological and physical limits. In this paper, we investigate a radically different approach that can arbitrarily reduce the overall latency by means of video extrapolation. We propose two latency compensation schemes where video extrapolation is performed either at the encoder or at the decoder side. Since a loss of fidelity is the price to pay for compensating latency arbitrarily, we study the latency-fidelity compromise using three recent video prediction schemes. Our preliminary results show that by accepting a quality loss, we can compensate a typical latency of 100 ms with a loss of 8 dB in PSNR with the best extrapolator. This approach is promising but also suggests that further work should be done in video prediction to pursue zero-latency video transmission.

1 Introduction

La diffusion de vidéos à très faible latence est une caractéristique essentielle de nombreuses applications impliquant des interactions entre humains (*e.g.*, vidéoconférence, réalité virtuelle et augmentée) ou entre humains et machines (*e.g.*, téléopération de véhicules ou de robots sans pilote, chirurgie à distance, *etc*. Dans ces scénarios, la latence Glass-to-Glass (G2G), entendue comme le délai entre l'acquisition d'une image vidéo par un agent et son affichage par un second agent (distant) [1], joue un rôle majeur dans la qualité globale de l'expérience perçue par les utilisateurs [2]. La latence G2G comprend l'acquisition, le codage, la mise en mémoire tampon à l'émetteur, la transmission, la mise en mémoire tampon au récepteur, le décodage et les délais d'affichage. Au cours des dernières décennies, des efforts importants ont été consacrés à l'optimisation de chacune de ces sources individuelles de délai. Néanmoins, la latence minimale atteignable est toujours limitée par des contraintes technologiques et physiques (la plus évidente étant la vitesse de la lumière), qui représentent une limite inférieure stricte au-delà de laquelle la latence ne peut plus être réduite. En gardant à l'esprit ces limitations claires, nous étudions dans cet article comment nous pouvons réduire davantage la latence du G2G pour qu'elle soit arbitrairement faible. Étant donné les délais physiques incompressibles de la diffusion vidéo, une façon possible d'atteindre cet objectif est de *prédire* les images vidéo qui n'ont pas encore été reçues en utilisant celles déjà disponibles au niveau du récepteur.

L'utilisation de la prédiction/extrapolation pour compenser la latence n'est pas nouvelle. Elle a déjà été employée dans un certain nombre d'applications, notamment la réalité virtuelle [3], les interfaces tactiles [4], le cloud gaming [5], *etc*. Néanmoins,

