360-Degree Video Super Resolution and Quality Enhancement Challenge

TII Organizers: Ahmed Telili, Brahim Farhat and Wassim Hamidouche
University of Klagenfurt Organizer: Hadi Amirpour

I. Overview

Omnidirectional visual content, commonly referred to as 360-degree images and videos, has garnered significant interest in both academia and industry, establishing itself as the primary media modality for VR/XR applications. 360-degree videos offer numerous features and advantages, allowing users to view scenes in all directions, providing an immersive quality of experience with up to 3 degrees of freedom (3DoF). When integrated on embedded devices with remote control, 360-degree videos offer additional degrees of freedom, enabling movement within the space (6DoF). However, 360-degree videos come with specific requirements, such as high-resolution content with up to 16K video resolution to ensure a high-quality representation of the scene. Moreover, limited bandwidth in wireless communication, especially under mobility conditions, imposes strict constraints on the available throughput to prevent packet loss and maintain low end-to-end latency. Adaptive resolution and efficient compression of 360-degree video content can address these challenges by adapting to the available throughput while maintaining high video quality at the decoder. Nevertheless, the downscaling and coding of the original content before transmission introduces visible distortions and loss of information details that cannot be recovered at the decoder side. In this context, machine learning techniques have demonstrated outstanding performance in alleviating coding artifacts and recovering lost details, particularly for 2D video. Compared to 2D video, 360-degree video presents a lower angular resolution issue, requiring augmentation of both the resolution and the quality of the video. This challenge presents an opportunity for the scientific research and industrial community to propose solutions for quality enhancement and super-resolution for 360-degree videos.

II. Dataset

We provide a dataset containing 200 360-degree videos, predominantly sourced from YouTube and ODV360 (https://github.com/360SR/360SR-Challenge#track-2360-omnidirectional-video-super-resolution-x4) characterized by high quality and resolution (4K and 2K) in ERP format. All videos are licensed under Creative Commons Attribution (reuse allowed), and our dataset is exclusively designed for academic and research purposes. The video dataset encompasses various content characteristics, including outdoor and indoor scenes, as well as high motion sport contents. Each video consists of 100 frames. The dataset is partitioned into 170 videos for training, 15 for validation, and 15 for testing. Note that additional external content can be incorporated for training.

	# of pristine videos	#frames/video	#Resolution 4k	#Resolution 2k	Bitrates	#Total videos
Training data	160	100	80	80	4 points @ {2, 4, 6 and 8} Mbps	640
Validation data	20	100	20	20		80
Test data	20	100	20	20		80

a. Dataset characterization

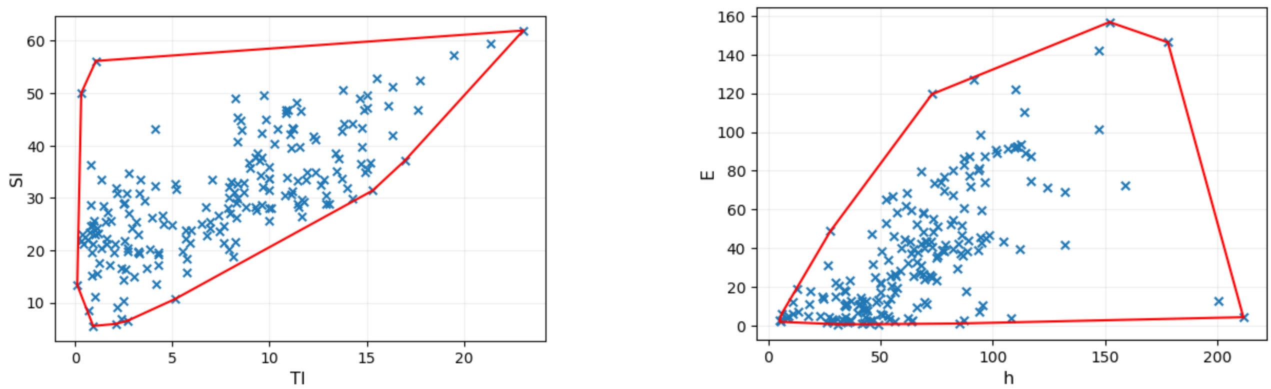

Figure2 : Source content distribution in paired feature space with corresponding convex hulls. Left column: TI×SI and right column: hxE.

SI: Spatial information (VQEG implementation)
TI: Temporal information (VQEG implementation)
E: Spatial complexity (VCA)
H : Temporal comlexity (VCA)

III. Evaluation portal mechanism

As it can be seen in figure 1, the processed video's quality is evaluated frame-by-frame against the ground truth using the widely used quality metric Weighted-to-Spherically-uniform Peak Signal to Noise Ratio (WS-PSNR). Additionally, information on the neural network's complexity, including model size and the number of operations, is included in the final score for the model.

