# DynTCG: Dynamic View Synthesis from Monocular RGBD Streams via Flow-Tracked Compact Gaussians

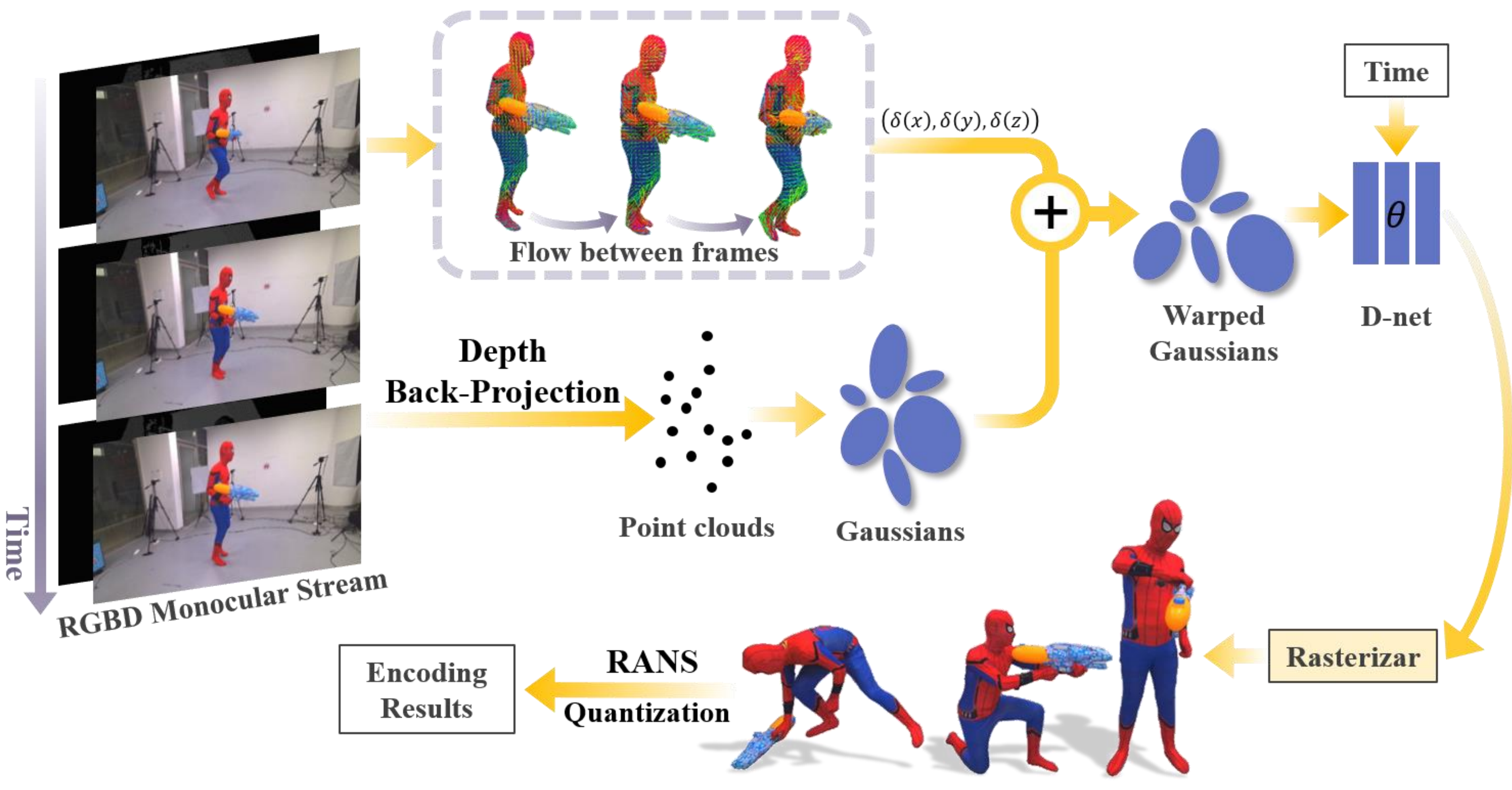

上海科技大学 ShanghaiTech University

Zhehao Shen[1] Penghao Wang[1] Yu Hong[1]
Zhirui Zhang[1] Shouchen Zhou[1]
[1]ShanghaiTech University


