POLITECNICO DI TORINO

Master's Degree in Mathematical Engineering

Master's Degree Thesis

Segmenting Dynamic Objects in 3D from Egocentric Videos

Supervisors

Prof. Tatiana TOMMASI

Dott. Chiara PLIZZARI

Candidate

Francesco BORGNA

March 2024

Abstract

With the increasing availability of egocentric wearable devices, there has been a surge in first-person videos, leading to numerous studies aiming to leverage this data. Among these efforts, 3D scene reconstruction stands out as a key area of interest. This process allows for the recreation of the scene where the video was captured, providing invaluable support for the growing field of augmented reality applications. Some egocentric datasets include static 3D scans of recording locations, usually requiring costly hardware or dedicated scans. An alternative approach involves reconstructing the scene directly from video frames using Structure from Motion (SfM) techniques. This method not only captures the motion of the actor and the objects they interact with, including transformations (e.g., slicing a carrot) but also enables the use of any egocentric footage for scene reconstruction, even without physical access to the environment in real life. However, the task of decomposing dynamic scenes into objects has received limited attention. For example, SfM finds it challenging to distinguish between moving and static parts, resulting in cluttered point cloud reconstructions where the same object may appear superimposed or in multiple places within the scene.

In this thesis, we combine SfM with egocentric methods to segment moving objects in 3D. This is achieved by creating a scene with COLMAP, a SfM algorithm, and then modifying a recent algorithm called NeuralDiff, originally designed for producing 2D segmentations of static objects, foreground, and actors, to extract 3D geometry. Additionally, we explored ways to reduce the overall computational demands, such as by simplifying the NeuralDiff architecture to better meet our goals by merging the foreground and actor streams, and by developing an intelligent video frame sampling technique that captures the essence of the scene using fewer frames.

Acknowledgements

ACKNOWLEDGMENTS

"HI"
Goofy, Google by Google

Table of Contents

List of Tables

List of Figures

Acronyms

SfM
: Structure from Motion

pcd
: Pointcloud

MLP
: Multi Layer Perceptron

Introduzione

Qua ci andrà la introduce

Part I

Our Contribution

Chapter 1

Methodology

Now that the reader has grasped the basics and some nuances of photogrammetry and neural rendering, let us see how we exploited the presented topics and used them to reach our goal, that is segmenting Dynamic Objects in 3D from egocentric videos and intelligently filter samples to remove redundant informations.

1.1 Goals

Main Goal The main goal of this thesis is to Segment Dynamic Objects in 3D from egocentric videos. Up to now SfM algorithms finds it challenging to reconstruct dynamic scenes, resulting in messy pointclouds, where the same object can appears multiple times within the scene. An explicative example is reported in Figure 1.1. A scene where a pizza is prepared from dough is reconstructed. It is clearly visible from the frames that on the induction cooker is slowly proceeding the making of the pizza but in the reconstruction it is reported in its integrity. This is due to a lack of temporal reasoning of structure from motion procedures. Also, above the pizzas there seem to be multiple pans intersected one with each other. Obviosuly this is not corresponding to the reality, but moving of few centimeters at some time intervals, the process register it as a new object each time it is in a new position. Other examples are visible like the chopping board and some object on the inductor stove. In this example our goal would be to reconstruct the kitchen cleaned of all the object that moved during the video recording. This would be of immense potential since any environment could be reconstructed from just a video of it and the presence of dynamic objects/person would not interfere with it.

Second Goal The second goal of this thesis is to reduce the computational times of the overall pipeline. Being based on the heavy neural network of NeuralDiff, our pipeline tries to speed it up by finding a filtering method that could reduce

Figure 1.1: Reconstruction of a pizza preparation video. The top and bottom frames give us a glance of the action performed during the video while the central pointcloud highlighht the problems of SfM in dynamic environments like the superimposition of the same object on itself or the reconstruction of objects that are not always present in the scene, e.g. the two pizzas.

the number of frames while keeping the same important information of *larger* samplings. This will be presented in Section 1.3. Also, we took a simplified version of NeuralDiff, in which the actor,the person who is wearing the camera, and the foreground,the dynamic objects, are fused together, since for our scopes the distinction was not needed and thus we could remove one of the three neural radiance fields by accelerating the computations.

1.2 Pipelines

In this section we will present the actual methods that we considered and implemented for actually achieve our goals. As we have seen from Related Works, 3D dynamic object segmentation is still an evolving field. We took inspiration from various works to actually come up with some different ideas for actually segmenting 3D dynamic objects. All the methods revolve around a COLMAP reconstruction and a NeuralDiff renderer. The basic block would in fact be NeuralDiff but a reconstruction of the scene with camera intrinsic and extrinsic parameters is required

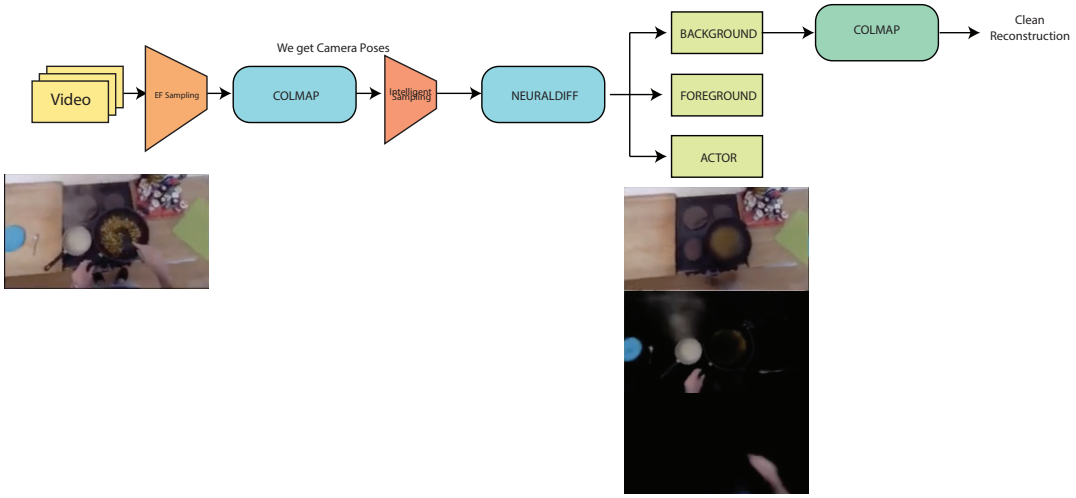

Figure 1.2: Basic Pipeline. In the Basic Pipeline a video(represented by the image of a person cooking) is subsampled through EF-Sampling, reconstructed via COLMAP, re-sampled on the reconstructed frames and then fed to NeuralDiff. At this stage the frames are decomposed in actor,foreground and background(as can be seen in the frames reported below NeuralDiff). The Clean reconstruction is obtained by running another COLMAP step on the extracted background frames.

to make it work.

Colmap Pipeline The first and most trivial idea is to reconstruct the scene with the SfM algorithm,COLMAP, and then separate the static and dynamics objects using NeuralDiff [1]. Once the cleaning has been done we could reconstruct from the cleaned frames the static scene. In Figure 1.2 is reported the pipeline of this first approach.

Monocular Pipeline Anyway this way the previous method is not optimal because we have two COLMAP steps and one of NeuralDiff. We wanted to do better. So we tried with a second pipeline that we will call *Monocular-Pipeline*. This pipeline remove the last COLMAP step by using a pre-trained neural monocular depth estimator. This last block allows to project any frame from 2D to 3D knowing the camera extrinsic and intrinsic parameters. So we could train NeuralDiff(reconstruction of the scene included) and the project the processed frames into the space. Then to segment dynamic objects we can project all dynamic frames,actor or foreground or both, in 3D and take all the points of the COLMAP pointcloud that are at a distance less than a predefined value from the projected

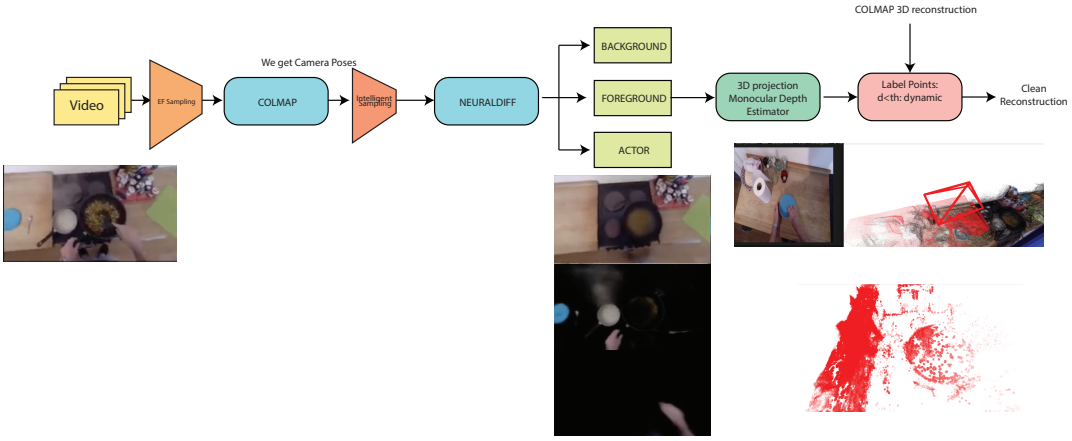

Figure 1.3: Monocular Pipeline. The Monocular pipeline share the first part with the Basic Pipeline. A video is subsampled,reconstructed via COLMAP, subsampled again and fed to Neuraldiff. The difference is that here the dynamic layers are projected in 3D and points closer than a distance th are segmented as dynamic.

ones and classify them as dynamic. The same could be done with static points. This second pipeline is reported in Figure 1.3.

