GaussianPrediction: Dynamic 3D Gaussian Prediction for Motion Extrapolation and Free View Synthesis

by

김유겸

2024.07.04

Index

Index

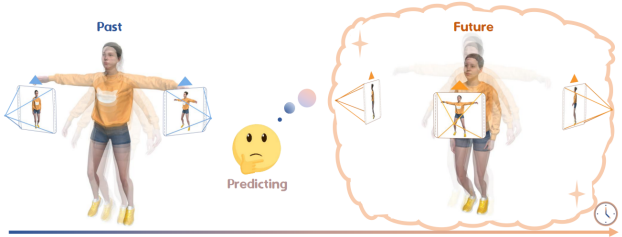
주제

- 미래의 scenario를 예측하는 것은 컴퓨터비전, 로보틱스 등 여러 분야에서 중요
  - Intelligent decision-making 등에서 필수적
- video prediction의 경우 시점 하나를 기준으로만 예측할 수 있었고, novel-view synthesis는 다양한 시점에서의 이미지를 렌더할 수 있었으나 시간에 따른 예측을 할 수 없었음

Video Prediction

Novel-view Synthesis

주제



GaussianPrediction 프레임워크

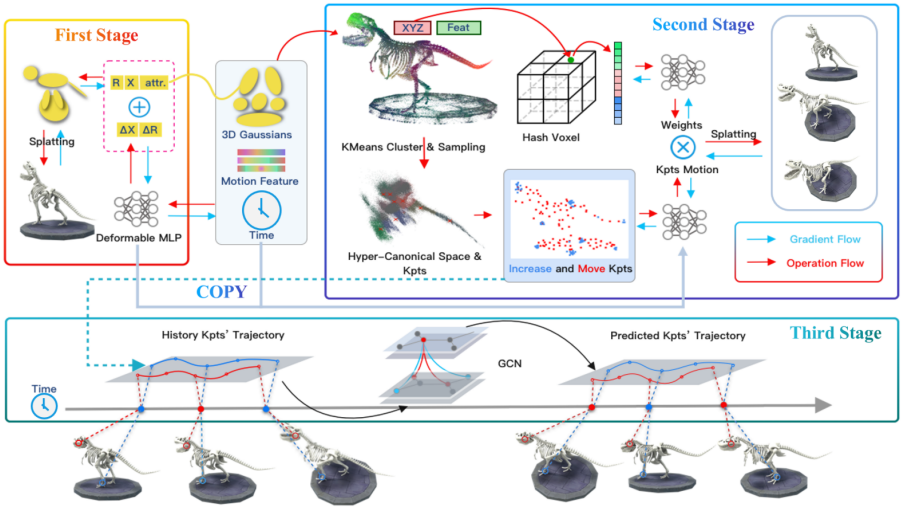
- 3D Gaussian 표현과 dynamic scene modeling, future scenario synthesis를 합쳐 다양한 시점에서 미래 장면을 예측할 수 있음
- 일반적인 움직임, 물건을 자르고 붙이는 등의 비가역적 움직임을 모두 모델링 할 수 있게 각 3D Gaussian에 lifecycle을 도입한 새로운 Canonical Space를 제안함

Index

방법

- 방법은 크게 세 단계로 나뉨
  - 다양한 시점에서 촬영된 사진을 통해 새로운 3D Gaussian의 Canonical Space를 복원함
  - Concentric Motion Distillation을 통해 keypoint를 선별해서 parameter 수를 줄임
  - Graph Convolutional Network를 적용하여 3D keypoint들의 모션을 예측함

방법 (First Stage)



방법 (First Stage)

- 동적인 장면을 렌더링하는 널리 알려진 방법으로는 Canonical Space를 만들고, 그 3D 정보를 변형시켜 time stamp에 맞게 변형하는 것이 있음
- 이러한 방법은 인접한 위치에서 모션이 다를 경우 blur가 일어난다는 문제가 있었음
- 그래서, 각 Gaussian의 차원별 모션 특징을 사용하여 동적인 장면을 인코딩함(center location과 rotation)

방법 (First Stage)

