POLITECNICO DI TORINO

Master's Degree in Mathematical Engineering

Master's Degree Thesis

Segmenting dynamic points in 3D scenarios

Supervisors

Prof. Tatiana TOMMASI

Prof. Chiara PLIZZARI

Candidate

Francesco BORGNA

March 2024

Summary

Ma che dici Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

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

Part I

Related Works

1. Epic Kitchens
2. Epic Fields
3. Photogrammetry
4. COLMAP
5. NeRF
6. NeuralDiff
7. Monocular Depth Estimation
8. motion estimation
9. (N3F)
10. (Gaussian Splatting)

Chapter 1

Datasets

Motivazioni per cui abbiamo usato questi datasets? The choice of the following datasets was dictated by the fact that up to our knowledge these are currently the largest datasets available in the egocentric scenarios. Let us see these in details.

1.1 EPIC-Kitchens

EPIC-Kitchens [**EPICKITCHENS**] is the largest and most varied dataset in egocentric vision up to our knowledge. It contains 55 hours of annotated video data recorded by a head-mounted camera of non scripted actions, meaning that the actors were not following any *scripted* actions(we will see this in more detail later).

1.1.1 Introduction/motivation

EPIC-Kitchens was born to fill the gap in the scarcity of annotated video datasets. As a leading comparison, at the time of writing significant progress have been seen in many domains such as image classification [**residualImage**],object detection [**fasterRCNN**], captioning [**captioning**] and visual question answering [**vqa**]; due to the advances in deep learning but mainly due to the availability of large-scale image benchmarks such as PASCAL VOC [**pascalImage**],ImageNet [**imagenet**],Microsoft COCO [**COCO**], ADE20K [**ADE20K**]. In the same way the authors thought that by introducing a large scale video dataset could contribute to the development of video domains.

Some video datasets were already available for action classification [**somethingSomething**, **yt**, **movieBench**, **movieQA**, **vlogs**] but, a part from [**movieQA**], these all contain very short videos, focusing on just a single action. A solution to this problem was given by Charades [**charades**] where 10k videos have been collected of humans performing daily tasks at home. The problem with this dataset is that the action

recorded were scripted, meaning that the actor had a text in which he was asked to perform some steps. In this way the actions lose their naturalness, their inbred evolving and multi-tasking properties.

To solve these problems they decided to focus on first-person vision, such that the recording would not interfere with the actor actions, increasing the possibilities of a succesful recording. Also, the viewpoint given by first-person vision allows us to record multi-task actions and the many different ways to perform a variety of important everyday tasks. In Table **??** we report a summary of the datasets compared by the authors.

Dataset	Ego?	Non-Scripted?	Native Env?	Year	Frames	Sequences	Action Segments	Action Classes	Object BBs	Object Classes	Participants	No. Env.s
EPIC-KITCHENS	✓	✓	✓	2018	11.5M	432	39,596	149*	454,255	323	32	32
EGTEA Gaze+ [16]	✓	×	×	2018	2.4M	86	10,325	106	0	0	32	1
Charades-ego [41]	70%✓	×	✓	2018	2.3M	2,751	30,516	157	0	38	71	N/A
BEOID [6]	✓	×	×	2014	0.1M	58	742	34	0	0	5	1
GTEA Gaze+ [13]	✓	×	×	2012	0.4M	35	3,371	42	0	0	13	1
ADL [36]	✓	×	✓	2012	1.0M	20	436	32	137,780	42	20	20
CMU [8]	✓	×	×	2009	0.2M	16	516	31	0	0	16	1
YouCook2 [56]	×	✓	✓	2018	@30fps15.8M	2,000	13,829	89	0	0	2 K	N/A
VLOG [14]	×	✓	✓	2017	37.2M	114 K	0	0	0	0	10.7 K	N/A
Charades [42]	×	×	✓	2016	7.4M	9,848	67,000	157	0	0	N/A	267
Breakfast [28]	×	✓	✓	2014	3.0M	433	3078	50	0	0	52	18
50 Salads [44]	×	×	×	2013	0.6M	50	2967	52	0	0	25	1
MPII Cooking 2 [39]	×	×	×	2012	2.9M	273	14,105	88	0	0	30	1

Table 1.1: Comparative overview of relevant datasets(action classes with > 50 samples)

1.1.2 Data Collection

To the data collection were involved 32 people in 4 cities in different countries(in North America and Europe): 15 in Bristol/UK, 8 in Toronto/Canada, 8 in Catania/Italy and 1 in Seattle/USA between May and Nov 2017. Participants were asked to record each time they visit the kitchen for three consecutive days, starting filming just before entering the kitchen and stopping before leaving it. They participated to the process of their own free will without being paid in any way.

