디퓨전 모델과 텍스트 입력을 활용한 키포인트 기반 동영상 편집 기법에 관한 연구

강동경, 황재연, 장혜령*

동국대학교

kdk2146@dgu.ac.kr, khanman409@dgu.ac.kr, hyeryung.jang@dgu.ac.kr

Keypoint-driven Video Editing Method using Diffusion Models and Text Prompts

Dongkyung Kang, Jaeyeon Hwang, Hyeryung Jang*

Dongguk Univ.

요 약

Diffusion Model을 기반으로 하는 이미지 생성 작업은 컴퓨터 비전 분야에서 중요한 문제로 자리 잡고 있다. 여기에 더해, Text를 추가 입력 값으로 받아 조건부로 이미지를 생성 및 편집하는 작업 역시 많은 관심을 끌고 있다. 본 논문의 시스템은 Keypoint Model을 Text와 함께 사용하여 변형을 주고 싶은 부분을 추출하고, 이를 원하는 방향으로 편집할 수 있도록 Drag 기술을 사용하였다. 최종적으로 이 과정을 반복적으로 진행한 뒤 영상으로 제작한다. 또한, 기존 방법들의 문제점인 이미지 깨짐 현상을 개선하기 위해서, Controlnet의 Unet구조를 참고하여 zero conv block을 추가함으로써 성능을 개선하였다. 이로써 본 논문에서 제안한 방법은 주어진 이미지에 대하여 Text를 통한 직관적인 편집을 가능하게 하며, 실험을 통해 원본에 대한 복원율을 높인 영상을 만들어냄을 확인하였다.

Ⅰ. 서 론

최근 Diffusion model 기반의 이미지 생성 기술이 크게 주목받고 있으며, Text를 추가 조건으로 이미지 생성 및 편집을 수행하는 방법이 활발히 연구되고 있다[1]. 하지만, 기존의 이미지 편집 방식은 text-prompt에 의존해 의미론적인 접근을 취하므로 이미지의 공간적인 영역의 직접적인 조작이 어렵다. 이를 해결하기 위해 사용자가 원하는 만큼 원본 이미지의 공간적인 변형을 주어 편집을 수행하는 drag 기반의 방법[3][4][5]이 제안되었다. 본 논문에서는 keypoint detection 모델을 도입하여, 직접 클릭하는 drag input 대신 text input을 사용하여 조건부로 이미지를 편집하는 방법을 제시하고자 한다. 이러한 편집된 이미지를 여러 장 생성한 뒤 이를 video의 프레임 단위로 삼아서 1차 동영상을 만들고, 동영상 보간 모델을 적용해서 프레임 수를 늘려 좀 더 자연스러운 최종 동영상을 만든다. 이는 text를 통한 직관적인 편집을 가능하게 하고, zero conv를 사용해 원본에 대한 복원율을 보완한 자연스러운 비디오 생성을 목표로 한다. 실험을 통해, LPIPS 지표를 사용하여 프레임 간 시간적 연속성을 평가하고, GScore, SSIM, PSNR 지표를 사용해 편집 이미지의 자연스러움을 평가해 성능 향상을 검증하였다.

Ⅱ. 배경지식

Ⅱ.1 Drag-based Editing

그림 1. DragDiffusion[3]의 예시

Drag 기반 편집 기법은 diffusion 모델에도 적용되고 있다[2][3]. 그림 1의 예시와 같이, 사용자는 일정 픽셀 영역으로 구성된 handle point(빨간색)와 target point(파란색)을 이미지와 함께 입력한다. Drag 과정은 motion supervision과 point tracking으로 구성된다. Motion supervision은 handle point가 target point 방향으로 이동하는 과정으로, 두 point 사이의 거리를 loss로 정의하고 gradient descent 를 반복하여 handle point가 target point에 점차 가까워지도록 한다. 다음으로 point tracking이란, motion supervision으로 인해 위치가 변경된 handle point를 nearest neighbor search 방식을 통해 재설정하는 과정이다. Drag 이후 denoising을 진행하는 DragDiffusion[2]과 달리, GoodDrag[3]는 drag 과정 사이에 denoising을 반복적으로 수행함으로써 오차가 과도하게 누적되는 것을 억제하는 효과를 얻을 수 있다.