## Contributions

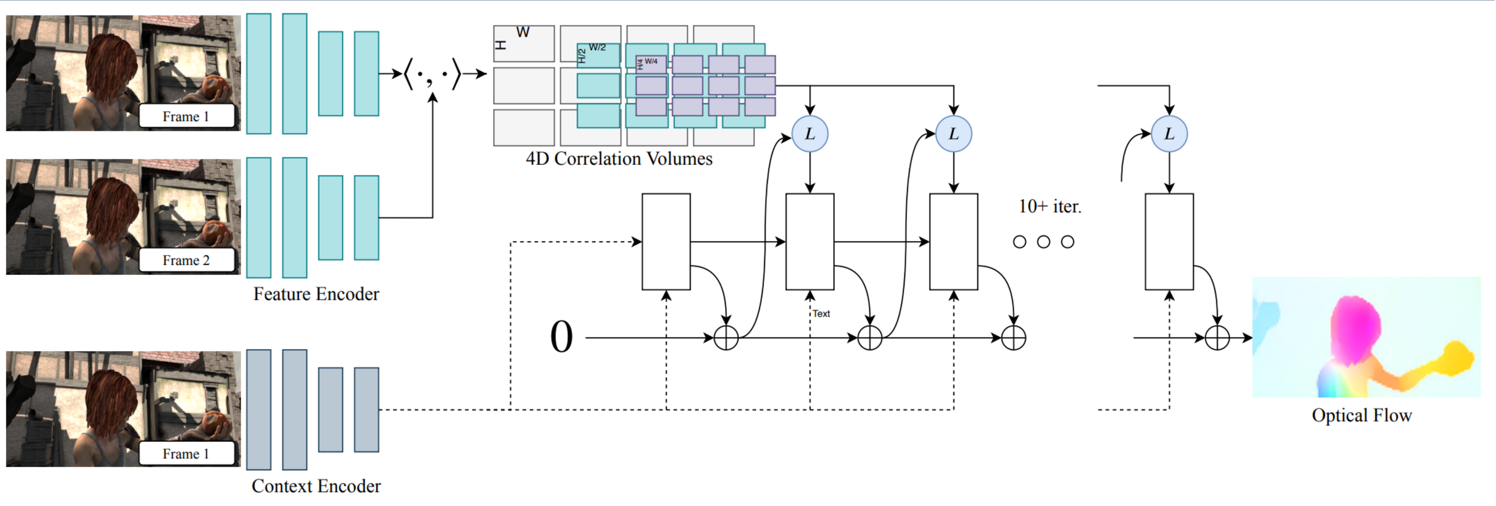
- We propose a deformable Gaussian representation based on monocular RGBD input, aimed at accomplishing the task of dynamic novel view synthesis.
- Utilizing optical flow tracking from 2D images, we provide an optimized initialization for the dynamic Gaussian deform field, significantly accelerating the training speed and enhancing the rendering quality under a monocular setting.
- We showcase a companion compression scheme, supporting high quality rendering with low storage, even under various platforms..

## Pipeline




- **Setting:** RGB-D monocular stream
- **Initialize:** depth back-projection to generate initial point clouds, further generating Gaussians
- **Prior:** RAFT method to complete pixel-to-pixel track, projecting initial Gaussians onto pixels in different frames to obtain displacement of corresponding points
- **D-net:** deformable net to predict $(\delta(x), \delta(y), \delta(z), \delta(r))$
- **Encoding:** use RANS and Quatization to compress