NeuralDiff Pipeline The last pipeline instead take advantage of the intrinsic knowledge of the neural renderer, removing the necessity of the last step, being that SfM or a Depth extractor. For the definition of a nural renderer we know that it is a neural network that take as input a point spatial coordinate with the direction from which it is observed and returns back the color and density of that point. This implies that when we are training we are already grasping the 3D structure of the scene, and any additional step would be unnecessary. The overall pipeline can thus be simplified as in Figure 1.4, and it will be referred as *NeuralDiff-Pipeline*. A point is segmented as moving if its corresponding point in the static scene has a density<threshold.

NeuralCleaner The last modification that we made was to remove the distinction of the actor and the foreground, combining them into a single stream. Since it was out of our scope to distinguish between these two layers, we hope to reduce the computational times by removing one of the three neural streams of NeuralDiff. This pipeline has been named *Neural Cleaner*.

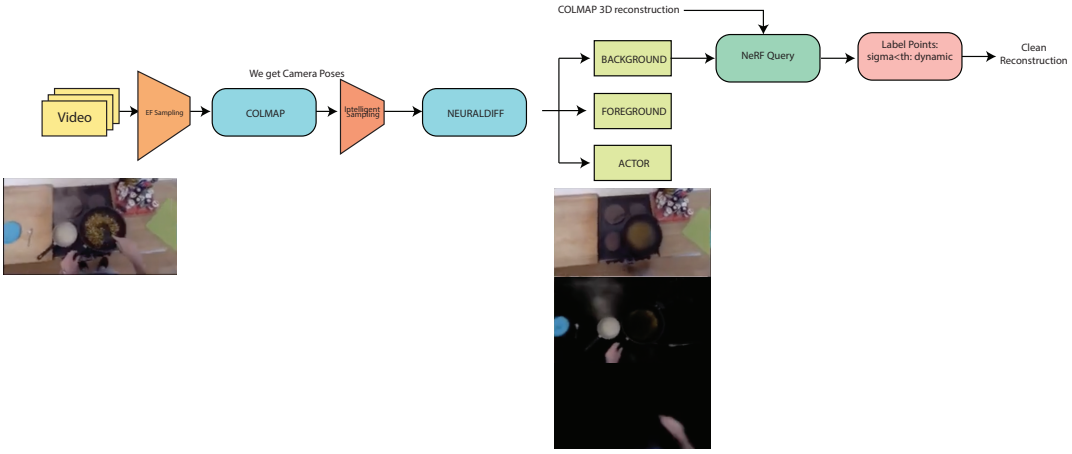

Figure 1.4: NeuralDiff Pipeline. In this pipeline we obtain the three motion layers as in the other methods but actually the segmentation is performed querying the static neural renderer with the positions of the points belonging to the COLMAP reconstruction. Each point is segmented as dynamic if its density is less than a predefined value.

1.3 Filtering

The filtering method consists in seeking temporal windows where frames are overlapped and keeping just a frame per window. The overlap is computed by estimating homographies [1] on matched SIFT **??** features. In Figure 1.6 some examples of overlaps are visualized.

EF Sampling. This technique was proposed in EPIC-Fields [2] to obtain accurate 3D reconstructions from egocentric videos, which present the challenging problems of: dynamic objects,long duration video($\tilde{9}$min on avg) and the skewed distributiton of viewpoints, namely the fact that in videos there are phases of slow motion around hot-spots (e.g. around the gas stove) alternating to high motion in transition actions(e.g. taking something from the pantry). The main idea was to remove redundant frames while maintaining enough overlap and temporal coverage to allow an accurate reconstruction. We will refer to this method as *EF-Sampling*.

Intelligent Sampling. TWe took inspiration from this technique for our sampling method that we called *Intelligent Sampling*. This sampling was used to keep only

[1]A homography is a transformation that maps points from one image to corresponding points in another image.

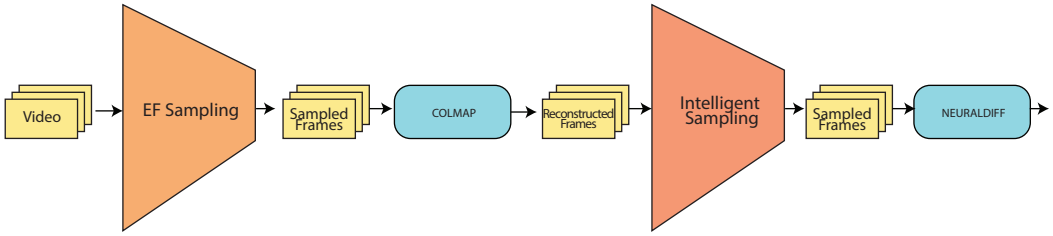

Figure 1.5: Sampling Steps in our pipelines. The initial video is subsampled by EF-Sampling and the resulting frames are fed to COLMAP. Onve COLMAP reconstructed the scene these frames are again subsampled via Intelligent Sampling and fed to NeuralDiff.

important frame, trying to remove any redundant information for the neural rendering step. The idea to maintain important frame is different from the SfM step. Here we thought that a better set of images would have less overlap between itself, such that all the areas of the scene are covered, or at least the frames are equispaced in the scene if they are too few. In this way there should not be any blind spot and the environment is captured in its entirety. And this is actually the contrary of the idea of EF-Sampling. Its position in our pipeline is reported in Figure 1.5. The COLMAP reconstructed frames are *Intelligently* sampled and fed to NeuralDiff.

The actual implementation is strictly based on EF-Sampling and since its goal is the contrary of the latter, they share part of the method. Intelligent sampling in fact given a set of N frames perform a EF-sampling with a overlap threshold such that it gives $N - N_{desired}$ frames. These frames are then discarded and the remaining $N_{desired}$ frames are kept.

AU Sampling. We also tried a less rigid approach with the *AU-Sampling*. This other method is an hybrid of Intelligent-Sampling and a Uniform sampling(which is the baseline we tried to improve). In AU-Sampling we perform a Intelligent-Sampling relaxing the $N_{desired}$frames. The $N_{desired}$ frames are then uniformly extracted from the 'relaxed' samples. In this way we hoped to keep some overlapping frames by relaxing the threshold and then obtain equitemporal spaced samples from all the duration of the video.

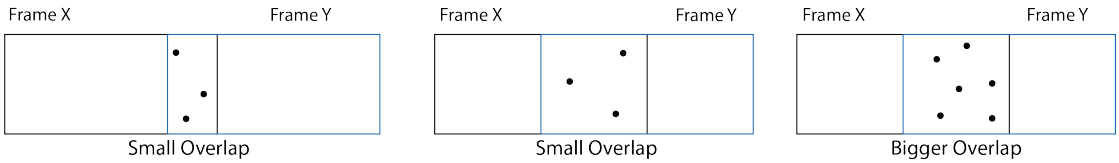

Figure 1.6: Example of overlapping frames(X and Y). The left and central examples present the same level of overlap even though the image frames are closer together in the central example, because the number of features shared are the same. The right examples instead present a higher level of overlapping due to the bigger number of features.

Chapter 2

Experiments

2.1 Data selection

The data selection was dictated by our problem. Indeed evaluating 3D scenes reconstructions is not an easy task due to the lack of 3D ground-truths. These are usually very expensive due to costly hardware scanners but sometimes are also not really available, as in our scenario, where we would like a static-dynamic segmentation. In our case in fact obtaining the static part would mean to actually clean the scene from all the possible moving objects which adds an extra cost in terms of time, but in other scenarios 'cleaning' the environment could not be allowed.

EPIC-Diff. For this fact we evaluated our scene reconstructions on a subsample[1] of the EPIC-KITCHENS extension proposed in NeurlDiff [1],known as EPIC-Diff. In this extension the authors of NeuralDiff added manually pixelwise segmented masks for ten scenes of which we just considered P01-01,P03-04,P04-01,P09-02,P16-01,P21-01.

VISOR. We looked also for the more recent VISOR [3] dataset in which pixel annotations of hands and active objects are given but unfortunately their definition of *active* was not suitable for our work. They labeled as active any object that is included in the current action, so it is common to see as active the sink or the gas stove, but in our case they should be considered as static(see Figure 2.1).

[1] Come giustifichiamo l'aver preso non tutte le scene? Per motivi di tempo è accettabile?

Scene	Frames
P01-01	98935
P03-04	100251
P04-01	69292
P09-02	22187
P16-01	74592
P21-01	41583

Table 2.1: Total Frames for each scene

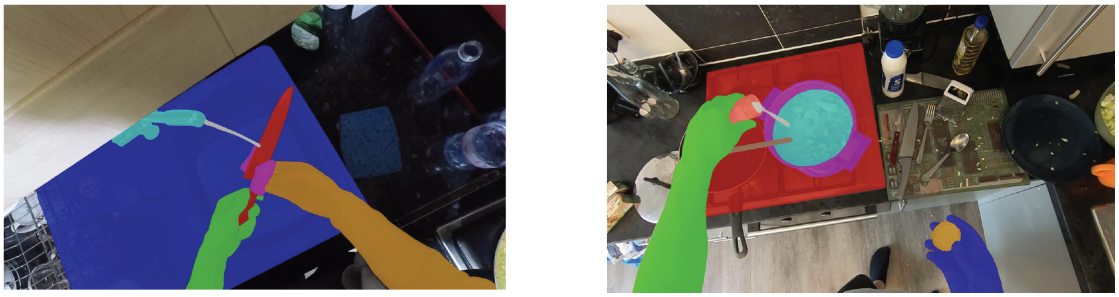

Figure 2.1: Example of VISOR active annotations, on the left 'wash a knife' include the static sink as active; on the right 'pour spice' static gas stove is active

2.2 Metrics

As regards metrics we looked in literature for a way to evaluate our results but unfortunately each method involved a ground truth which for our dataset is not available. Possible ways to obtain a groundtruth could be manual annotations or simulating the environments. Both these two methods would take a considerable large amount of time and are also beyond the scope of this thesis.

