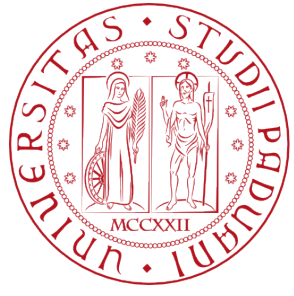
Local Feature Compression Using Autoencoders

3D Vision and Extended Reality

Pasian Francisco and Zhang Qiqi

July, 2024



Università degli Studi di Padova

Motivation

- The ever-increasing amount of data, especially high-dimensional data like images, videos, and sensor data, necessitates efficient storage solutions. Compressing data for transmission reduces bandwidth requirements, which is crucial for applications involving real-time data transfer, such as streaming services and IoT devices.
- We study a compression strategy for local descriptors using SURF (Speeded Up Robust Feature), to create a 3D reconstruction of an object using SfM (Structure from Motion) from pictures taken from a camera.
- An approach that achieve compression is autoencoders, we particularly use the Variational Autoencoders

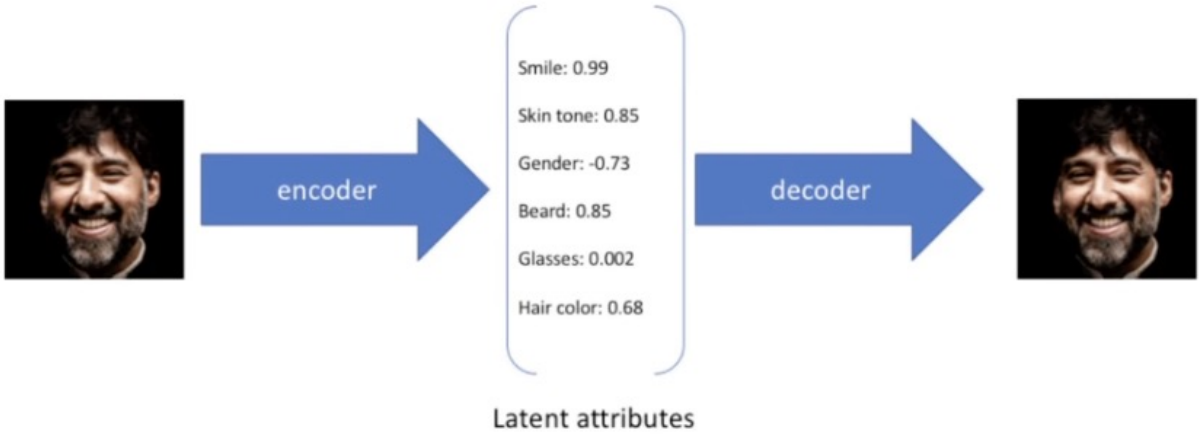
SURF: Speeded Up Robust Features

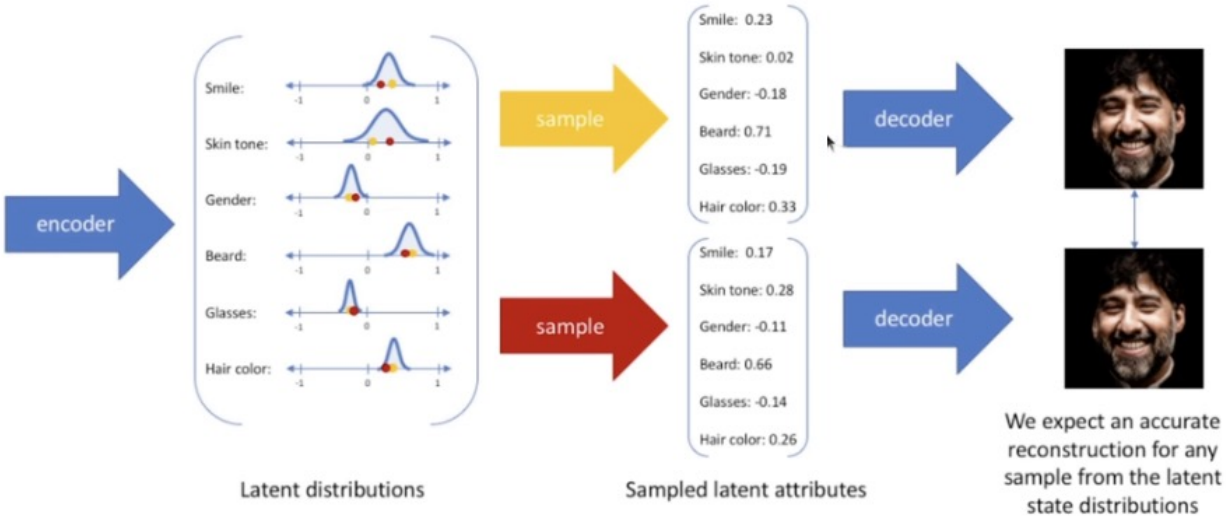
SURF [1] is a fast and performant interest point detection-description scheme.
Some key features:

- Speed: is faster than previous methods like SIFT due to integral images.
- Robust: is invariant to scale, rotation, and some degree of illumination changes and affine transformations.
- Applications: is used in object recognition, image stitching, and 3D reconstruction.

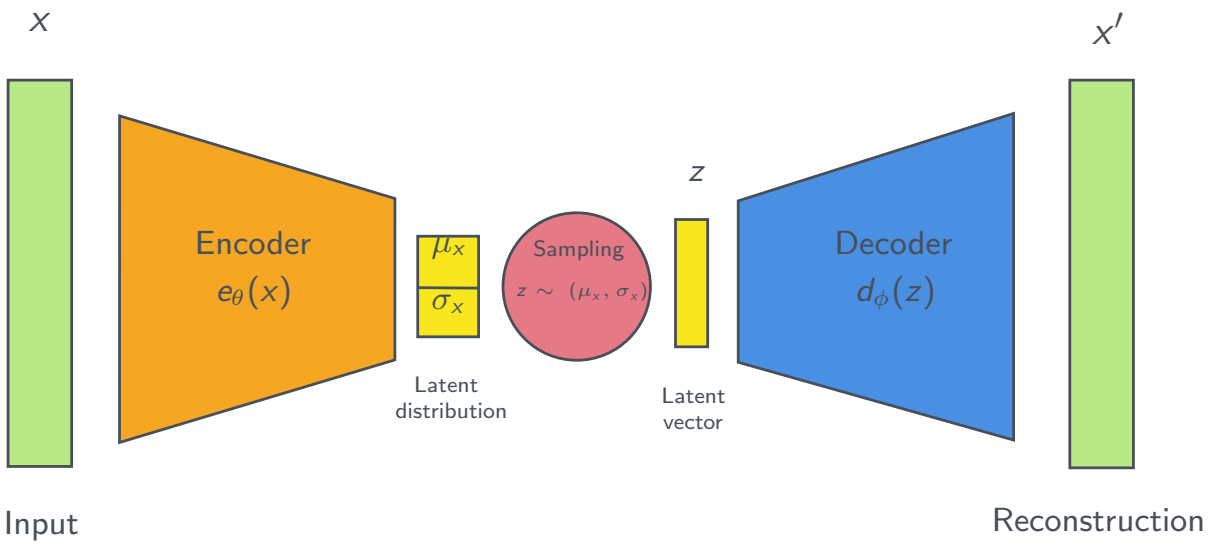
[1] https://link.springer.com/chapter/10.

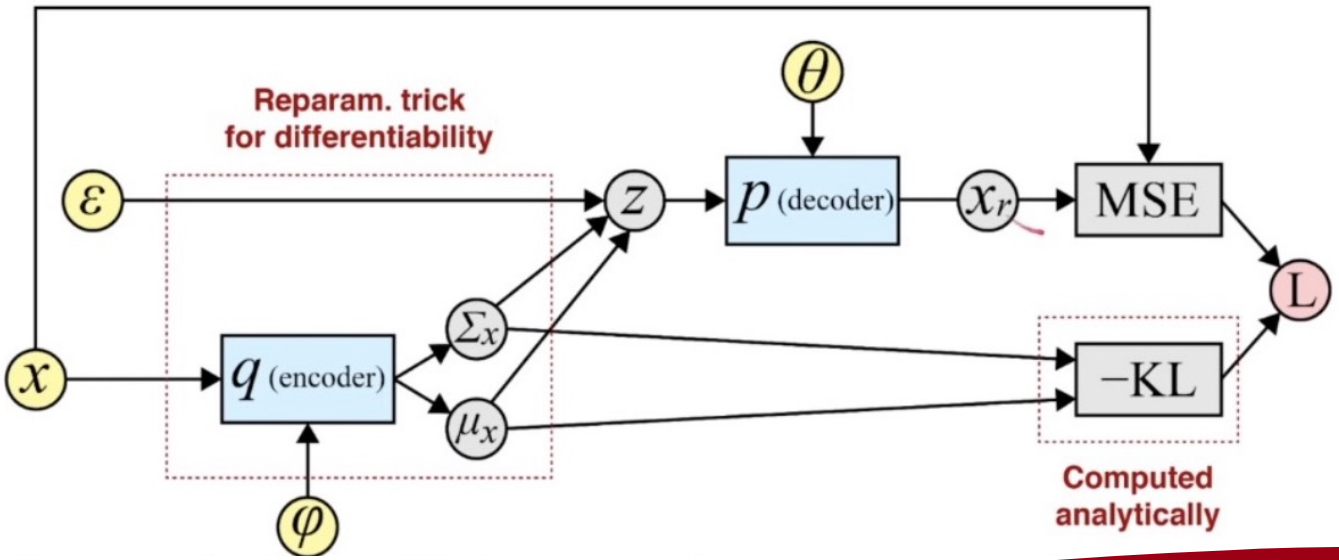
Variational Autoencoder (VAE)