Few requests were asked to them. The first was to be in the kitchen alone during the recording, such that no inter-person interaction could interfere. The second one instead was to remove all items that could disclose their identity, for example portraits or mirrors. In this way they could remain anonymous.

Each participant was equipped with a head-mounted camera with adjustable mounting such that it could be adapted to the participant's height and possibly different environment. They had to check, before each recording, the battery life and the viewpoint, such that their stretched hand were approximately located at the middle of the camera frame. The camera settings was set for most of videos to linear field of view, using 59.94fps as frame rate and Full HD resolution of 1920x1080, however some subjects made minor changes like wide or ultra-wide

FOV or resolution. In particular 1% of the videos were recorded at 1280x720 and 0.5% at 1920x1440. Also 1% at 30fps, 1% at 48fps and 0.2% at 90fps.

On average, each participant recorded 13.6 sequences, each of those lasted on average 1.7 h while the maximum duration recorded was of 4.6h. The duuration of the recording was obviously linked to the person's kitchen engagement. In Figure **??** we can see some statistics of the data acquired.

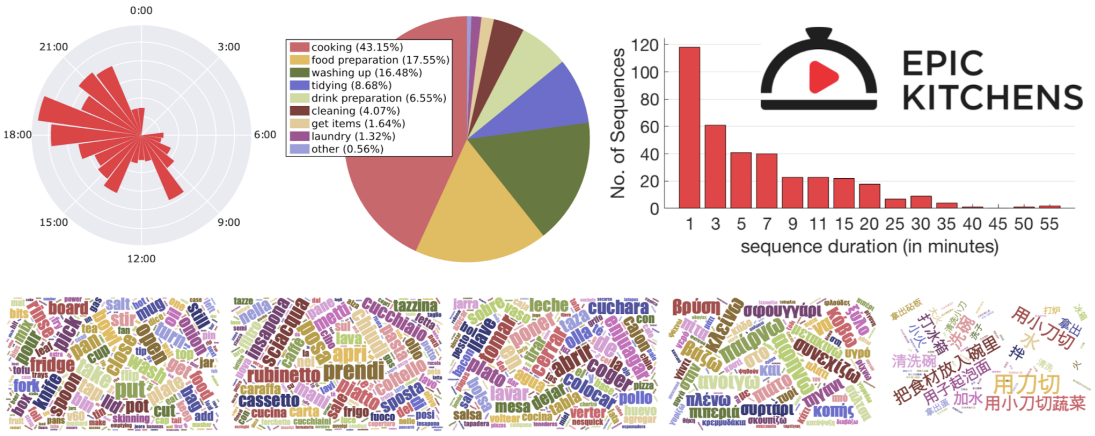

Figure 1.1: Top (left to right): time of day of the recording, pie chart of high-level goals, histogram of sequence durations and dataset logo; **Bottom**:Wordles of narrations in native languages(English, Italian, Spanish, Greek and Chinese).

1.1.3 Data Annotation pipeline

After the end of a sequence each participant was asked to watch the recording and narrate verbally the actions carried out to a microphone. The sound narration was chosen beacuse it was faster than a written one, and participants were thus more willing to provide these annotations. The guide lines for narrations are reported in Figure **??**.

The most used language was English, but other languages was used, if the participant was not so fluent in English. In particular a total of 5 languages were used: 17 people narrated in English, 7 in Italian, 6 in Spanish, 1 in Greek and 1 in Chinese.

The motivation to obtain the narrations directly from the actors was due to the fact that they surely knew what they were doing, avoiding misinterpreting some possible actions. The posthumous narration was instead motivated by the fact that actors could perform their actions in the most natural way, without being concerned about labelling.

Use any word you prefer. Feel free to vary your words or stick to a few.
Use present tense verbs (e.g. cut/open/close).
Use verb-object pairs (e.g. wash carrot).
You may (if you prefer) skip articles and pronouns (e.g. “cut kiwi” rather than “I cut the kiwi”).
Use propositions when needed (e.g. “pour water into kettle”).
Use ‘and’ when actions are co-occurring (e.g. “hold mug and pour water”).
If an action is taking long, you can narrate again (e.g. “still stirring soup”).