For this reason we ended up by using the metrics proposed in [1]. Namely these are:

- **PSNR**
- **mAP**

2.2.1 PSNR:Peak signal-to-noise ratio

The Peak Signal-to-Noise Ratio (PSNR) is a metric commonly used in image and video processing to quantify the quality of a reconstructed or processed signal, like an image or video. It gives a measures of the ratio between the maximum possible

power of a signal (MAX) and the power of the distortion or noise that affects the signal (MSE).

The formula for PSNR is usually expressed in decibels (dB) and is given by:

$$\text{PSNR} = 20 \cdot \log_{10} \left(\frac{\text{MAX}}{\text{MSE}} \right)$$

where:

- MAX is the maximum possible pixel value of the image (1 in our case).
- MSE is the Mean Squared Error, which represents the average squared difference between the original signal and the reconstructed or distorted signal.

It is worth noting that a high PSNR does not guarantee that the processed signal will be perceived as visually pleasing or high-quality by humans, especially in the case of perceptually sensitive applications like image and video compression.

2.2.2 AP:Average Precision

Average Precision (AP) is a metric commonly used in object detection and information retrieval to evaluate the performance of machine learning models. It measures the *precision-recall* trade-off of a model.

It can be useful to remind what *Precision* and *Recall* are. Namely:

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

where:

- TP=True positive
- FP=False positive
- TN=True negative

Average precision is then computed as the area below the precision-recall curve, specifically the curve obtained by varying the confidence threshold of the inference model as shown in Figure 2.2. That is why it can also be found in literature as AUC(Area Under Curve). Its scalar value summarize the precision-recall performance of the model. A higher AP is desirable, indicating a model that effectively retrieves relevant instances while minimizing false positives.

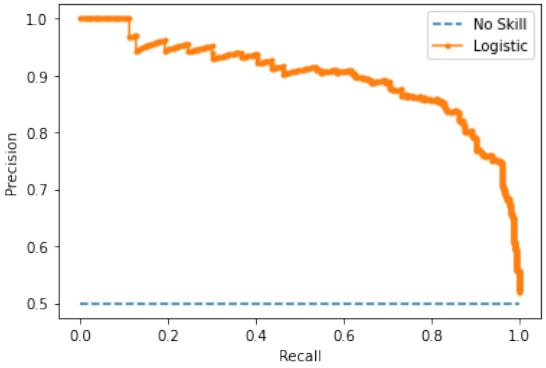

Figure 2.2: Example of Precision-recall curve.We can see how the bottom line model represents the worst a model can perform, e.g. predict every sample as it is coming from the same class,if the dataset is balanced. A better model would *tend* to the upper-right corner, which instead represents the best possible model, a model that have maximum precision and recall.

2.3 COLMAP Reconstruction

Once we have fixed the data we were working on, we proceeded to do some experiments on COLMAP reconstructions. In particular we took the scene P01-01 and tried varying both the number of frames and their resolution for the reconstruction. In fact most of the scene have too many frames to handle, which could results in out of memory issues or at the least worst in a long computational time. Our aim was to find a good compromise between *quality of reconstruction* and *computational time*.

Quantity vs Quality The first thing we did was to subsample the frames using the same technique as reported in Epic FIELDS [2] and explained in Section 1.3. We report the results of the COLMAP reeconstructions both quantitatively and qualitatively in Table **??** and Figure **??**. It is worth noting few things watching these references. The first one is that the resolution plays an important role in the successfulness of the reconstruction as we can see from Table 2.3. The same split of frames is reported and the right one at a resolution of 114x64 failed. This is due to the feature extractor, that in a high resolution image can retrieve informations that instead are lost in low resolution frames. A lack of significant features means no matching between images so the reconstruction has very few frames matched. The second thing is that the higher the frames the better. In fact chances of matching increases and also we will have more areas of the environment covered, as shown in Figure 2.3. We can see that augmenting the number of frames more parts of the kitchen are revealed, *e.g.* the round table at the center of the room, the sideboard

Scenes	P01_01_04	P01_01_06	P01_01_08	P01_01_09
Initial Frames	1231	1487	2598	5223
Reconstructed Frames	648	911	2045	4741
PCD points	152763	204024	460914	1079375
Duration	44min 14s	1h 12min 34s	3h 6min 32s	10h 41min 8s
Feature Extraction	7s	8s	13s	28s
Exhaustive Matcher	33s	49s	2min 32s	10min 22s
Mapper	23min 57s	44min 5s	2h 1min 25s	8h 18s
Image Undistorter	0s	1s	1s	2s
Patch Match Stereo	19min 28s	27min 15s	1h 1min 18s	2h 24min 16s
Stereo Fusion	9s	16s	1min 3s	5min 42s

Table 2.2: Comparison of Recostruction details for scene P01_01 using different Initial frames at same resolution of 228x128. The higher the frames, the better the reconstruction but at a higher computational time.

Scene	P01_01_04	P01_01_04	P01_01_04
Initial Frames	1231	1231	1231
Reconstructed Frames	765	648	6
Resolution	456x256	228x114	114x64
PCD points	**629270**	152763	-
Duration	1h 7min 5s	44min 14s	-
Feature Extraction	16 s	7s	-
Exhaustive Matcher	42s	33s	-
Mapper	18min 35s	23min 57s	-
Image Undistorter	1s	0s	-
Patch Match Stereo	47min 1s	19min 28s	-
Stereo Fusion	30s	9s	-

Table 2.3: Comparison of Recostruction of scene P01_01 using different Resolutions. The higher the resolution, the better. Too low resolution, as 114x64 in this case can lead to a unsuccessful reconstruction.

in front of the sink. But also some important objects that are visible from the video,like dishes on top of the table. This is a keypoint to the development of our pipeline, because we need to be sure that actually the scene contains points deriving from the motion of objects.

By considering these results and always keeping in mind the time of computation at our disposal we opted to feed the next pipeline with around five thousands frames at a resolution of 228x128. The pipeline of NeuralDiff in fact is really heavy and working at full resolution was prohibitive in the number of experiments we could try.

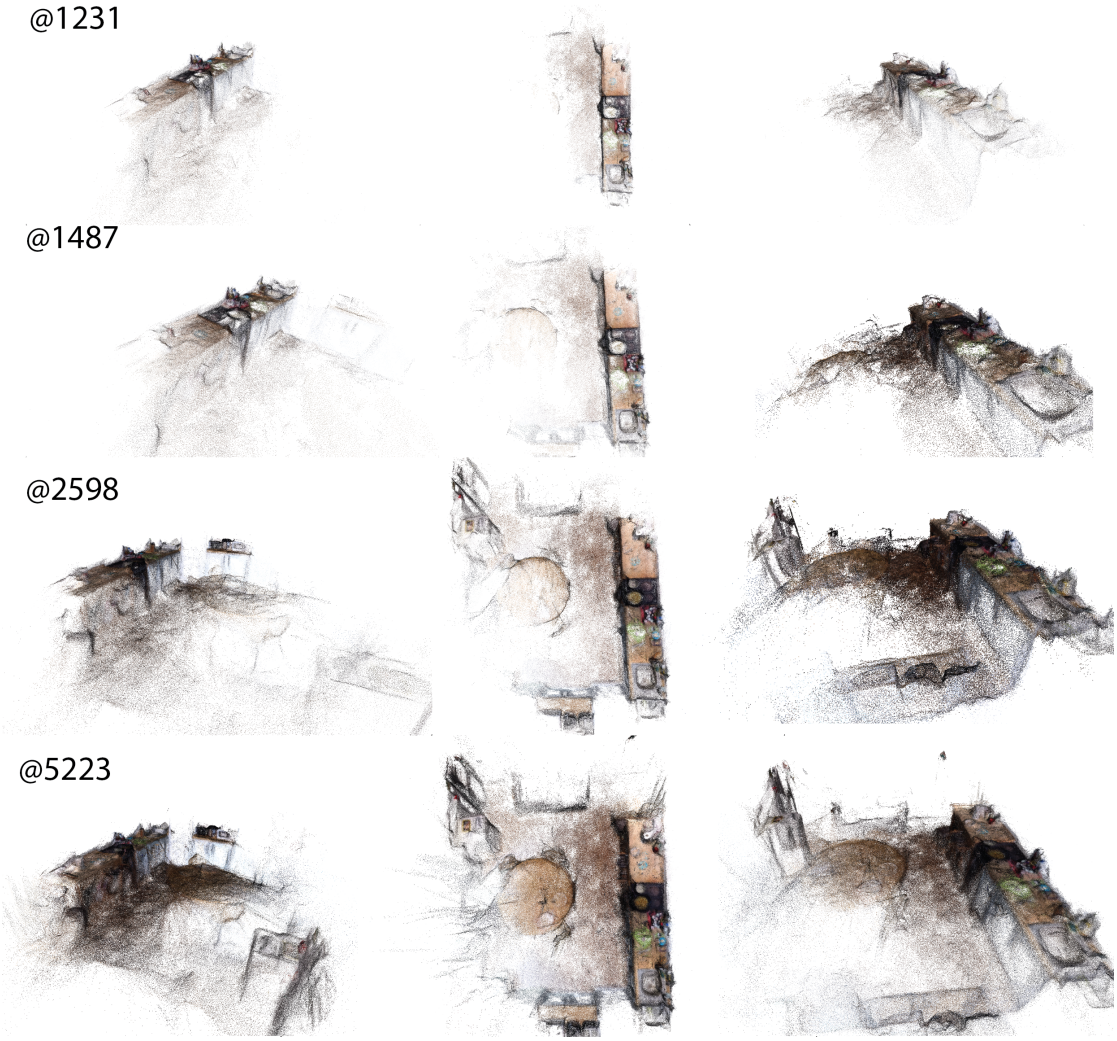

Figure 2.3: Different COLMAP pcd reconstructions changing number of samples. Each row is the same reconstruction viewed from different viewpoints.From top to bottom the number of frames increase. We can see how the number of frames positively affect the reconstruction.