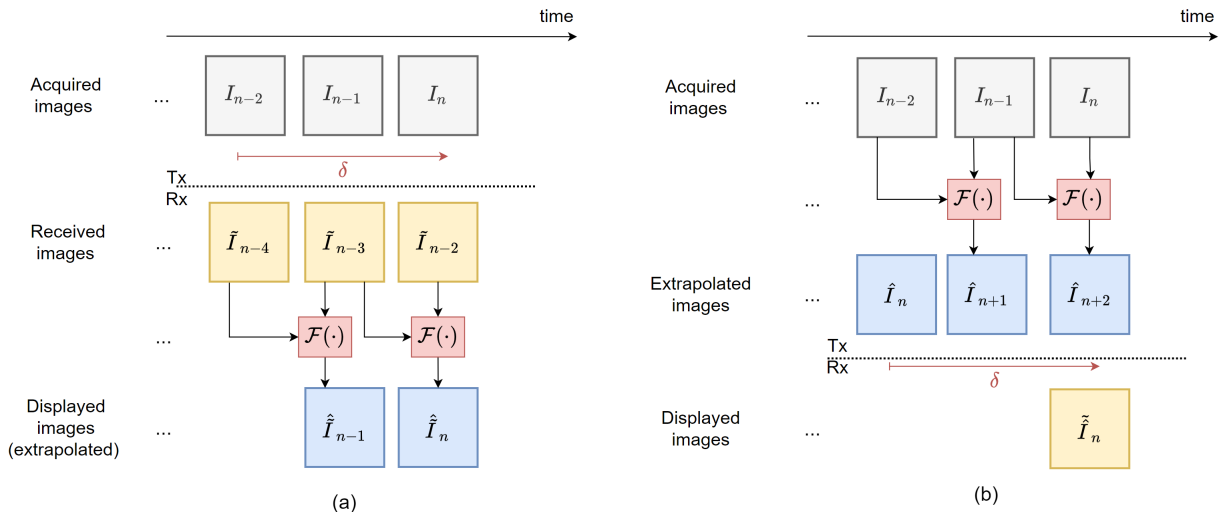

FIGURE 1 – Exemple de compensation de latence avec (a) extrapolation au décodeur et (b) extrapolation à l'encodeur. $\tilde{I}$ et $\hat{I}$ indiquent les images décodées (quantifiées) et prédites (extrapolées), respectivement. $\mathcal{F}$ est la fonction d'extrapolation.

à notre connaissance, cette idée simple d'utiliser l'extrapolation n'a pas encore été explorée et analysée pour le cas de la compensation de la latence dans la transmission vidéo. Dans cet article, nous considérons pour la première fois l'application de l'extrapolation à la communication vidéo à latence ultra-faible ou même *nulle*. À cette fin, nous analysons deux schémas possibles pour intégrer l'extrapolation à un codec vidéo conventionnel, *i.e.*, en extrapolant à partir des images originales du côté de l'encodeur avant le codage et la transmission, ou du côté du décodeur sur la base des décodées (quantifiées) disponibles.

Comme les images extrapolées sont en général différentes des images réelles, la compensation du temps de latence entraîne une perte de fidélité. Dans cet article, en utilisant différents algorithmes récents d'extrapolation basés sur l'apprentissage, nous caractérisons ce compromis entre latence et fidélité. Notre objectif est d'étudier la la faisabilité de la compensation de latence basée sur l'extrapolation d'images.

2 Compensation de latence par extrapolation

2.1 Extrapolation au décodeur

Dans ce schéma, aucune modification de l'émetteur n'est requise par rapport à un pipeline de transmission standard : les images acquises sont compressées et transmises au récepteur, où elles arrivent avec une certaine latence. La figure 1(a) illustre l'extrapolation côté décodeur à l'aide d'un exemple. Dans cette figure, I_n représente la n-ème image de la vidéo. Nous supposons dans cet exemple que la latence G2G δ est égale à deux images (ce qui correspond à un temps de $2/f$ secondes, où f est le nombre d'image par seconde ou plus simple : le frame rate). Le décodeur reçoit donc $\tilde{I}_{n-2}$ (la version compressée de I_{n-2}) alors que l'encodeur est déjà en train d'acquérir I_n. Afin de compenser cette latence, le décodeur exécute un algorithme d'extrapolation d'image $\mathcal{F}$ qui prend en entrée un nombre donné k de images décodées disponibles, et produit une prédiction $\hat{I}_n$ de l'image n comme :

$$\hat{\tilde{I}}_n = \mathcal{F}(\{\tilde{I}_{n-h}, \tilde{I}_{n-h-1}, \ldots, \tilde{I}_{n-h-k+1}\}; h). \quad (1)$$

Les images d'entrée k sont également appelées images*contexte*. Par exemple, dans la figure 1, $k = 2$ images de contexte sont considérées. h est l'*horizon temporel* du mécanisme d'extrapolation. Il détermine la quantité de latence que nous voulons compenser. Plus h est grand, plus il est difficile d'obtenir une prédiction fiable, comme nous le montrons dans les résultats expérimentaux. dans le tableau 1. Le paramètre h détermine le compromis entre la compensation de la latence et la dégradation de la qualité. Dans l'exemple, nous avons $h = 2$, ce qui signifie que nous voulons compenser entièrement la latence δ. Avec ces paramètres, l'algorithme d'extrapolation produit :

$$\hat{\tilde{I}}_n = \mathcal{F}(\{\tilde{I}_{n-2}, \tilde{I}_{n-3}\}; 2). \quad (2)$$

Par conséquent, l'estimation $\hat{\tilde{I}}_n$ de l'image I_n, extrapolée à partir des images compressées, peut être affiché au niveau du décodeur *lorsque* l'image I_n est acquise par l'encodeur.