Variational Autoencoder (VAE)



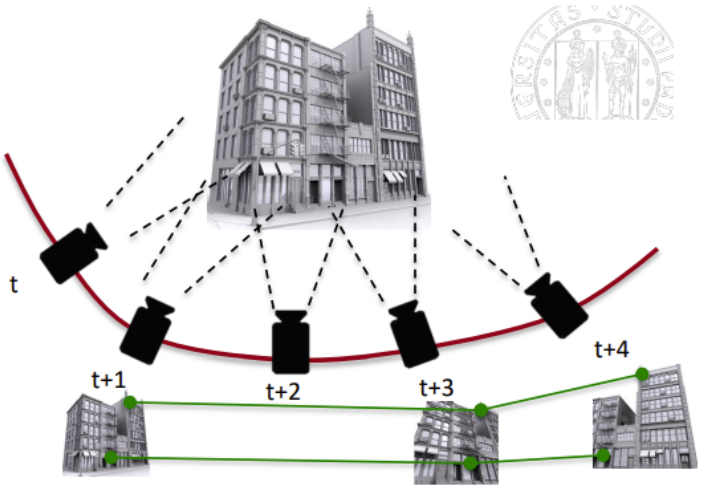


Structure from Motion (SfM)

Assume that a camera is moving around a scene, intrinsics (camera params) are known, and conjugate points are available (e.g. via matching). Compute 3D structure of the scene with orientation and location of the camera at the different instants (e.g. $t, t+1, ...t+n$).

SfM pipeline:

- Images are collected with a camera
- Corresponding points in the different views are found with SURF
- Epipolar geometry between couples of cameras is computed and integrated progressively (order matters)
- Matrix can be decomposed into R and T; intrinsics can be estimated
- Refinement using some bundle–adjustment strategy



Data process

- We implemented a Variational Autoencoder (VAE) model and the generated descriptor were trained with the images of dataset of Portello and Tiso ([2]). We used a dataset of Fountain and Castle ([3]) for testing.

Dataset	Image Size
Training (Portello)	41
Validation (Tiso)	10
Test 1 (Fountain)	11
Test 2 (Castle)	30

[2] https://github.com/openMVG/SfM_quality_evaluation/tree/ma

[3] https://drive.goo

Data process

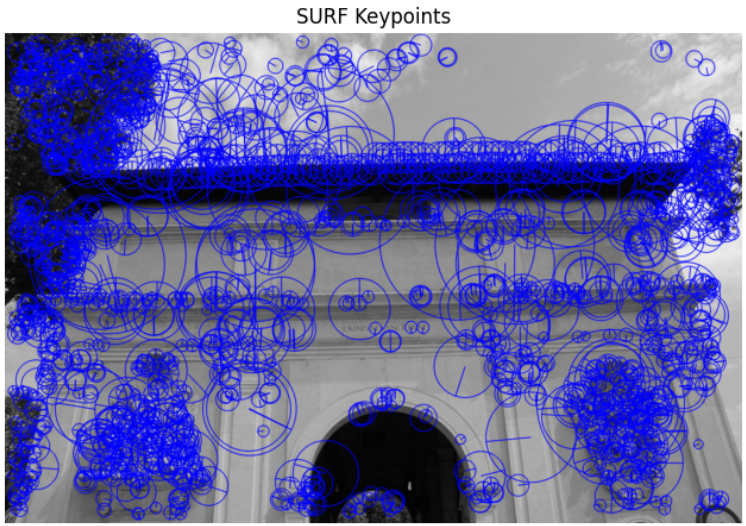
- Data augmentation step has been done to increase the training dataset.