Figure 2.4: Different COLMAP pcd reconstructions changing resolution. Each row is the same reconstructions viewed from different viewpoints. The first report the reconstruction for a resolution of 456x256 and the bottom one the half resolution. It is clear how the resolution has a beneficial impact on the overall reconstruction.

Here in Figure **??** I report also some visualization of other kitchens.

2.4 Monocular Pipeline

Here I report the qualitative result obtained from the Monocular Pipeline. As expected the results are really poor. The main reason of the failure of this technique is due to the inaccuracy of the depth estimator. In Figure 2.5 we can see that in some scene it seems to capture the main elements, like the hands in scene P01-01 or the pot in scene P16-01. However others like P09-02 seem to completely get it wrong.

Once the frames are projected in the environment space we also have to find a threshold for the distance at which a reconstruction point is labeled as dynamic or static. This make the pipeline highly scene-specific requiring each time a lot of fine tuning for a mediocre result.

Also using a distance principle for segmentation, we can see how the scene is deteriorated in this form of globular groups of points(see Figure 2.6). While in Figure 2.7 we can see how the segmentation of what is dynamic and what is static change changing the distance threshold. In particular the scene take was P03-04 and the values reported are 0.5, 5 and 20.

Figure 2.5: Different scenes where monocular depth estimation was performed. In particular on the right we can find the frame that is instead projected(in red) on the left in the 3D reconstruction of that kitchen. The red frustum(pyramid) represents the camera position and orientation in the space.

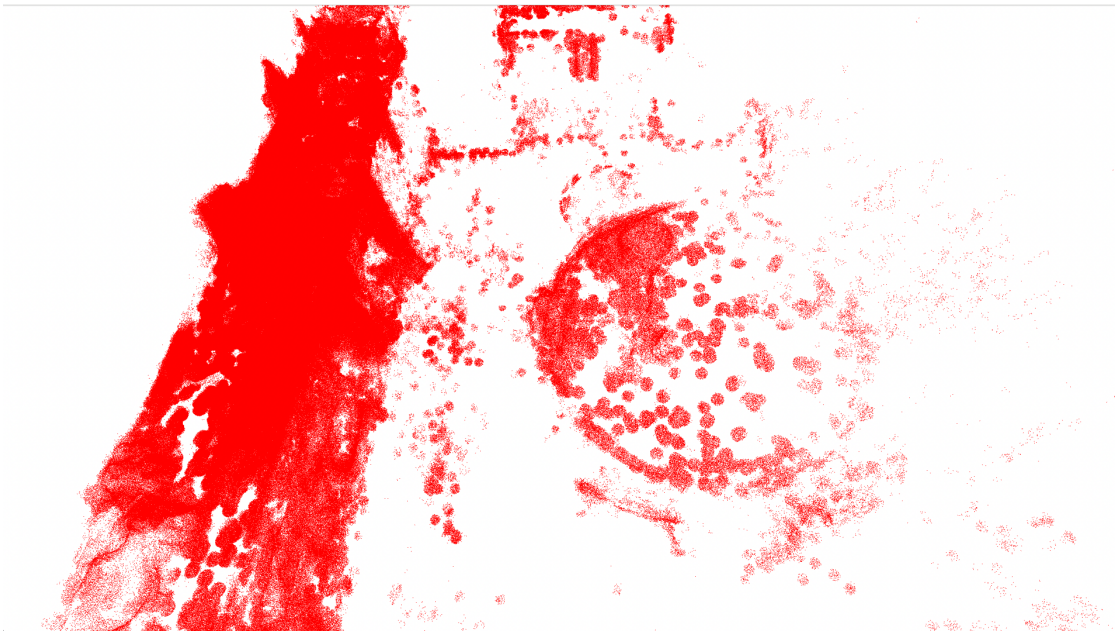

Figure 2.6: Dynamic points segmented in scene P01-01 using Monocular Pipeline.

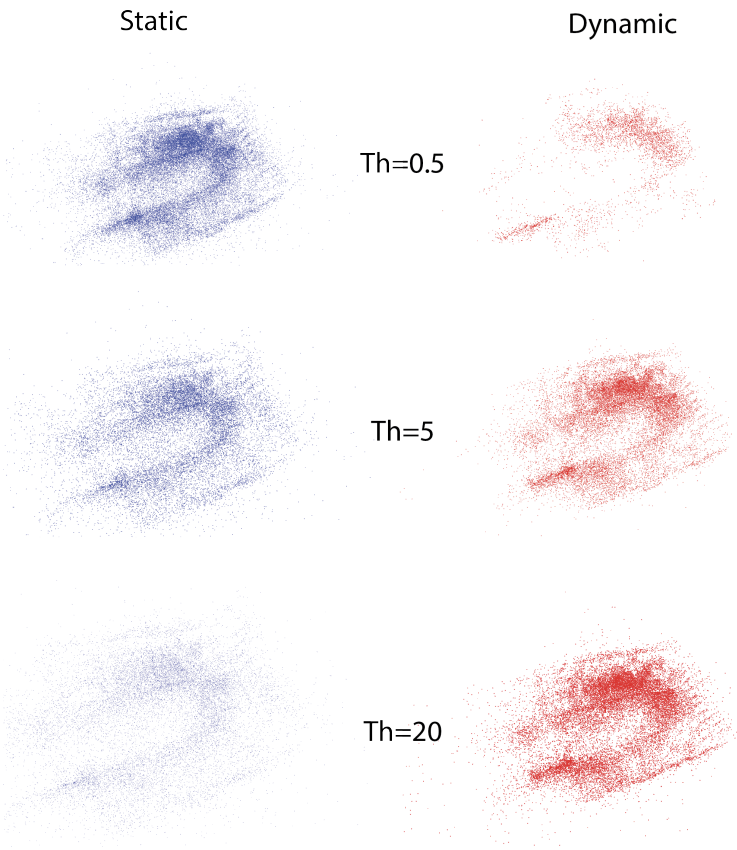

Figure 2.7: Different Segmentation changing the distance threshold. Each point is segmented as dynamic if its distance from a pixel projected in 3D space is less than a threshold Th. The scene is P03-04 and in the left side we can find the static part while in the right side we have the dynamic points. Here we can notice how this method is pretty inaccurate.

	Original	Sampled	Reconstructed	Obtained	Int/Unif Samples	Threshold
P01-01	98935	5223	4741	3652	**1089**	0.9
P03-04	100251	5060	4522	3654	**868**	0.855
P04-01	69292	4269	3242	2144	**1098**	0.89
P09-02	22187	4953	4398	3495	**903**	0.975
P16-01	74592	5531	5480	4476	**1004**	0.945
P21-01	415853	4655	4588	3733	**855**	0.94

Table 2.4: Split$\sim$ 1000. Number of frames resulting from the different sampling steps. In particular from Original the frames are reduced with the Homography filter to remove redundancy and keep overlap. The reconstructed frames are the ones which were successfully reconstructed by COLMAP. The obtained ones are the reconstructed frames filtered again with the homography filter. Int/Unif Samples are the reconstructed frames without the Obtaineds. The thresholds reported are referred to the last Homography filter step.

	Originali	Sampled	Reconstructed	Obtained	Int/Unif Samples	Threshold
P01-01	98935	5223	4741	4019	**722**	0.91
P03-04	100251	5060	4522	3839	**683**	0.86
P04-01	69292	4269	3242	2463	**779**	0.896
P09-02	22187	4953	4398	3776	**622**	0.977
P16-01	74592	5531	5480	4837	**643**	0.95
P21-01	415853	4655	4588	3942	**646**	0.945

Table 2.5: Split$\sim$ 700. Number of frames resulting from the different sampling steps. In particular from Original the frames are reduced with the Homography filter to remove redundancy and keep overlap. The reconstructed frames are the ones which were successfully reconstructed by COLMAP. The obtained ones are the reconstructed frames filtered again with the homography filter. Int/Unif Samples are the reconstructed frames without the Obtaineds. The thresholds reported are referred to the last Homography filter step.

2.5 Sampling Frames

In order to work with the NeuralDiff pipeline that follows this section, we have to further downsample the frames reconstructed by COLMAP to have some reasonable computational times. We found that at a resolution of $114 \sim 1000$ a scene took $\sim 4h$ while at $228 \sim 700$ a scene took $\sim 11h$. In Table 2.5 and Table 2.4 we can see the number of frames kept at each sampling step.

2.6 NeuralDiff Pipeline

Different number of samples for the same scene. To evaluate our final pipeline we started by creating the splits upon which we would have trained the neural render. In particular we focused on one scene, P01-01, to see how the number of frames selected and the method which selects them affect the pipeline performances. In Figure 2.8 it is given a visualization of the three different subsampling obtained using the methods presented in Section 1.3 for a total of 217 frames.