2.2 Extrapolation à l'encodeur

L'extrapolation côté encodeur est illustrée sur la figure 1(b), suivant l'exemple de la section précédente. La méthode d'extrapolation $\mathcal{F}(\cdot; h)$, est utilisée cette fois avec les images acquises : $I_n, I_{n-1}, \ldots, I_{n-k+1}$ utilisées comme images de contexte. De même, la prédiction dépend de l'horizon temporel h, *i.e.*, nous calculons

$$\hat{I}(n+h) = \mathcal{F}(\{I_n, I_{n-1}, \ldots, I_{n-k}\}; h). \quad (3)$$

Comme pour l'exemple précédent, on a k=2 et h=2 ici. Ensuite, les images extrapolées sont compressées, transmises,

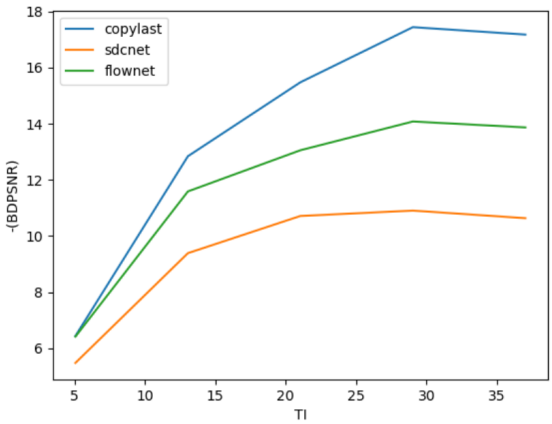

FIGURE 2 – Analyse de l'information temporelle (TI)

décodées et affichées. Dans ce cas, seul l'émetteur doit être adapté et un récepteur standard ordinaire est utilisé. L'extrapolation effectuée au niveau de l'encodeur permet de compenser de manière préventive la latence de transmission. Dans l'exemple considéré, l'image extrapolée et compressée $\tilde{\hat{I}}_n$ est affichée *lorsque* l'image I_n est acquise par l'émetteur.

3 Expériences et discussion

Cette section évalue la faisabilité de l'extrapolation d'image comme outil efficace de compensation de la latence. Plus précisément, nous caractérisons la perte de qualité pour différentes valeurs de latence compensée, en utilisant différents extrapolateurs disponibles pour implémenter $\mathcal{F}$. Toutes les expériences sont menées à l'aide du codec HEVC HM (16.24) [6] en utilisant la configuration encoder_intra_main.cfg.

3.1 Choix des méthodes d'extrapolation

FlowNet-2 [7] est une technique basée sur le mouvement combinée à une opération de déformation, qui peut prédire efficacement la prochaine image. **MCNet** [8] est une technique basée sur le pixel construite sur le réseau de neurones convolutif-LSTM pour la prédiction au niveau du pixel seulement. Enfin, **SDCNet** [9] est une approche basée sur la fusion de ces deux techniques qui modélise les apparences futures à l'aide de deux réseaux convolutifs.

3.2 Étude du compromis distorsion-fidélité

Pour évaluer le schéma où l'extrapolation est effectuée au niveau du décodeur, l'ensemble de test DriveSeg partitionné à 30 fps est compressé en utilisant HEVC avec différents paramètres de quantification $QP \in \{22, 27, 32, 37\}$ pour toutes les méthodes sélectionnées. Les points de débit-distorsion (RD) sont calculés en considérant un horizon d'extrapolation $h \in \{1, \ldots, 5\}$ pour compenser une latence $\delta \in \{33\text{ ms}, 67\text{ ms}, 100\text{ ms}, 133\text{ ms}, 167\text{ ms}\}$. Pour le cas de l'extrapolation au niveau de l'encodeur, le même jeu de test est considéré, avec le même horizon $h \in \{1, \ldots, 5\}$ et les mêmes QP pour la compression des images extrapolées avec HEVC.