Figure 1.2: Narration Guidelines given to each participant to be followed after the completion of a recording.

The second step of annotations consists in the transcription of the speech narrations. After testing some automatic audio-to-text algorithms, which led to inaccurate transcriptions, they opted for manual transcriptions and translation via Amazon Mechanical Turk(AMT), a crowdsourcing marketplace that allows to do task that computers are still unable to complete. More in detail requests to AMT are called HIT(Human Intelligence Tasks). To ensure consistency, the authors divided speechs in chunks of around 30 seconds by also removing silent parts and sent each chunk 3 times as HIT. In this way they selected just HIT which had a correspondance. An example of transcription is shown in Figure **??** Qua

0:14:44.190,0:14:45.310	0:00:02.780,0:00:04.640	0:04:37.880,0:04:39.620	0:06:40.669,0:06:41.669	0:12:28.000,0:12:28.000	0:00:03.280,0:00:06.000
pour tofu onto pan	open the bin	Take onion	pick up spatula	pour pasta into container	open fridge
0:14:45.310,0:14:49.540	0:00:04.640,0:00:06.100	0:04:39.620,0:04:48.160	0:06:41.669,0:06:45.250	0:12:33.000,0:12:33.000	0:00:06.000,0:00:09.349
put down tofu container	pick up the bag	Cut onion	stir potatoes	take jar of pesto	take milk
0:14:49.540,0:15:02.690	0:00:06.100,0:00:09.530	0:04:48.160,0:04:49.160	0:06:45.250,0:06:46.250	0 :12:39.000,0:12:39.000	0:00:09.349,0:00:10.910
stir vegetables and tofu	tie the bag	Peel onion	put down spatula	take teaspoon	put milk
0:15:02.690,0:15:06.260	0:00:09.530,0:00:10.610	0:04:49.160,0:04:51.290	0:06:46.250,0:06:50.830	0:12:41.000,0:12:41.000	0:00:10.910,0:00:12.690
put down spatula	tie the bag again	Put peel in bin	turn down hob	pour pesto in container	open cupboard
0:15:06.260,0:15:07.820	0:00:10.610,0:00:14.309	0:04:51.290,0:05:06.350	0:06:50.830,0:06:55.819	0:12:55.000,0:12:55.000	0:00:12.690,0:00:15.089
take tofu container	pick up bag	Peel onion	pick up pan	place pesto bottle on table	take bowl
0:15:07.820,0:15:10.040	0:00:14.309,0:00:17.520	0:05:06.350,0:05:15.200	0:06:55.819,0:06:57.170	0:12:58.000,0:12:58.000	0:00:15.089,0:00:18.080
throw something into the bin	put bag down	Put peel in bin	tip out paneer	take wooden spoon	open drawer

Figure 1.3: Extracts from 6 transcription files in .sbv format

potrei aggiungere ancora qualcosa a pag 7 del paper in cui descrivono come ogni HIT è composta da 10 consecutive narrated phrases... in più 4 annotators per ogni HIT -> Overlap regions come in Figure **??** o magari modificare immagin senza fare veedere $\alpha()$

In the end they collected 39,596 action narrations, corresponding to a narration every 4.9s in the video. These narrations gave them a good starting point for labelling all actions with a rough temporal alignment, obtained from the timestamp of the audio narration with respect to the video, but still were not perfect. Infact:

- The narrations can be incomplete. So only narrated action will be considered in evaluation.

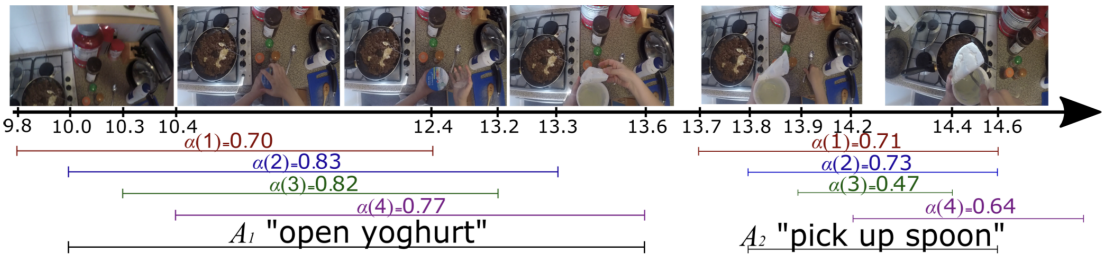

Figure 1.4: Example of annotated action segments for 2 consecutive actions

- The narration can be belated, after the action takes place.
- The narration consists of participants' vocabulary and free language. Similar terms have been grouped in minimally overlapping classes.