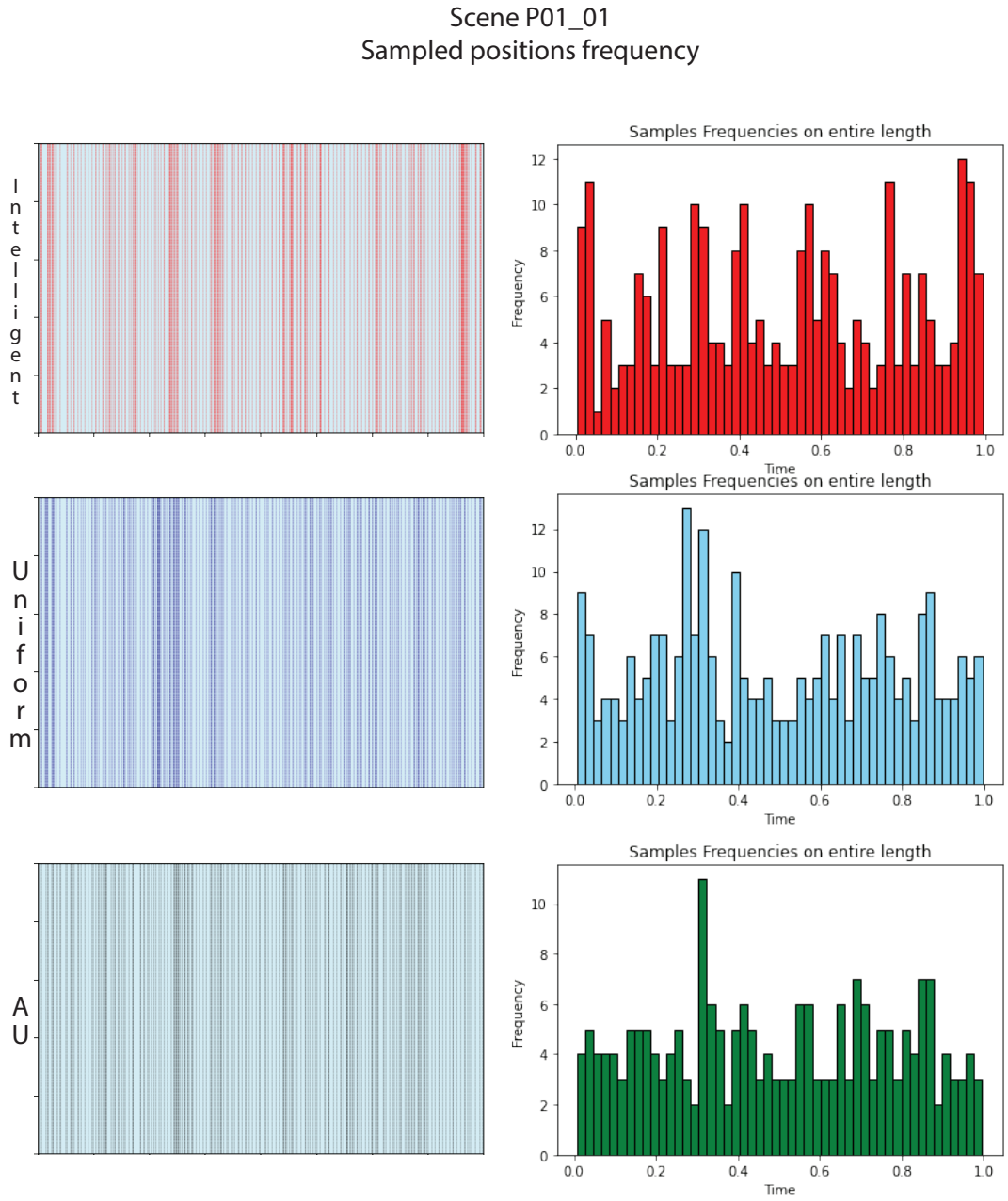

Figure 2.8: Visualization of the sampling of scene P01-01 for the three different methods: Intelligent, Uniform and AU using 217 frames in total. The left box is a proposal we gave to visualize how the frames actually spread along the temporal axis, where a line is drawn in correspondence of each sample. The right boxes represent instead histograms with the frequencies of the sample on the entire duration of the video.

P01-01	Sampling	Durata [s]	PSNR	PSNR statico	mAP
2938	Int.	11 h36 min 59 s	**24.82**	20.41	72.21
2938	Unif	11 h16 min8 s	24.42	20.41	**72.79**
2015	Int.	8 h3 min50 s	**24.51**	**20.46**	**70.49**
2015	Unif	7h 52min 34s	23.93	20.33	69.87
1000	Int.	3h 43min 41s	**23.59**	**20.37**	67.55
1000	Unif	3 h43 min1 s	22.8	20.10	66.51
1000	AU	4 h2 min45 s	23.43	20.31	**67.99**
722	Int.	2h 34 min	**22.65**	**20.20**	**65.42**
722	Unif	2 h30 min7 s	22.09	19.65	62.95
722	AU	2h 36 min46 s	22.48	20.05	64.10
397	Int.	1h 19 min53 s	**21.33**	19.64	56.63
397	Unif	1h 21 min 42 s	20.98	19.54	**61.6**
397	AU	1h 14min 36s	21.2	**19.95**	59.97
217	Int.	40 min55 s	20.32	**19.60**	51.69
217	Unif	41 min8 s	**20.51**	19.42	**53.00**
217	AU	25 min39 s	20.26	19.39	50.58

Table 2.6: NeuralDiff Pipeline Results on P01-01 at 114x64. For the same scene P01-01 results of NeuralDiff pipeline trained on different amount of frames are reported.The Frames are selected using the three different sampling strategis:Intelligent,Uniform and AU(see Section 1.3). The frames are all at a 114x64 resolution.

We then proceeded in testing the various split for P01-01 obtaining the results reported in Table 2.6. As we can see the Intelligent sampling is actually working. The PSNR is always higher with respect to the other methods. It can be seen that actually all methods suffer the scarcity of frames and as a matter of fact in the last split, 217, uniform sampling beats the intelligent one. For the static PSNR instead the Intelligent method is always better than the uniform one, even at low frames. For the mean Average Precision instead we can see some oscillations, but we have to be careful since our mask is actually combining the actor and the foreground layer, meaning that the average precision is not actually assessing the ability of the model to distinguish these two parts. For example in Figure 2.9 we can see that in the 400 frames splits is exactly present this deficit, where the uniform sampling is totally unable to detect the actor even though its mAP is higher than the Intelligent one($mAP_{Uniform}61.6\% > mAP_{Intelligent}56.63\%$).

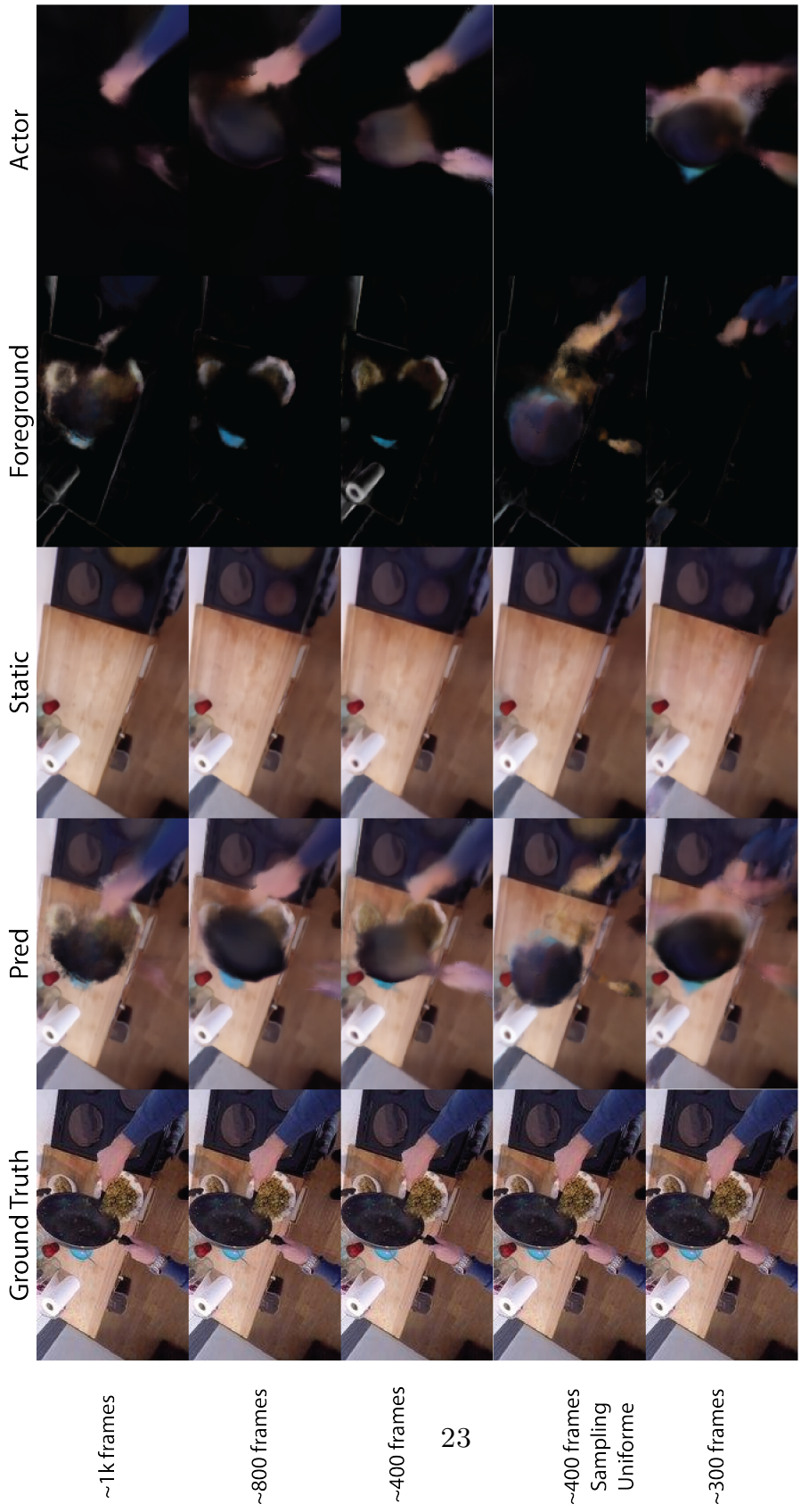

Figure 2.9: Qualitative results on P01-01 at 228x128 Visualization of the output of the different models trained on different sampling splits for scene P01-01. The first coloumn represent the real frame while the next ones are respectively: the predicted image,which is the combination of:the static part, the foreground and the actor part.