La distorsion entre les images affichées et les images originales est évaluée à l'aide de trois mesures objectives : PSNR dans l'espace couleur YCbCr, SSIM, et VMAF. On obtient ainsi une courbe débit-distorsion pour chaque horizon d'extrapolation (compensation de latence) possible h. Nous comparons toutes les courbes au cas $h = 0$, *i.e*, sans aucune extrapolation. Les pertes de qualité moyennes sur différents débits, exprimées par la métrique delta négative de Bjøntegaard (BDPSNR, BDSSIM et BDVMAF), sont indiquées dans le tableau 1, où les premiers chiffres sont les valeurs de la métrique BD pour l'extrapolation du côté du décodeur, et les chiffres entre parenthèses sont les différences par rapport à ce schéma lorsqu'on utilise l'extrapolation au niveau de l'encodeur. Nous pouvons observer que l'extrapolation au niveau de l'encodeur/décodeur produit essentiellement les mêmes pertes de qualité pour une valeur donnée de latence compensée.

3.3 Effets sur l'information temporelle

Pour évaluer l'impact de la complexité temporelle sur l'extrapolation (ou sur le compromis fidélité-latence), nous utilisons l'index temporel (TI) largement utilisé dans la communauté traitement vidéo. Nous étudions ici l'effet de la performance des réseaux d'extrapolation en fonction de l'information temporelle. Pour ce faire, nous découpons l'ensemble de la base de test de Caltech en clips de 150 images, ce qui donne 789 séquences. Nous calculons à la fois la qualité et l'information temporelle et observons comment les réseaux se comportent sous cet angle. La figure 2 met globalement en évidence une augmentation de la distorsion proportionnelle à la valeur du TI.

3.4 Discussion

Les mesures de fidélité seules pourraient ne pas être suffisantes pour expliquer la faisabilité de la compensation de la latence par extrapolation. La figure 3 montre les résultats de la compensation de 100 ms sur une image de la base de données Kitti, qui est similaire à la base d'entraînement. L'interpolation de SDCNet présente de nettes déformations visuelles par rapport à l'image originale. Néanmoins, nous pouvons observer que la position du vélo est approximativement alignée avec l'image originale. En revanche, Copylast (qui n'utilise que la dernière image disponible et qui correspond à une absence de compensation de latence) produit un décalage entre la position réelle du cycliste et celle affichée. On peut imaginer qu'en fonction de l'application (par exemple, la téléopération), la prédiction SDCNet apporte des informations pertinentes (la position du vélo), même si des mesures de qualité objectives telles que le PSNR ne sont pas en capacité de mesurer ces aspects. Cet exemple met en évidence l'objectif essentiel de la réduction de la latence, à savoir d'acquérir davantage de connaissances

	−BDPSNR ↓			−BDSSIM ↓			−BDVMAF ↓		
latency (ms)	**33**	**100**	**167**	**33**	**100**	**167**	**33**	**100**	**167**
Copylast	10.78 (+0.02)	14.21 (+0.02)	15.61 (+0.01)	0.19 (+0.00)	0.35 (+0.00)	0.43 (+0.00)	46.52 (-0.14)	65.44 (-0.09)	71.65 (-0.07)
MCNet	8.34 (-0.15)	12.26 (-0.28)	14.79 (-0.29)	0.06 (+0.00)	0.17 (-0.01)	0.28 (-0.02)	30.28 (-0.94)	50.86 (-2.49)	64.54 (-3.08)
FlowNet-2 + warp	7.35 (+0.03)	11.23 (-0.03)	13.17 (-0.04)	0.05 (+0.00)	0.15 (+0.00)	0.24 (+0.00)	25.42 (-0.20)	51.15 (-0.03)	65.29 (+0.23)
SDCNet	**5.20 (+0.14)**	**8.28 (+0.10)**	**9.74 (+0.09)**	**0.05 (-0.02)**	**0.15 (-0.07)**	**0.24 (-0.13)**	**17.27 (-0.13)**	**36.00 (-0.13)**	**46.63 (-0.09)**