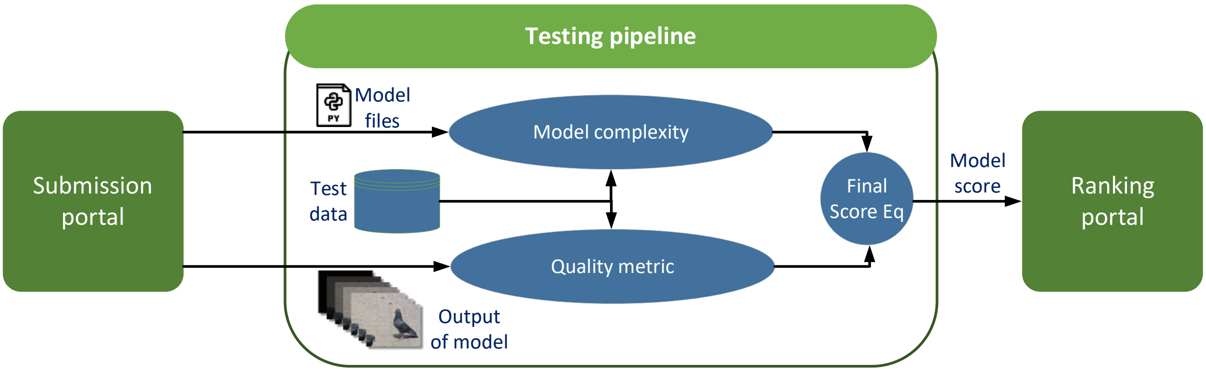

Figure 1: ranking pipeline

a. Model complexity equation

This step is mainly used to evaluate the model complexity. This is by considering two main features: the size of the model and the inference time.
All submitted models will undergo the same testing over our machine with the following characteristics:

- CPU:
- GPU:
- RAM:

First, the inference time by pixel for an input sequence is computed following the below equation:

$$M_C = \frac{I_{time}}{W * H * N_{frames}}$$

M_C is the model complexity per pixel, I_{time} is the inference for an input video sequence. $W, H \text{ and } N_{frames}$ are the width, height and number of frames of the input sequence.

The final score is computed by taking the average of M_C over the number of tested videos N_{videos}, as shown in the next equation.

$$C_{score} = \frac{1}{N_{videos}} * \sum_{n=0}^{N_{videos}-1} M_C$$

b. Quality metric equation

To evaluate the performance of the submitted model we use quality metric for 360 videos named Weighted-to-Spherically-uniform Peak Signal to Noise Ratio (WS-PSNR). This is made publicly available at link.

The WS-PSNR is computed for every bit rate target and for every frame of the video sequence, as shown in the next equation.

$$VQ_{score} = \frac{1}{4 * N_{frames}} * \sum_{n=0}^{N_{frames}-1} \sum_{br=0}^{3} (S_{PSNR}(br))$$

The VQ_{score} is the Video Quality score which is the average quality score of the submitted model's WS-PSNR, S_{PSNR}, per video for all target bitrates and all frames of the sequence.

The final score Q_{score} for the model is the average VQ_{score} over all the test data video.

$$Q_{score} = \frac{1}{N_{videos}} * \sum_{n=0}^{N_{videos}-1} VQ_{score}$$

c. Final score equation

The final score is computed following the next equation:

$$M_{score} = w_1 C_{score} + w_2 Q_{score}$$

M_{score} is the model final score which a weighted combination between the Complexity score C_{score} and the Quality score Q_{score}. $w_1 and\ w_2$ are weights.

Weights will be defined after testing the baseline model.

IV. Framework

The framework for assessing model performance will be publicly available. To ensure compatibility with our framework, every model must adhere to specific criteria and requirements.

a. Format

b. Requirements

V. CodaLab competition organization

a. How it works:

During the development and feed-back phases, participants have access to training data to develop and refine their algorithms. During the final competition phase, participants are provided with final test data to generate results, which they can then submit to the competition. Results are calculated at the end of each phase, at which point participants can see the competition results on the leaderboard.

b. Bandle file structure:

Competitions creation requires a set of files collectively known as a "bundle". Although technically CodaLab considers any zipped archive to be a bundle, competition bundles generally contain a specific assortment of files:

- **Competition.yaml:** Configuration file defining all the features of the competition and linking to resources needed to organize the competition (html files, data, programs).
- **HTML pages:** Descriptive text and instructions to participants.
- **program files:** Files that make up the ingestion program, the scoring program, and the starting kit. With the default computer worker and docker on the public instance, we support Python (.py) files and Windows-based binary executables. Via the use of your own docker images and running your own computer workers, you have the flexibility of using any programming language and OS.
- **data files:** Contain training data and reference data for the competition.

Basic tutorial: https://github.com/codalab/codalab-competitions/wiki/Create-your-first-competition

VI. Planning

- *Week 25 to 29 December 2023:*
  - Data set collection
  - Challenge submission to ICIP
- *Week 2 to 5 January 2024:*
  - Data set collection
  - Testing baseline model
- *Week 7 to 12 January 2024:*
  - Organizing dataset over drive
  - Starting framework preparation
  - Creating competition over Codalab or Kaggle (TBD)
- *Weeks 15 to 25 January 2024:*
  - Framework preparation
  - Preparing github for Framework
- *Weeks 27 January to 05 February 2024:*
  - Testing the framework
  - Testing the submission and ranking portal
  - Evaluating the metrics
- *Weeks 08 to 12 February 2024:*
  - Final check over all the pipeline
  - Starting the releasing process