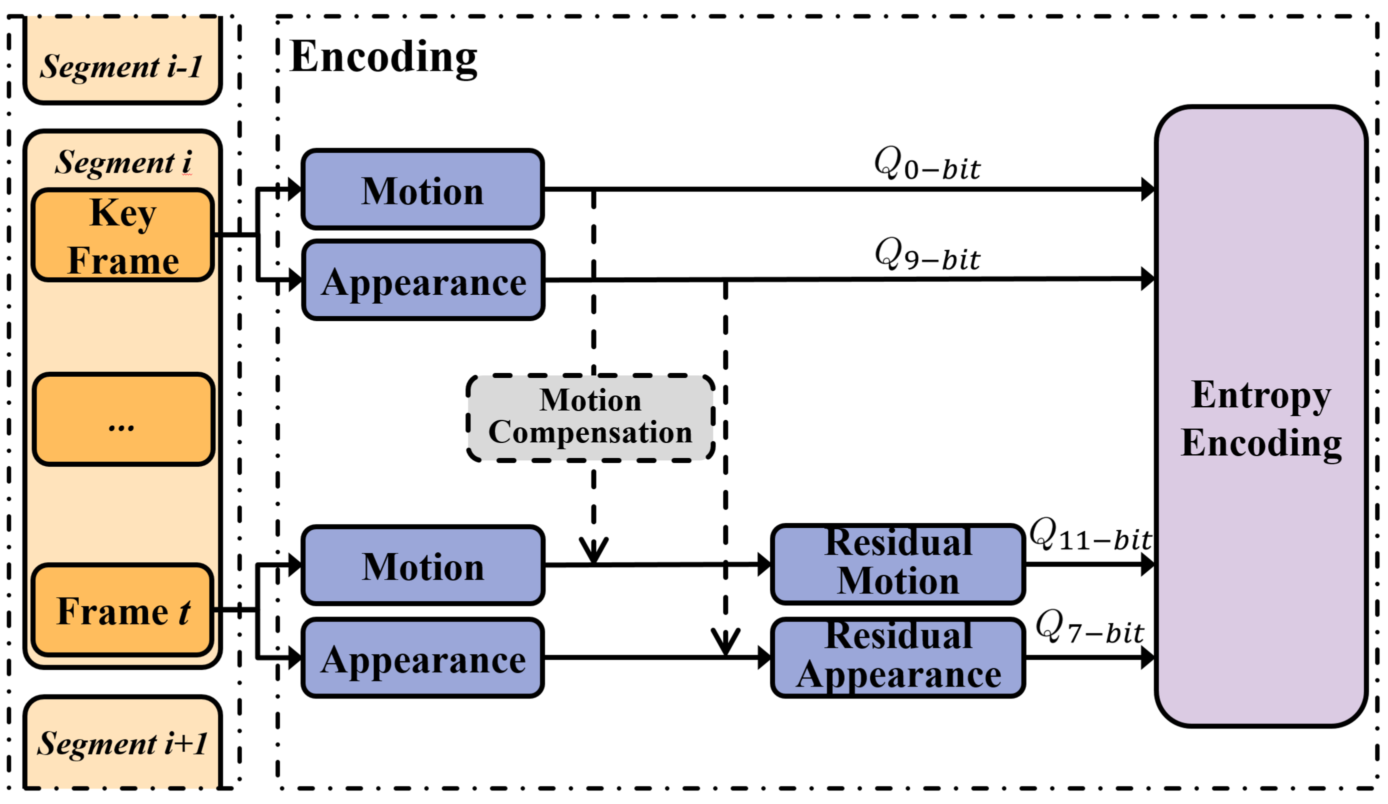
## Component: Optical Flow Tracking




We employ RAFT as our method for optical flow tracking. RAFT comprises a Feature Extractor, Context Extractor, Visual Similarity Calculator, and Updater to optimize the flow tracking process.