It is worth adding few words about verb and noun annotations. Due to the freedom of terms and language, a variety of verbs and nouns have been collected. To reduce the number of them they grouped these into classes with minimal semantic overlapping. More in detail, as regards verbs they tried using automatic tools to cluster them but ended up manually clustering, due to the inefficient results; on the other hand for nouns they semi-automatically cluster them, preprocessing the compound nouns e.g. "pizza cutter" as a subset of the second noun e.g. "cutter" and also manually adjusting the clustering, merging the variety of names used for the same object, e.g. "cup" and "mug". In total they obtained 125 verb classes and 331 noun classes. In Figure **??** we can see some examples of grouped verbs and nouns into classes, while in Figure **??** the authors show the verb classes oredered by frequency of occurrence in action segments, as well as the noun classes ordered by number of annotated bounding boxes.

	ClassNo (Key)	Clustered Words
VERB	0 (take)	take, grab, pick, get, fetch, pick-up, ...
	3 (close)	close, close-off, shut
	12 (turn-on)	turn-on, start, begin, ignite, switch-on, activate, restart, light, ...
NOUN	1 (pan)	pan, frying pan, saucepan, wok, ...
	8 (cupboard)	cupboard, cabinet, locker, flap, cabinet door, cupboard door, closet, ...
	51 (cheese)	cheese slice, mozzarella, paneer, parmesan, ...
	78 (top)	top, counter, counter top, surface, kitchen counter, kitchen top, tiles, ...

Figure 1.5: Sample Verb and Noun Classes

In addition to verb and nouns annotations they also provide active object bounding box annotations. Similarly to verbs and nouns, they use AMT also