- 또한, 동적인 움직임에는 장난감을 붙이거나, 레몬을 자르거나 하는 비가역적 변형이 일어남
- 이러한 변형은 표면의 일부가 사라지거나, 새로운 표면이 생기게 함
- 각 3D Gaussian의 opacity에 lifecycle이라는 새로운 속성을 부여하여 시간에 따라 보이고 안보이고를 조절할 수 있음

방법 (Second Stage)

- 3D Gaussian의 변형을 예측하는 직접적인 방법은 입력 시간을 각 Gaussian의 인코딩에 넣는 것이나, 이 방법은 3D Gaussian이 기하학적 특징을 잃을 수 있음
- 그래서 모든 3D Gaussian을 변형할 수 있는 keypoint를 기반으로 설계함
  - keypoint 초기화
  - 임의 시점 동작의 복잡도에 따라, keypoint를 적응적으로 늘림
  - 각 keypoint가 3D Gaussian에 미치는 영향을 학습
- 이러한 방법을 통해, 20만개 이상의 3D Gaussian을 몇 백개의 keypoint로 단순화시킬 수 있음

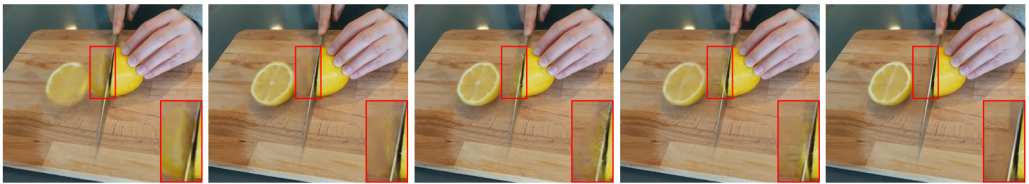
방법 (Third Stage)

- 두 번째 단계에서 장면의 정보가 담겨있는 keypoint들을 얻을 수 있음
- 이 keypoint들의 관계를 고려하여 Graph Convolution Network를 통해 주어진 장면에서 움직임의 패턴을 예측함
- 여러 프레임에서의 keypoint의 관계를 GCN을 통하여 추출하고, MLP를 통해 다음 frame에서 keypoint들의 위치를 예측하여 3D Gaussian을 얻음
- 이전 step의 값을 이용하는 슬라이딩 윈도우 방식으로 연속된 예측이 가능함

Index

결과

- Synthetic Dataset으로는 D-NeRF dataset을 사용하고, Real-World Dataset으로는 Hyper-NeRF real-world dataset을 사용



(a) w/o Hyper Space Init. (b) w/o Adap Increasing. (c) w/o Hyper Space K-NN. (d) w/o Lifecycle. (e) Full Model.

- 각 component 없이 수행하였을 때 차이를 볼 수 있음

결과

- PSNR(Peak Signal-to-noise ratio), SSIM(Structural Similarity Index Measure), LPIPS(Learned Perceptual Image Patch Similarity)를 비교

Method	Trex			Jumpingjacks			Bouncingballs			Hellwarrior		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	20.72	.9284	.0751	19.87	.9115	.0954	25.92	.9677	.0853	29.36	.9097	.1138
4D-GS	20.72	.9401	.0579	20.28	.9176	.0825	29.42	.9753	.0433	31.48	.9266	.0929
Deformable-GS	20.81	.9426	.0461	20.21	.9150	.0800	28.90	.9784	.0271	29.82	.9141	.0834
Ours-MLP	21.51	.9444	.0452	20.68	.9194	.0742	29.58	.9816	.0225	29.99	.9176	.0789
Ours	21.09	.9406	.0461	20.51	.9184	.0760	26.63	.9714	.0361	30.75	.9281	.0729
Methods	Mutant			Standup			Hook			Average		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	24.40	.9282	.0700	21.77	.9169	.0927	21.05	.8817	.1033	22.83	.9229	.0886
4D-GS	24.61	.9269	.0582	22.25	.9140	.0870	23.93	.9042	.0755	23.98	.9305	.0697
Deformable-GS	24.32	.9300	.0469	21.38	.9133	.0837	21.41	.8872	.0824	23.35	.9285	.0623
Ours-MLP	25.05	.9359	.0409	23.04	.9250	.0700	22.6	.8971	.0702	24.14	.9339	.0560
Ours	28.16	.9560	.0256	25.96	.9403	.0481	23.42	.9089	.0573	24.62	.9387	.0514