Scene	Sampled Frames	Sampling	Durata [s]	PSNR	Improv.	PSNR statico	Improv.	mAP	Improv.
P01-01	1089	Intelligent	3h 43 min41 s	23.59	0.79	20.37	0.27	67.55	1.04
		Uniform	3 h43 min1 s	22.8		20.10		66.51	
P03-04	868	Intelligent	3h 14 min44 s	19.90	0.77	16.73	0	61.92	-2.08
		Uniform	3h 10 min48 s	19.13		16.73		64.00	
P04-01	1098	Intelligent	4h 10 min 12 s	24.32	0.51	21.23	0.74	71.37	5.56
		Uniform	4h 2 min 9*s*	23.81		20.49		65.81	
P09-02	903	Intelligent	3 h25 min58 s	23.97	0.58	19.43	-0.02	60.05	-1.18
		Uniform	3 h17 min5 s	23.39		19.45		61.23	
P16-01	1004	Intelligent	3h 46 min57 s	22.89	0.14	20.17	0.23	66.89	3.28
		Uniform	3h 43min 11s	22.75		19.94		63.61	
P21-01	855	Intelligent	3h 15 min 59 s	20.02	0.91	15.73	0.66	72.94	4.06
		Uniform	3 h7 min50 s	19.11		15.07		68.88	

Table 2.7: NeuralDiff models trained on different scenes at ∼1000frames, resolution 114x64. The coloumn Improv represents the difference between the previous coloumn of the Intelligent split minus the Uniform one.

Also for our aim to segment dynamic objects, we are interested in the static PSNR (as we obtain dynamic objects as what is NOT static) and Figure 2.9 shows us that the static part is almost identical for each scene.

Qualitative Results On the other hand,going towards qualitative results, here we present the 3D static reconstruction for P01-01. As shown in Figure 2.10,the first row is the COLMAP pointcloud extracted from the sampled videosequence. Below are placed instead the static reconstructions for the three different sampling strategies. The first thing that comes to our eyes is the overall colour which in the Intelligent sampling seem more faithful to the reality. The second thing is the segmentation of the plate on the table top, which can be seen in the COLMAP row. The plate is successfully removed in the Int. sampling while it is still visible in the other splits, although the best model was the Unif. one according to the metrics. Another example is given by the pan highlighted with the green circle which is removed in the Intelligent sampling while not in the others. We can also look at scene P03-04 in Figure **??** where Intelligent method manage to remove the can highlighted in red while the Uniform methods can not.

Other comparison for the same scene with different frames splits are provided in Figure 2.11. Bastano questi esempi sul 3D?

NeuralDiff Pipeline on all Scenes. To further validate our results, we repeated the experiments using different scenes using a split of ∼1000 frames and one of ∼ 700. The results are reported in Table 2.7 for resolution 114x64 and Table 2.8 at a resolution of 228x128. The results assess that our method is working.

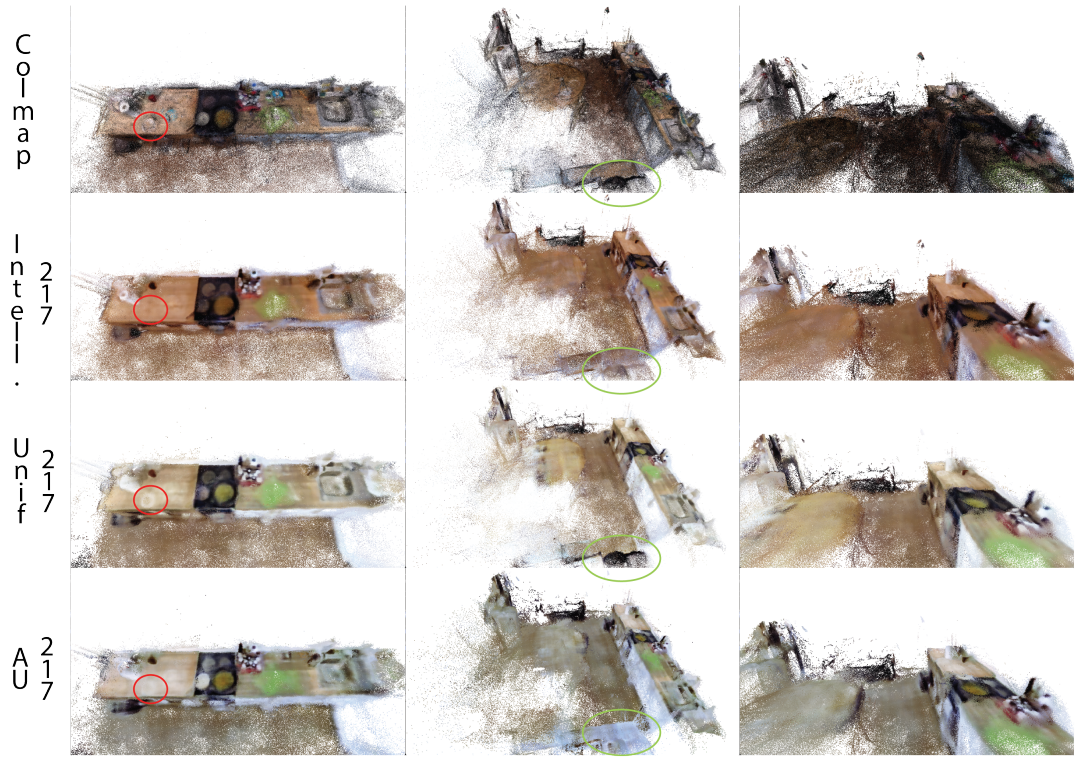

Figure 2.10: Qualitative results for the static reconstruction of P01-01 scene at 217 frames. In red is highlighted a dynamic plate, while in green a dynamic pan.

Scene	Sampled Frames	Sampling	Durata [s]	PSNR	I-U	PSNR statico	I-U	mAP[%]	I-U
P01-01	722	Intelligent	10 h25 min35 s	21.64	0.18	19.58	0.08	66.26	1.93
		Uniform	10h 5 min 7 s	21.46		19.50		64.33	
P03-04	683	Intelligent	9h 40 min5 s	18.89	0.39	16.17	-0.15	61.08	-1.09
		Uniform	9h 16 min59 s	18.5		16.32		62.17	
P04-01	779	Intelligent	10h 45 min 46*s*	22.15	0.81	20.04	0.46	67.65	4.81
		Uniform	11h 17 min34 s	21.34		19.58		62.84	
P09-02	622	Intelligent	8 h35 min	22.39	0.96	19.10	0.22	67.56	9.34
		Uniform	8 h53 min13 s	21.43		18.88		58.22	
P16-01	643	Intelligent	9*h*5 min 38*s*	21.25	0.38	19.38	0.39	63.34	-1.2
		Uniform	9h 7 min 52s	20.87		18.99		64.54	
P21-01	646	Intelligent	9*h*8 min 54*s*	18.09	-0.23	15.20	-0.28	65.49	-3.11
		Uniform	9 h8 min 7 s	18.32		15.48		68.60	

Table 2.8: NeuralDiff models trained on different scenes at ∼700frames, resolution 228x128. The coloumn I-U represents the difference between the previous coloumn of the Intelligent split minus the Uniform one.

Action positions and samples positions. Other than the idea behind the Intelligent sampling which was expressed in Section 1.3, we found a link between

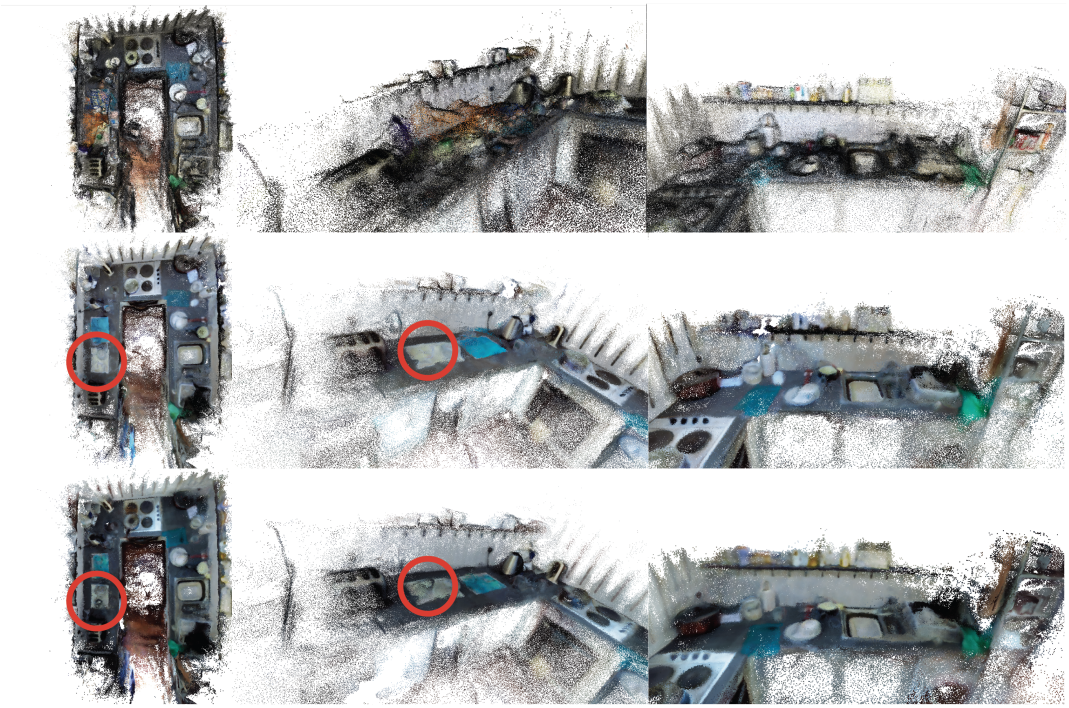

Figure 2.11: Qualitative results for the static reconstruction of P03-04 scene. In red is highlighted a dynamic plate, while in green a dynamic pan.