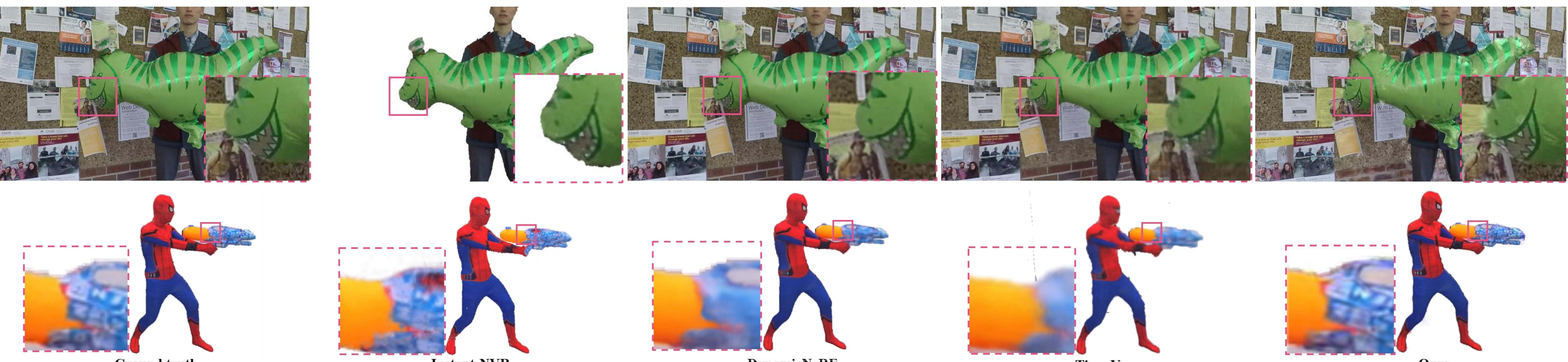
- **Feature Extractor & Context Extractor:** extract features from two images, followed by the extraction of semantic information from one of the images.
- **Visual Similarity Calculator:** constructs a 4D correlation volume by calculating the dot product between feature vectors.
- **Updater:** updates and refines flow tracking by iteratively searching the correlation volume based on current optical flow estimates.

## Component: Residual Compression




- **Residual:** compute residual point cloud data for smaller numerical range
- **Quantization:** numerical quantization to delta point cloud and low bit on SH attribute for low storage
- **RANS Encoding:** entropy encoding to efficiently compress the data

## Comparison



Ground truth Instant-NVR DynamicNeRF TineuVox Ours

## Ablation: Scale

- In the Gaussian parameters (rotation, position, scaling) for shape and position, we found that optimizing the scaling parameter didn't notably improve rendering but increased training time. Consequently, in designing our deform-net, we omitted optimizing the Gaussian scaling.
- Results from training with and without scaling indicate a 9% time difference (with scale: 102 min, without scale: 93 min).

Ground truth Ours_w_scale Residual_w_scale Ours_w_o_scale Residual_w_o_scale

## Ablation: Tracking

| | Training (min) ↓ | PSNR(dB) ↑ |
|---|---|---|
| Without Flow Tracking | 93.23 | 31.65 |
| Ours-full(before compression) | 70.34 | 32.47 |
| Ours-full(after compression) | 70.64 | 32.30 |

- **Without Flow Tracking:** directly use deform net to construct deformable gaussians
- **Ours-full:** preprocess images to get flow between frames. Utilizing depth and optical flow as an effective initialization strategy for the deform net, aiming to expedite the optimization process.
- **Experimental equipment:** We completed the experiment with 4 Nvidia 2080Ti graphics cards.

## Ablation: Compression

| | Per-frame Storage(MB) ↓ | PSNR(dB) ↑ |
|---|---|---|
| Raw Point Cloud | 46.23 | 32.47 |
| High-bit Quantization | 7.34 | 32.23 |
| Low-bit Quantization | 3.23 | 29.56 |
| Ours Residual Encoding | 1.35 | 32.30 |

- **Raw:** directly saved per-frame gaussian point cloud
- **High-bit Quantization:** 11-bit quantization on the point cloud
- **Low-bit Quantization:** 9-bit quantization on the point cloud
- **Ours:** compute the residual between the current frame and the previous frame, and then adopt 9-bit quantization and RANS encodig on the residual data

## Results

- **HyperNeRF** (top-left): pinhole camera, using depth estimation method from Dynamic View Synthesis from Dynamic Monocular Video
- **Dynamic Scene Dataset** (top-right): hand-held monocular camcera, short-term dynamic events (~5 seconds)
- **D-NeRF** (bottom-left and bottom-right): a sparse set of images of a dynamic scene
- **Instant-NVR** (center bottom): RGB-D Kinect camera