Ⅱ.2 Keypoint Detection

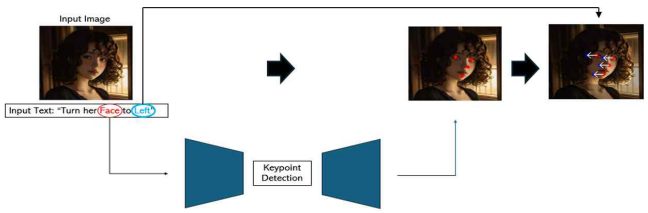

그림 2. X-Pose를 Drag에 접목시킨 방법의 예시

X-Pose[5]는 LLM구조를 일부 차용하여 학습함으로써, text prompt를 입력받아 기존 방법들 대비 다양한 객체의 키포인트를 감지하는 keypoint detection 모델이다. 또한, output keypoint는 pixel-based의 좌표 값으로 제공되어, drag 기반의 방법과의 접목 가능성이 높다. 본 논문에서는 text input이 사용자에게 좀 더 직관적이라는 가정 하에, X-Pose를 사용하여 drag input을 text input으로 대체하고자 한다. 그림 2는 X-Pose를 drag에 적용시킨 동작 예시이다. 사용자는 움직이고 싶은 물체와 방향을 text로 입력한다. 이로부터 keypoint 모델의 label, 예를 들어 Face, Car, Chair 등에 해당하는 값을 추출해 handle point로 지정하고, 방향에 해당하는 단어를 검출하여 target point로 지정한다.

Ⅱ.3 Video Interpolation

Video frame interpolation 모델인 LDMVFI[6]는 연속된 프레임 사이에 자연스러운 중간 프레임을 생성하여 영상의 프레임 수를 늘리는 방법으로, 본 논문에서도 이를 사용하여 영상의 자연스러움을 향상시킨다.

Ⅲ. 본론

Ⅲ.1 전체 파이프라인

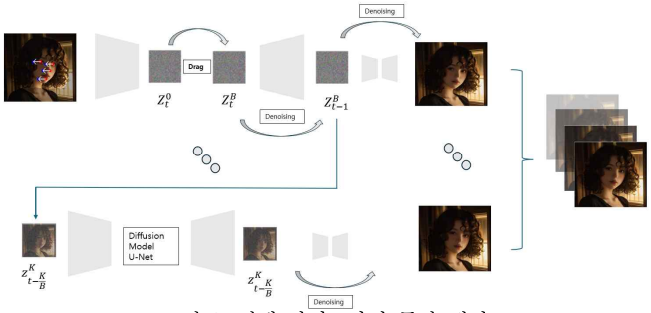

그림 3. 전체 파이프라인 동작 예시

본 논문에서 제안하는 방법론(그림 3)은 X-Pose를 통해 사용자 입력 text로부터 handle point와 target point를 추출하는 과정(그림 2)을 첫 단계로 한다. 이후 GoodDrag 구조를 기반으로 drag 과정을 K번 수행하되, handle point에서 target point로 이동하는 동안 $B(B<K)$번마다 denoising을 수행하여 중간 프레임 이미지를 하나씩 생성한다. 생성된 총 K/B장의 이미지를 프레임 단위로 합쳐 1차 영상을 만든 후 LDMVFI를 사용하여 영상의 프레임 수를 약 2배로 늘려 최종 동영상을 생성한다.

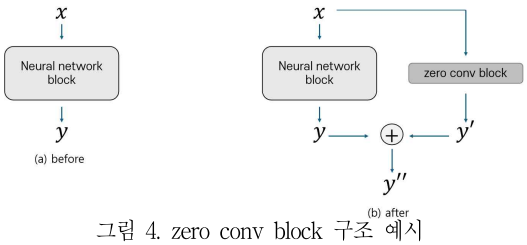

그림 4. zero conv block 구조 예시

단, GoodDrag의 구조를 그대로 사용할 경우 이미지 깨짐 현상이 나타났다. 이를 해결하기 위해 원본 이미지에 대한 복원율을 높인 ControlNet[4]의 U-Net구조에 영감을 받아, 본 논문에서도 기존 U-Net 구조의 mid와 up block에 zero conv block을 추가로 도입하였다(그림 4). 그림 4(b)에서 왼쪽은 각 drag 과정을 처리하는 부분이고, 오른쪽은 원본의 이미지에 대한 정보를 그대로 유지하는 부분이다. 오른쪽 결괏값 y'은 drag 과정을 거치지 않으며, 이를 y와 합쳐 다음 단계의 입력으로 사용한다. 이전 모델 구조 대비 발생하는 오차의 절대적인 크기는 같지만, 전체적으로 차지하는 비중 즉, 상대적인 크기가 작아지면서, 원본 이미지에 대한 충실성을 높이고, 오차가 중화되며 이미지 깨짐 현상을 완화할 수 있었다.

Ⅲ.2 실험 결과 및 검증

실험에 사용한 입력 이미지의 크기는 512x512이다. K값으로 drag 과정 수행 횟수를 결정하고, B값을 조정해 중간 이미지 개수(K/B)를 정한다. 이미지가 많을수록 더욱 자연스러운 영상이 만들어질 수 있지만, 반면 drag 과정을 통해 누적되는 오차가 커지므로 적절한 값 설정이 필요하다. 우리는 실험적으로 K/B값이 20장~35장일 때 이미지 깨짐 현상이 적음을 확인하였고, motion supervision에서 매 step 마다 움직이는 거리, 각 point의 영역 크기 등을 최적화하여 자연스러운 결과물이 나오도록 했다.