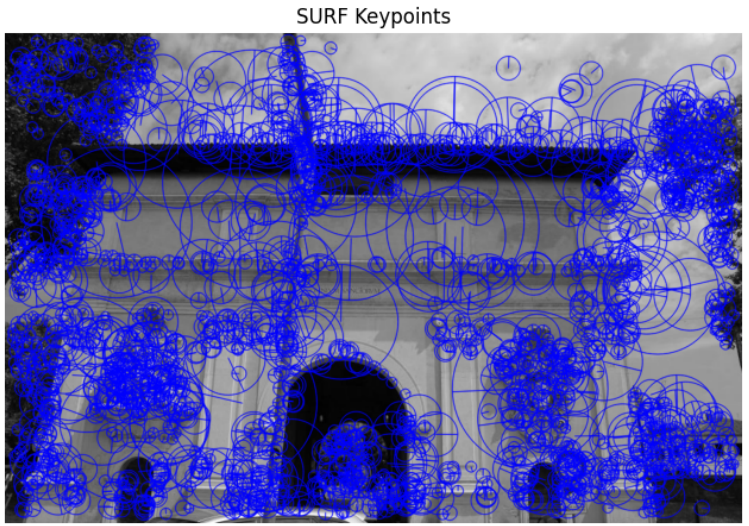
Dataset	Image Size
Training (Portello)	82
Validation (Tiso)	10
Test 1 (Fountain)	11
Test 2 (Castle)	30

```python
# Define torchvision transforms for augmentation
transform = transforms.Compose([
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomVerticalFlip(p=0.5),
    transforms.RandomRotation(degrees=30),
    transforms.ColorJitter(brightness=0.2,
    contrast=0.2, saturation=0.2, hue=0.1),
])
```

Data process

- Extract SURFT keypoints and descriptors from the images.



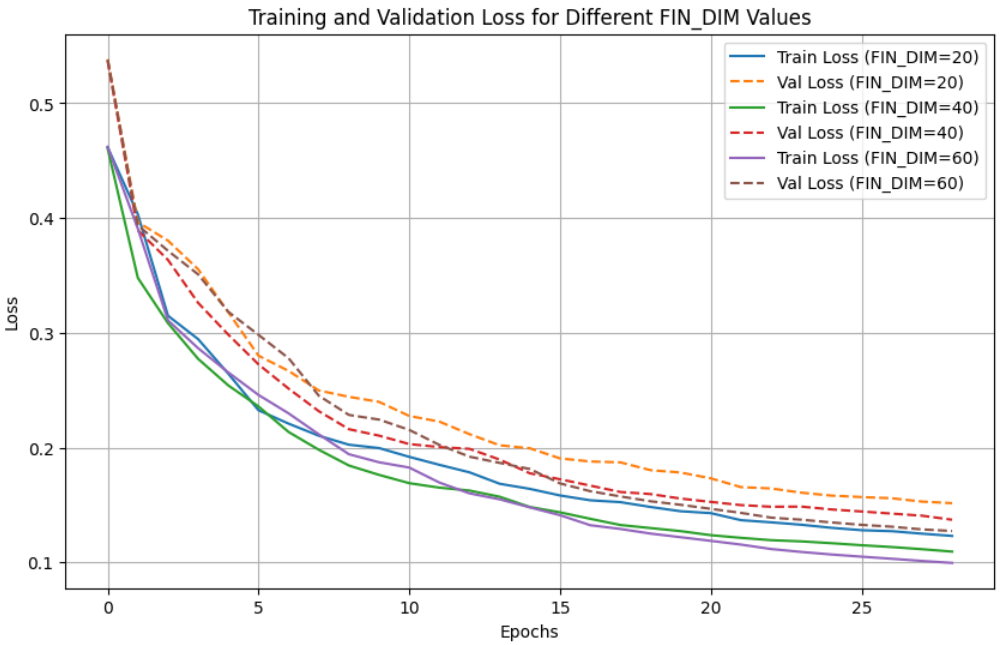


Model setup


```
VAE(
  (encoder): Sequential(
    (0): Linear(in_features=128, out_features=100, bias=True)
    (1): LeakyReLU(negative_slope=0.2)
    (2): Linear(in_features=100, out_features=80, bias=True)
    (3): LeakyReLU(negative_slope=0.2)
  )
  (mean_layer): Linear(in_features=80, out_features=20, bias=True)
  (logvar_layer): Linear(in_features=80, out_features=20, bias=True)
  (decoder): Sequential(
    (0): Linear(in_features=20, out_features=80, bias=True)
    (1): LeakyReLU(negative_slope=0.2)
    (2): Linear(in_features=80, out_features=100, bias=True)
    (3): LeakyReLU(negative_slope=0.2)
    (4): Linear(in_features=100, out_features=128, bias=True)
    (5): Sigmoid()
  )
)
```