Motion Prediction Result(D-NeRF dataset)

결과

Method	Trex			Jumpingjacks			Bouncingballs			Hellwarrior		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	31.24	.9771	.0326	34.29	.9799	.0360	35.00	.9835	.0391	39.20	.9763	.0508
4D-GS	33.60	.9863	.0188	35.59	.9844	.0210	37.69	.9919	.0150	38.52	.9754	.0524
Deformable-GS	38.10	.9933	.0098	37.72	.9897	.0126	41.01	.9953	.0093	41.54	.9873	.0234
Ours	37.39	.9926	.0110	37.93	.9906	.0099	41.57	.9954	.0086	41.73	.9874	.0214
Methods	Mutant			Standup			Hook			Average		
	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)	PSNR(↑)	SSIM(↑)	LPIPS(↓)
TiNeuVox-B	35.07	.9768	.0307	38.11	.9854	.0208	33.34	.9711	.0458	35.18	.9786	.0365
4D-GS	38.80	.9857	.0212	40.43	.9890	.0164	33.83	.9728	.0338	36.92	.9836	.0255
Deformable-GS	42.63	.9951	.0052	44.62	.9951	.0063	37.42	.9867	.0144	40.43	.9918	.0116
Ours	42.90	.9954	.0049	45.09	.9954	.0057	37.44	.9868	.0137	40.58	.9919	.0107

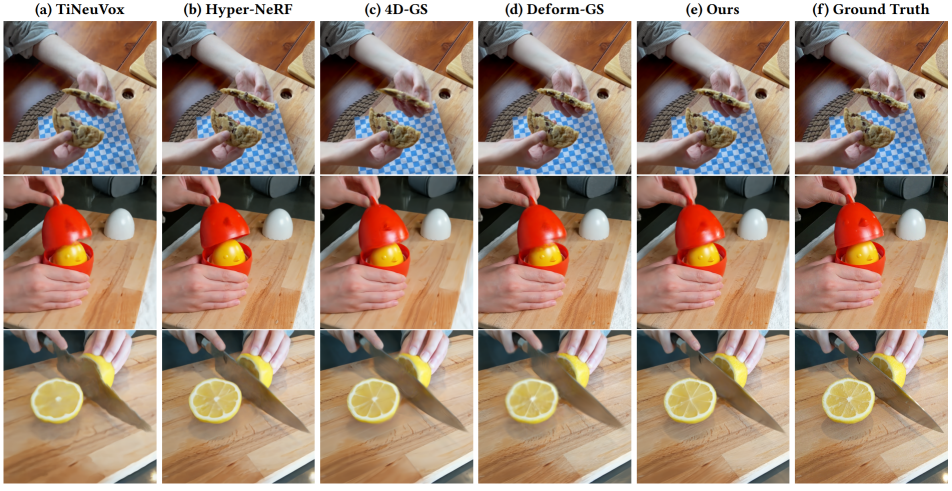
Quantitative Rendering Result(D-NeRF dataset)

결과

Method	CHICKEN (113 images)		CUT LEMON (415 images)		SPLIT COOKIE (134 images)		3D PRINTER (207 images)		AVERAGE	
	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)	PSNR(↑)	MS-SSIM(↑)
TiNeuVox-B	27.7	.951	28.6	.955	28.9	.965	22.8	.839	27.2	.928
HyperNeRF	28.7	.948	31.8	.956	30.9	.967	20.0	.821	28.4	.924
4D-GS	26.9	.911	30.0	.929	32.5	.975	22.0	.808	28.1	.905
Deform-GS	26.1	.902	29.1	.937	32.8	.981	20.3	.756	27.2	.896
Ours	27.1	.920	31.1	.952	34.0	.983	22.2	.814	28.9	.920

Quantitative Result(Hyper-NeRF real-dataset)

결과



Qualitative Result

Index

결론

- lifecycle 속성을 추가한 3D Gaussian canonical Space를 제시
- keypoint를 사용하는 새로운 motion distillation 방법을 사용 motion prediction의 단순화
- long-term prediction를 위한 향후 연구 방향을 제시

Q&A