Interpolación de Fotogramas de Vídeo (VFI): Un Análisis en el Contexto de la Animación 2D

Victor Manuel Fonte Chavez victor.fonte@cimat.mx
Claudia Esteves Jaramillo cesteves@cimat.mx
Jean Bernard Hayet jbhayet@cimat.mx

7 de octubre de 2023

Resumen

La animación 2D tradicional requiere mucha mano de obra y a menudo requiere que animadores dibujen manualmente doce ilustraciones por segundo de movimiento. Si bien la interpolación automática de fotogramas puede aliviar esta carga, la animación 2D plantea dificultades adicionales en comparación con el vídeo fotorrealista. En este trabajo en curso, mediante modelos de difusión del estado del arte, intentamos solventar estas deficiencias enfocándonos en mejorar la calidad de percepción de las imágenes interpoladas. Proponemos un método de VFI basado en modelos de difusión latente, LDMVFI. Esto aborda el problema del VFI desde una perspectiva generativa formulándolo como un problema de generación condicional.

1. Introducción

Los animadores 2D tradicionales suelen dibujar cada cuadro manualmente; este proceso es increíblemente intensivo en laboratorio, y requiere grandes equipos de producción con capacitación experta para dibujar y colorear las decenas de miles de ilustraciones necesarias para un serie animada. Con la creciente popularidad mundial del estilo tradicional, los estudios se ven en apuros para ofrecer grandes volúmenes de contenido de calidad. Los avances recientes en visión por computadora y gráficos pueden reducir la carga sobre animadores. En el contexto de la animación, el animador podría potencialmente lograr la misma velocidad de fotogramas para una secuencia, dibujando manualmente sólo una fracción de los fotogramas y utilizando un interpolador para generar el resto. Cabe destacar que este proyecto esta en desarrollo y por lo tanto sus resultados aun son escasos.

2. Metodología

2.1. Base de datos

Para llevar a cabo nuestra investigación, primeramente utilizaremos la base de datos ATD-12K Dataset [1], que comprende un conjunto de entrenamiento que contiene 10.000 tripletas de fotogramas de animación (en tiempos consecutivos) y un conjunto de prueba que contiene 2000 tripletas, recopilados de una variedad de películas de dibujos animados.

A diferencia del dominio del vídeo natural, donde, en casi todos los casos, tres fotogramas consecutivos de un corte se pueden utilizar como triplete de entrenamiento, la recopilación de datos para la animación 2D es mucho más ambigua. Los animadores suelen dibujar a velocidades de cuadro variables, con movimientos expresivos en forma de arco; cuando se combinan con altos desplazamientos de píxeles, estos resultan en una cantidad significativa

de tripletas con movimiento no lineal o espaciadas de forma desigual. Para evitar que el modelo se pierda en estas características no lineales, se propone un filtrado de la base de datos mediante la métrica RRLD [2].

2.2. Modelo Propuesto

Dados dos frames consecutivos I^0 e I^1 de un vídeo, el objetivo del VFI es generar el frame no existente I^n cuando $n = 0{,}5$. Nuestro objetivo es aproximar mediante un modelo de difusión latente (LDM)[3], la distribución $p(I^n|I^0, I^1)$. Específicamente, nos basamos en el LDMVFI [6] para llegar a este objetivo. Este modelo se compone de dos parte: (i) Codificador, que se encarga de llevar un frame a un espacio latente y viceversa y (ii) Modelo de Difusión parametrizada mediante una **U-Net**, la cual se encarga de realizar el proceso inverso de la difusión para generar la imagen intermedia dentro de este espacio latente. Para el Codificador utilizamos un modelo **VQ-VAE** (Vector Quantized Variational Autoencoder)[7][9]. Este modelo lleva cada frame a un espacio latente discreto, codificado mediante un Codebook aprendible. Tanto en el bloque codificador como en el decodificador de este **VQ-VAE**, se utilizan como método de atención NonLocalBlocks[8] previo a bloques convolucionales residuales que rebajan a la mitad o aumentan el doble la imagen. Luego de preentrenar la **VQ=VAE**, en el espacio latente de dimensión $16 \times 16 \times 3$, se crea un **DDPM**[4] con una **U-Net** que realizará el proceso de difusión inverso. Este modelo está generado mediante 4 bloques que contienen capas residuales y, en cada uno de ellos, una capa de self attention clásica. Todo la programación del modelo y su entrenamiento fue utilizando el paquete `pytorch`[5].

3. Bibliografía

1. Siyao, L., Zhao, S., Yu, W., Sun, W., Metaxas, D., Loy, C.C., Liu, Z.: Deep animation video interpolation in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6587–6595 (2021)

2. Chen, S., & Zwicker, M. (2022). Improving the Perceptual Quality of 2D Animation Interpolation. En ECCV 2022

3. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. arXiv:2112.10752.

4. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. arXiv preprint arXiv:2006.11239.

5. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. K¨opf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," CoRR, vol. abs/1912.01703, 2019. [Online]. Available: http://arxiv.org/abs/1912.01703

6. Danier, D., Zhang, F., & Bull, D. (2023). LDMVFI: Video Frame Interpolation with Latent Diffusion Models. arXiv preprint arXiv:2303.09508.

7. van den Oord, A., Vinyals, O., & Kavukcuoglu, K. (2018). Neural Discrete Representation Learning. arXiv preprint arXiv:1711.00937.

8. Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local Neural Networks. arXiv preprint arXiv:1711.07971.

9. Esser, P., Rombach, R., & Ommer, B. (2021). Taming Transformers for High-Resolution Image Synthesis. arXiv preprint arXiv:2012.09841.