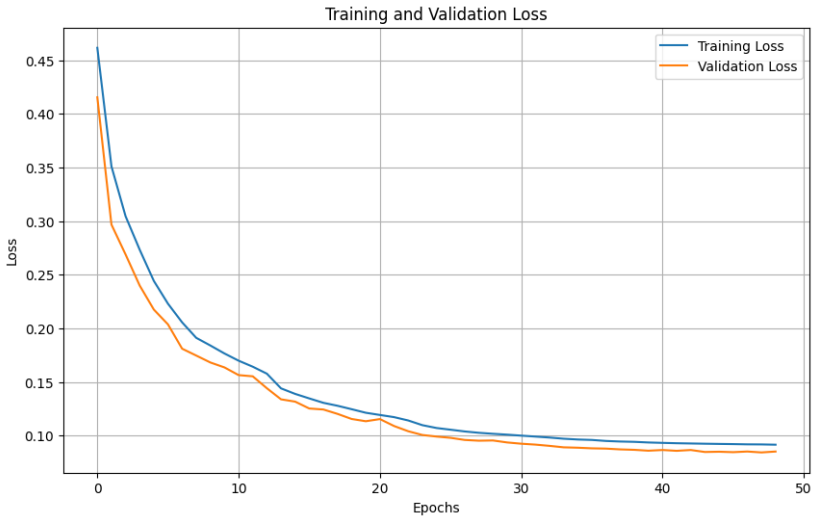
Parameters	Value
INP DIM	128
HID DIM	100
LAT DIM	80
FIN DIM	20
RELU T	0.2
Device	mps
Optimizer	Adam
Epochs	30

Hyper-parameters Fine Tuning



Parameters	Value
INP DIM	128
HID DIM	100
LAT DIM	80
FIN DIM	20, 40, 60
RELU T	0.2
Device	mps
Optimizer	Adam
Epochs	30

Training Results

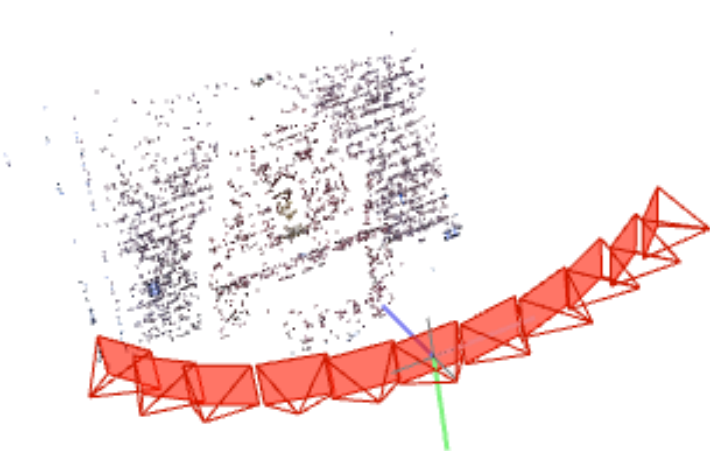


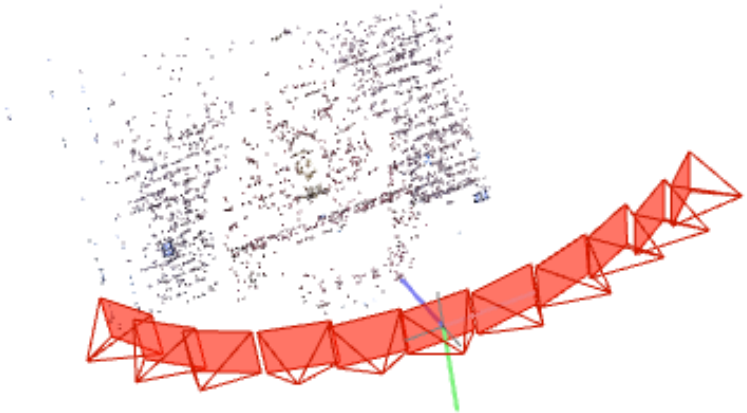
	Value
FIN DIM	60
Epochs	50
Training Error	0.0916
Validation Error	0.0853
Test 1 Error	0.1258
Test 2 Error	0.1210