그림 5. zero conv block 적용 예시

위 그림 5는 zero conv를 적용하지 않은 경우(가운데)와 적용한 경우(오른쪽)의 예시 결과물이다. Zero conv를 적용하지 않았을 때 중앙 부분의 가드레일에서 이미지 깨짐 현상이 발생함을 확인하였다. 이 두 개의 사진을 비교할 때, zero conv를 적용한 구조가 원본의 입력값을 유지하는 역할을 하여 이미지 깨짐 현상을 완화하고 입력 데이터의 특성을 유지하는 역할을 수행함을 실험으로 확인하였다.

Method	LPIPS ↓
w/o zero conv	0.0235
ours	**0.0217**

표 1. 연속된 두 프레임 간의 LPIPS 지표 평가

표 1은 두 이미지 간의 유사도를 측정할 수 있는 LPIPS를 사용하여 영상 속 연속된 두 프레임 간의 시간적인 연속성을 평가한 결과이다. 그림 5의 예시를 포함하여 다른 5개의 추가 예시에서 영상을 만들고 LPIPS를 구하여 평균을 낸 결과이며, zero conv를 사용함으로써 평균적으로 약 0.0018 정도 향상된 결과를 확인하였다.

Method	GScore ↑	SSIM ↑	PSNR ↑
DragDiffusion	6.6	0.8808	38.2302
w/o zero conv	6.0	0.9016	40.0462
ours	**8.4**	**0.9092**	**40.4728**

표 2. 세 모델 간의 GScore, SSIM, PSNR 지표 비교

표 2에서는 추가적으로 원본에 대한 결과물 이미지의 깨짐 현상을 평가하기 위해 GScore[3], SSIM, PSNR을 사용하였다. PSNR은 이미지 화질 손실양을, SSIM은 두 이미지 간의 유사성을 평가하는 지표이다. LPIPS 평가의 2개 예시를 포함, 3개를 추가하여 더 다양한 물체에 대한 평가를 진행했다. 제안한 방법론은 8.4의 GScore 값을 가짐으로써 DragDiffusion에 비해 1.8, GoodDrag (w/o zero conv) 보다 2.4 더 높은 값을 보였다. SSIM에서는 0.9092를 기록하여 DragDiffusion 보다 0.0284, GoodDrag 보다 0.0076 향상된 결과를 얻었다. 마지막으로, PSNR에서도 DragDiffusion 보다 2.2426, GoodDrag보다 0.4266 더 높은 수치를 기록했다. 이를 통해 기존의 방법과 비교하여 zero conv를 적용한 구조의 이미지가 더욱 자연스럽게 나타났음을 확인하였다.

Ⅳ. 결론

본 논문은 text 입력을 사용해 직관적으로 diffusion 모델 기반의 이미지 편집을 수행하고, 이를 기반으로 동영상을 생성하는 방법을 제시하였다. Zero conv 블록을 도입해 이미지 깨짐 문제를 해결함으로써, 기존의 drag 기반 방법보다 원본 이미지의 충실성을 높일 수 있었고, 동영상 보간 기법을 통해 자연스러운 성능을 얻을 수 있었다. 향후 연구에서는 text embedding을 활용하여 프레임 간의 자연스러움과 영상의 풍부함을 개선하는 방안을 수행하고자 한다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업(IITP-2024-2020-0-01789)과 2024년 SW중심대학사업(2023-0-00049)의 지원을 받아 수행되었음.

참 고 문 헌

[1] B. Kawar et al., “Imagic: Text-based real image editing with diffusion models”, arXiv preprint arXiv:2210.09276, 2022.

[2] Y. Shi et al., ”DragDiffusion: Harnessing diffusion models for interactive point-based image editing”, IEEE/CVF CVPR, pp. 8839-8849, 2024.

[3] Z. Zhang et al., “GoodDrag: Towards good practices for drag editing with diffusion models”, arXiv preprint arXiv:2404.07206, 2024.

[4] L. Zhang et al., “Adding Conditional Control to Text-to-Image Diffusion Models”, IEEE/CVF ICCV, pp. 3813-3824, 2023.

[5] J. Yang et al., “X-Pose: Detecting any Keypoints”, arXiv preprint arXiv:2310.08530, 2023.

[6] D. Danier et al., “Ldmvfi: Video frame interpolation with latent diffusion models”, AAAI, Vol. 38, No. 2, pp. 1472-1480, 2024