TABLE 1 – Résultats quantitatifs sur la scène DriveSeg : résultats pour l'extrapolation côté décodeur et gain/perte (entre parenthèses) obtenus avec le schéma d'extrapolation côté encodeur.

sur la représentation future de la scène dynamique en anticipant les images suivantes. Cette question est essentielle non seulement au niveau perceptif humain, mais aussi au niveau machine. Nous pouvons imaginer cette compréhension future comme un élément clé pour que la machine puisse anticiper les comportements.

FIGURE 3 – Résultats qualitatifs pour une compensation de latence de 100 ms. Le schéma d'extrapolation côté décodeur est utilisé. Le codec HEVC utilise un paramètre de quantification $QP = 32$. Images tirées de la base de données Kitti.

4 Conclusion

Cet article présente un outil qui permet de compenser la latence dans un schéma de transmission vidéo au prix d'une complexité supplémentaire mais aussi d'une dégradation de la fidélité de l'image. Selon l'application, plusieurs configurations sont possibles : extrapolation au niveau de l'encodeur, extrapolation au niveau du décodeur, ou les deux au niveau de l'encodeur et du décodeur. La dégradation est essentiellement causée par les approches d'extrapolation. L'objectif de cet article n'est pas de proposer une meilleure approche d'extrapolation pour réduire cette perte, mais plutôt de démontrer l'applicabilité de la compensation de latence reposant sur un tel outil. Les travaux futurs pourraient concerner l'amélioration des méthodes d'extrapolation existantes pour cette tâche.

5 Acknowledgments

Ce travail a été financé par le fonds national ANR AAPG2020 (ANR-20-CE25-0014).

Références

[1] C. Bachhuber, E. Steinbach, M. Freundl, and M. Reisslein, "On the Minimization of Glass-to-Glass and Glass-to-Algorithm Delay in Video Communication," *IEEE Trans. on Multimedia*, vol. 20, no. 1, pp. 238–252, Jan. 2018.

[2] K. Brunnström, E. Dima, T. Qureshi, M. Johanson, M. Andersson, and M. Sjöström, "Latency impact on quality of experience in a virtual reality simulator for remote control of machines," *Signal Processing : Image Communication*, vol. 89, pp. 116005, 2020.

[3] A. Garcia-Agundez, A. Westmeier, P. Caserman, R. Konrad, and S. Göbel, "An evaluation of extrapolation and filtering techniques in head tracking for virtual environments to reduce cybersickness," in *Joint International Conference on Serious Games*. Springer, 2017, pp. 203–211.

[4] N. Henze, M. Funk, and A. S. Shirazi, "Software-reduced touchscreen latency," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2016, pp. 434–441.

[5] K. Lee, D. Chu, E. Cuervo, J. Kopf, Y. Degtyarev, S. Grizan, A. Wolman, and J. Flinn, "Outatime : Using speculation to enable low-latency continuous interaction for mobile cloud gaming," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 2015, pp. 151–165.

[6] C. Rosewarne, K. Sharman, R. Sjöberg, and G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description Update 13 | MPEG," in *38th JCT-VC Meeting*, Brussels, Jan. 2020.

[7] E. Ilg, N. Mayer, T. Saikia, et al., "Flownet 2.0 : Evolution of optical flow estimation with deep networks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2462–2470.

[8] R. Villegas, J. Yang, S. Hong, et al., "Decomposing motion and content for natural video sequence prediction," *arXiv preprint arXiv :1706.08033*, 2017.

[9] F. A. Reda, G. Liu, K. J. Shih, et al., "SDCNet : Video prediction using spatially-displaced convolution," 2021.