Testing: Fountain reconstruction

Using COLMAP we obtain a 3D reconstruction from images which first recovers a sparse representation of the scene and the camera poses of the input images with SfM.
On the left side, a reconstruction with original descriptors via SURF which generated sparse point cloud consists of 4341 point, and on the right side, the reconstructed (descriptors from the VAE model) with 3488 points.

Descriptors	Points	Observations	Mean track length	Mean observations per image	Mean reprojection error
Original	4341	17603	4.05506	1600.27	0.421963
Reconstructed VAE	3488	13469	3.86153	1224.45	0.415662

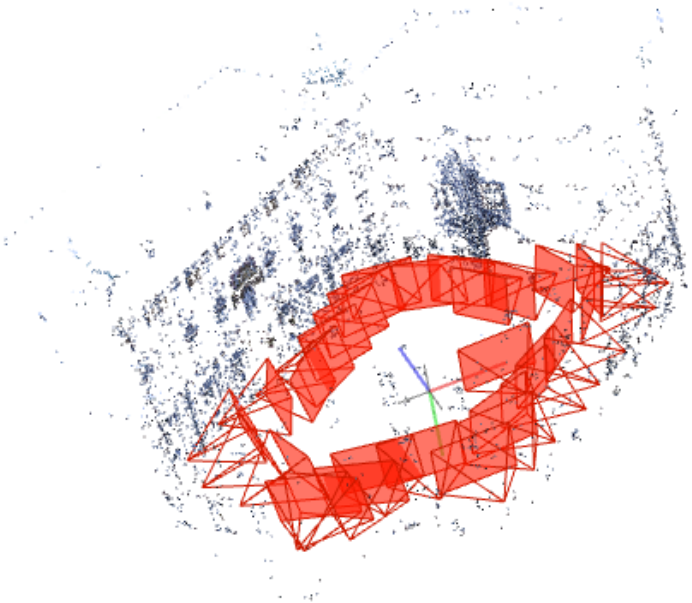


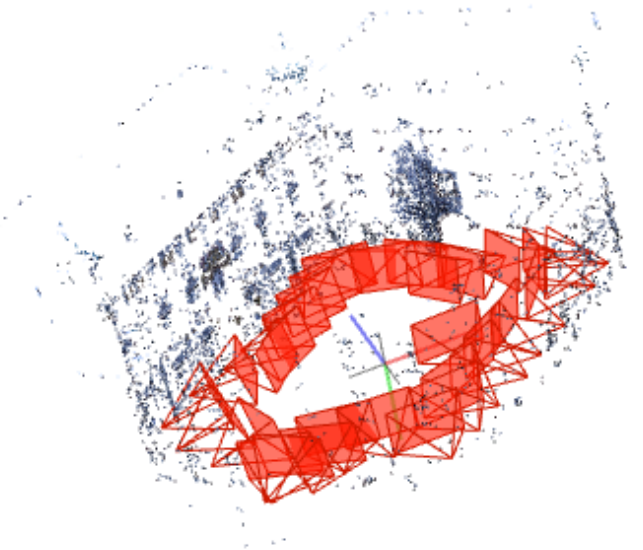


Testing: Castle reconstruction

Here we can see on the left side the original reconstruction with sparse point cloud consists of 12190 points and on the right, the reconstructed object (descriptors from the VAE model) with 11268 points.

Descriptors	Points	Observations	Mean track length	Mean observations per image	Mean reprojection error
Original	12190	51060	4.18868	1702	0.556477
Reconstructed VAE	11268	43621	3.87123	1454.03	0.562324





Final remarks

- We have analyzed the reconstruction of images in 3D using Structure-from-Motion, and feed it with the feature descriptors of the image via SURF and matching values.
- By compressing these descriptors, we can reduce data storage requirements and improve processing speeds without sacrificing accuracy.
- The compression process has been achieved with Variational Autoencoder model reducing the number of point needed to describe an image with the best performance at a dimension of 60 for the output latent vector.

Future works: explore different compression techniques on local descriptors, increase image quality and evaluate performance with dense reconstruction, to build a 3D mesh model.