the positions of the sampled freqencies and the frequencies of the objects that were used in different equispaced time intervals. In Figure 2.8 the three methods sampling frequencies are reported. In Figure 2.13 and Figure **??** instead are reported respectively the comparison of Intelligent and Uniform sampling with the objects count for scene P01-01 with varying number of samples, as can be read on each sub-figure; and the comparison of Intelligent and Uniform sampling with the objects count for each scene with fixed sampling at $\tilde{1}000$ frames.

As we can see from the plots,at exception from few scenes, the profile of the Intelligent sampling looks closer to the one of the object counts. This is giving a further explanation of the functioning of our proposed method. In fact it means that our sampling is focusing on those areas where a lot of actions are performed. In this way long redundant actions are filtered and only relevant frames are kept.

We also tried to give a quantitative measure of similarity and dissimilarity by comparing some different metrics: Cosine Similarity, Kullback-Leibler Divergence (KLD), Jensen-Shannon Divergence (JSD), Correlation Coefficient. Le devo spiegare? E' meglioo se le metto in Method?

Avrei anche questa tabella ma il sampling anti/uniform sono sbagliati. potrei rifarli? Altrimenti non abbiamo messo la parte di simplyfing NeuralDiff? Figure 2.15

	Cosine Similarity		K-L Div.		JS-Div		Correlation	
	Intelligent	Uniform	Intelligent	Uniform	Intelligent	Uniform	Intelligent	Uniform
P01-01	0.91319715	0.90718451	0.113071	0.13421	0.03931	0.0450167	0.5632	0.5391
P03-04	0.7638	0.6773	0.4224	0.5389	0.16092	0.2025	0.5483	0.2226
P04-01	0.6528	0.6316	0.5892	0.6631	0.2196	0.2438	0.1454	0.1353
P09-02	0.7438	0.7323	2.2962	1.2859	0.1377	0.1399	-0.1736	-0.1152
P16-01	0.8029	0.8130	0.2170	0.2431	0.0727	0.0815	0.3490	0.4416
P21-01	0.8804	0.8761	0.1694	0.1760	0.0574	0.0611	0.3633	0.1889

Table 2.9: Metrics for comparing the profile of the histograms. In particular higher values of Cosine similarity and Correlation indicates similarity; while the value of the two divergences represents the distance between the two distributions.

Non ho messo da nessuna parte il risultato dei sampling. Aggiungere la tabella..!

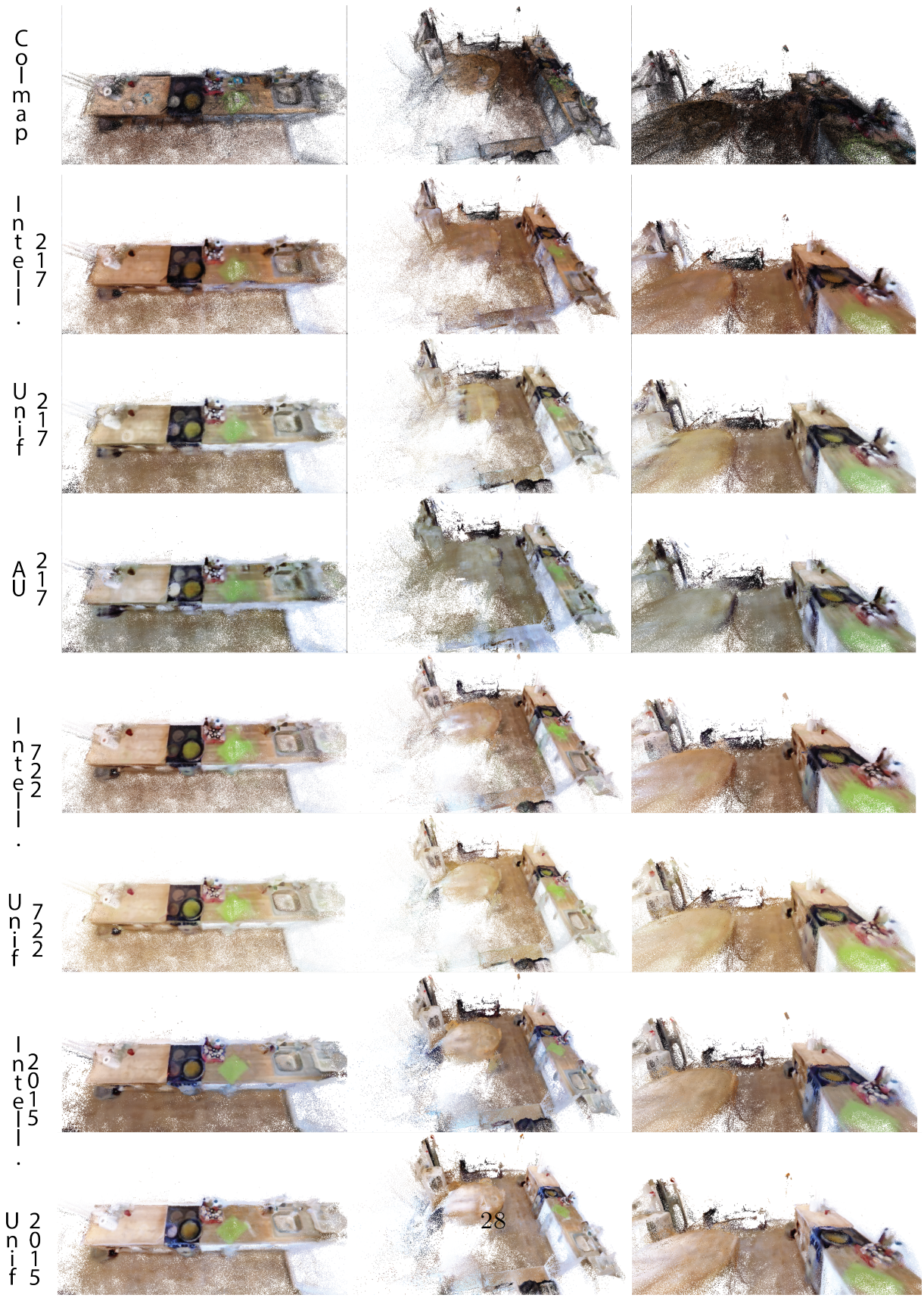

Figure 2.12: Comparative of the qualitative results for different samplings of the P01-01 scene.

Epic_finale_228 con durata 700 228x128

Scene	Sampled Frames	Sampling	Durata [s]	PSNR	PSNR statico	mAP
P01_01	**1089**	Intelligent				
		Uniform				
P03_04	**868**	Intelligent	9h 40min 5s	18.89	16.17	61.08
		Uniform	9h 16min 59s	18.5	16.32	62.17
P04_01	**1098**	Intelligent	10h 45min 46s	22.15	20.04	67.65
		Uniform	11h 17min 34s			
P09_02	**903**	Intelligent	8h 35min	22.39	19.10	67.56
		Uniform	8h 53min 13s	21.43	18.88	58.22
P16_01	**1004**	Intelligent	9h 5min 38s	21.25	19.38	63.34
		Uniform	9h 7min 52s	20.87	18.99	64.54
P21_01	**855**	Intelligent	9h 8min 54s	18.09	15.20	65.49
		Uniform	9h 8min 7s	18.32	15.48	68.60

Figure 2.13: Experiments performed on 228x128 frames for each scene

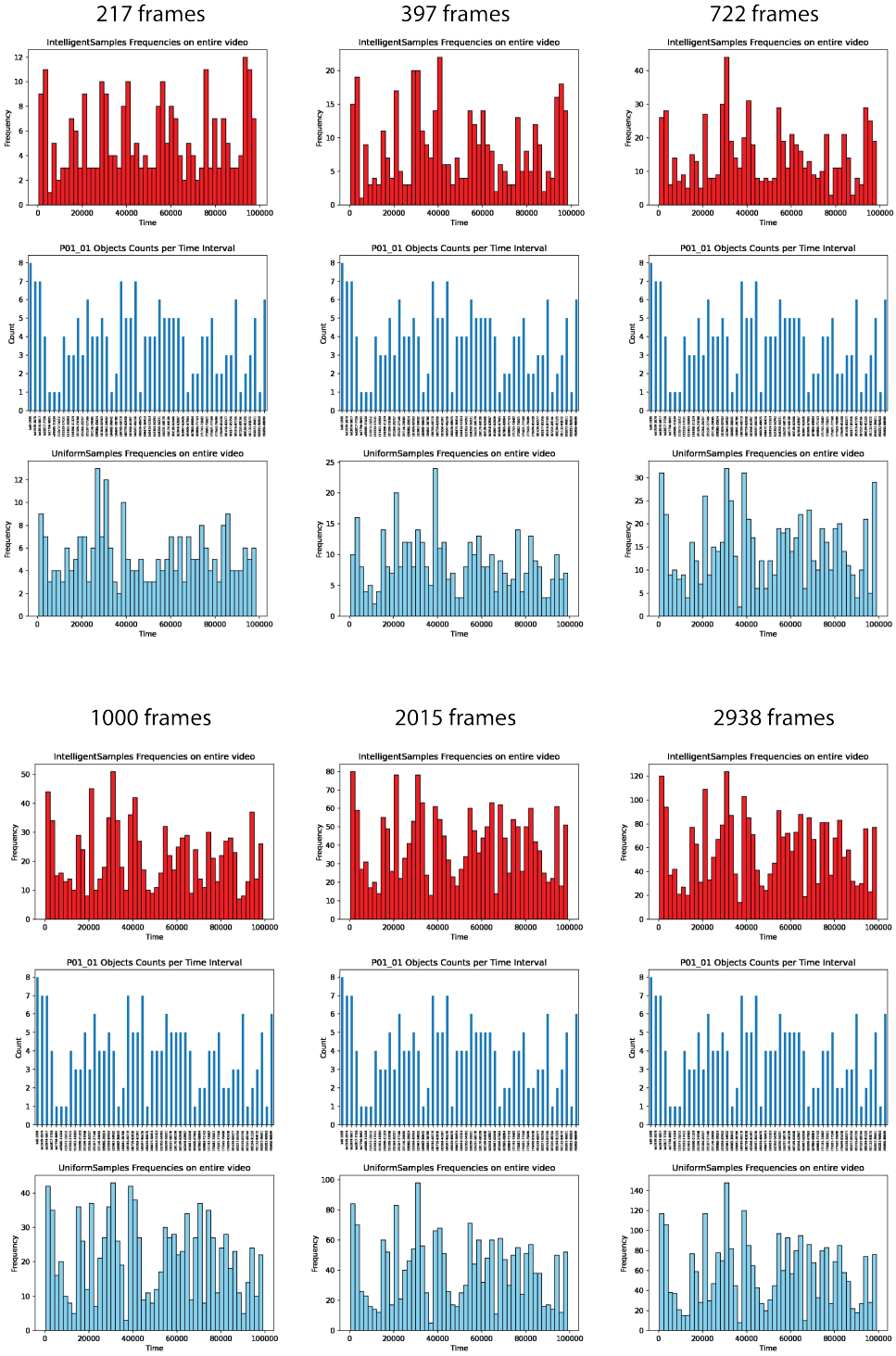

Figure 2.14: Comparison of frequencies for the Intelligent and Uniform sampling with the Object Count for the P01-01 scene changing the total number of sampled frames.

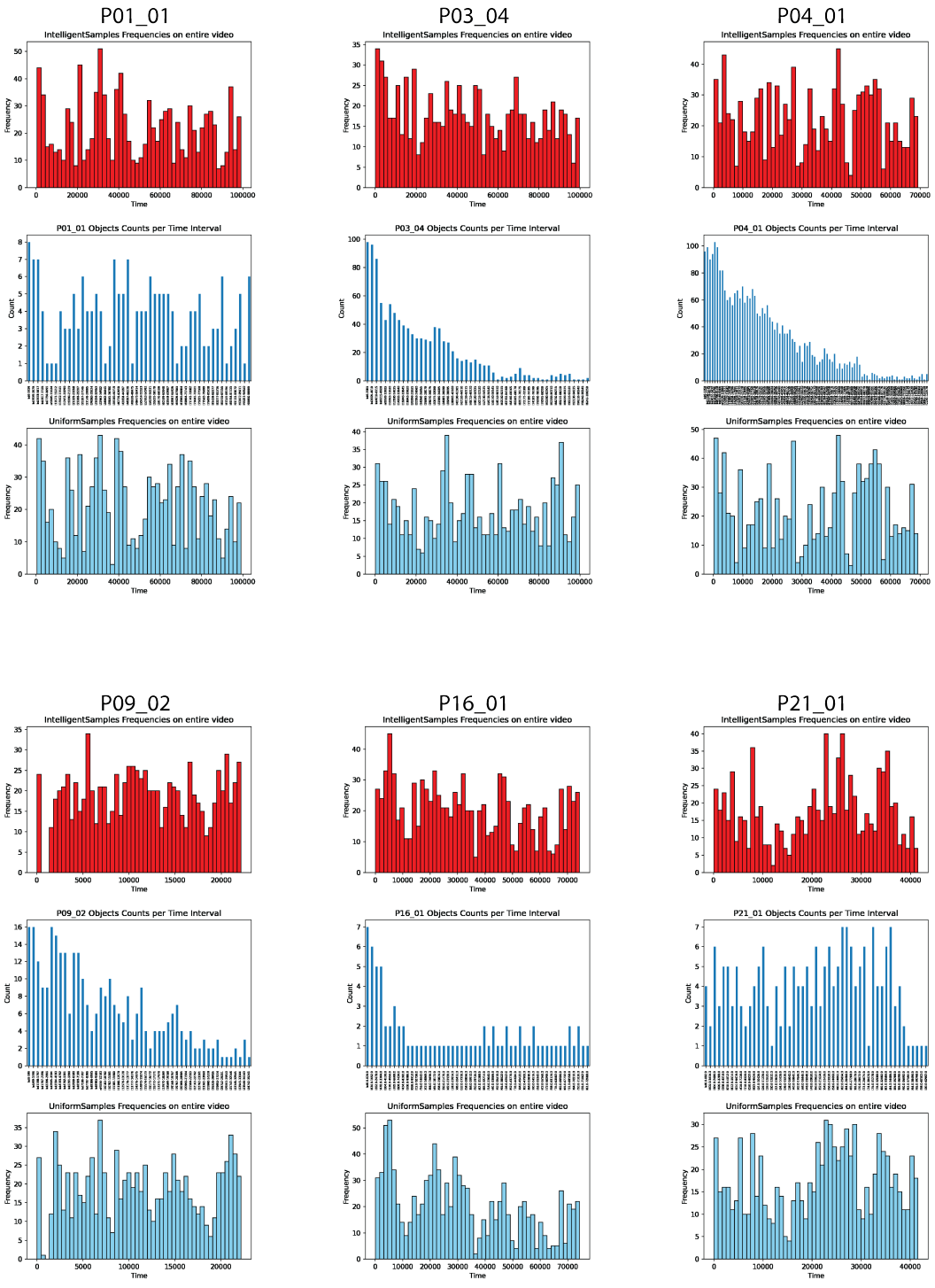

Figure 2.15: Comparison of frequencies for the Intelligent and Uniform sampling with the Object Count for each scene at a fixed split $\tilde{1}000$ frames.

NeuralCleaner on 114

P01_01	Sampling	Resolution	Durata [s]	PSNR	PSNR statico	mAP	Inizio	Fine
1089	Altro	114x64	3h 7min 53s	23.49	19.92	61	Fri 09 Feb 2024 11:57:07 PM CET	Sat 10 Feb 2024 03:05:00 AM CET
1089	Unif	114x64	3h 15min 54s	22.86	19.74	57.49	Fri 09 Feb 2024 11:55:36 PM CET	Sat 10 Feb 2024 03:11:30 AM CET
722	Altro	114x64	2h 2min 38s	22.51	19.47	50.03	Sat 10 Feb 2024 04:07:42 AM CET	Sat 10 Feb 2024 06:10:20 AM CET
722	Unif	114x64	2h 11min 45s	22.77	19.67	56.42	Sat 10 Feb 2024 04:30:48 AM CET	Sat 10 Feb 2024 06:42:33 AM CET
397	Altro	114x64	1h 2min 40s	21.46	19.72	53.69	Sat 10 Feb 2024 03:05:01 AM CET	Sat 10 Feb 2024 04:07:41 AM
397	Unif	114x64	1h 19min 16s	21.61	19.47	53.07	Sat 10 Feb 2024 03:11:31 AM CET	Sat 10 Feb 2024 04:30:47 AM CET
217	Altro	114x64	32min 48s	20.55	18.2	40.87	Sat 10 Feb 2024 06:10:20 AM CET	Sat 10 Feb 2024 06:43:08 AM CET
217	Unif	114x64	49min 42s	21.12	19.33	49.73	Sat 10 Feb 2024 06:42:34 AM CET	Sat 10 Feb 2024 07:32:16 AM CET

Figure 2.16: Results for architecture with foreground+actor= fused

Appendix A

Galileo

```
    import os
    os.system("echo 1")
```

$\mathcal{O}(n \log n)$
`numpy`

Appendix B

Math Notation

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$\mathbf{a} \times \mathbf{b} = [\mathbf{b}]_{\times}^{T} \mathbf{a} = \begin{bmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Bibliography

[1] Vadim Tschernezki, Diane Larlus, and Andrea Vedaldi. «NeuralDiff: Segmenting 3D objects that move in egocentric videos». In: *CoRR* abs/2110.09936 (2021). arXiv: 2110.09936. URL: https://arxiv.org/abs/2110.09936 (cit. on pp. 5, 10, 11).

[2] Vadim Tschernezki, Ahmad Darkhalil, Zhifan Zhu, David Fouhey, Iro Laina, Diane Larlus, Dima Damen, and Andrea Vedaldi. *EPIC Fields: Marrying 3D Geometry and Video Understanding*. 2024. arXiv: 2306.08731 [cs.CV] (cit. on pp. 7, 13).

[3] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. *EPIC-KITCHENS VISOR Benchmark: VIdeo Segmentations and Object Relations*. 2022. arXiv: 2209.13064 [cs.CV] (cit. on p. 10).