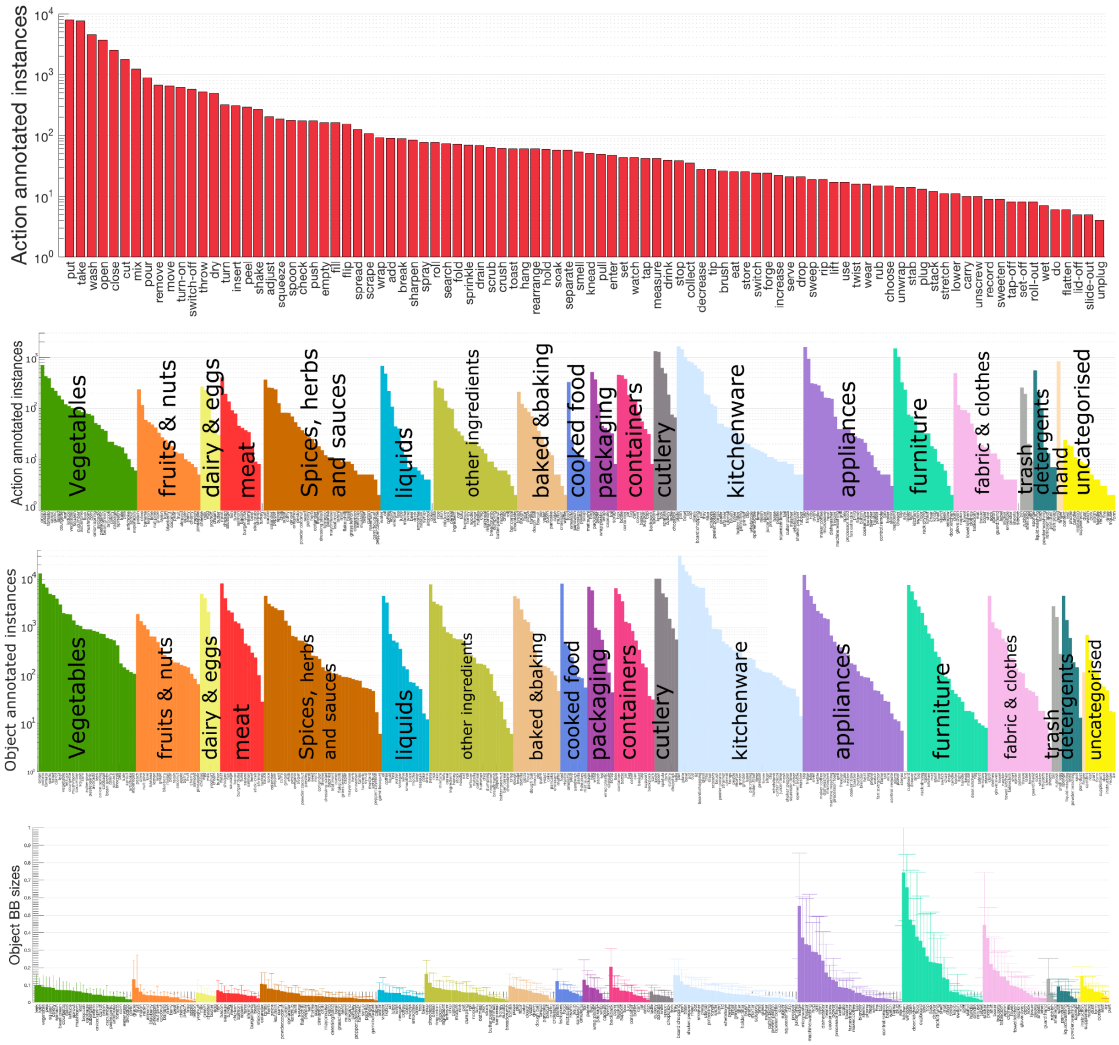

Figure 1.6: From Top: Frequency of verb classes in action segments; Frequency of noun clusters in action segments, by category; Frequency of noun clusters in bounding box annotations, by category; Mean and standard deviation of bounding box, by category

for this task. Where each HIT aims to get an annotation for one object, for the maximum duration of 25s, which corresponds to 50 consecutive frames at 2fps. The annotator can also state that the object is inexistent in at frame *f*. In total they collected 454,255 bounding boxes, some examples are provided in Figure **??**.

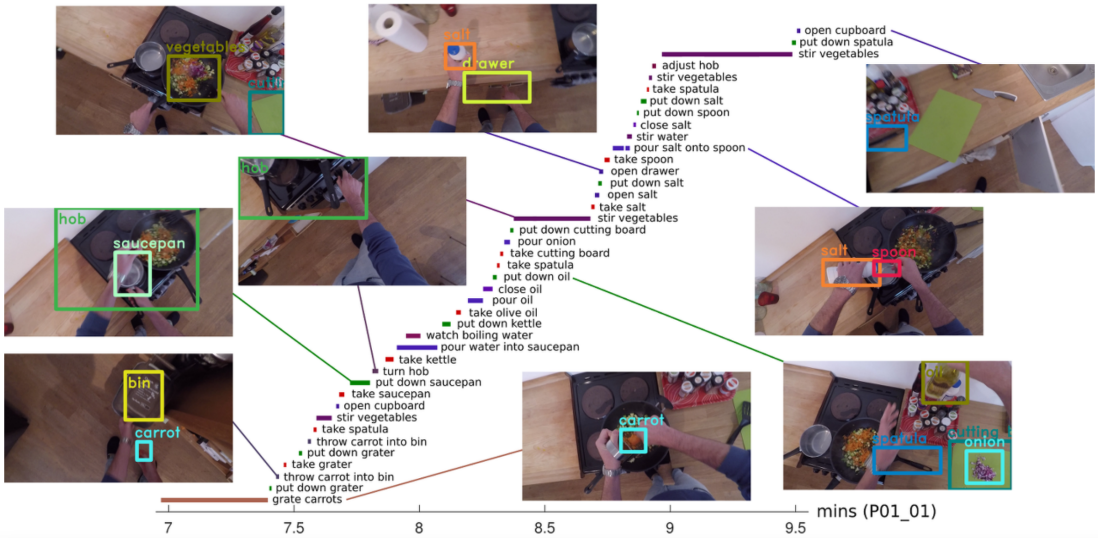

Figure 1.7: Sample consecutive action segments with keyframe object annotations

1.1.4 Benchmarks and Baseline Results

The introduction of a new video datasets implies a variety of potential challenges that were not available before. Some of these are routine understanding, activity recognition and object detection. To spur the beginning the authors define the previous stated three challenges, providing baseline results. Let us see the challenges in more detail.

Action Recognition Challenge

Provided a trimmed action segment, the challenge requires to recognize what action class is performed, detecting the pair of verb and noun classes that compose the action. To participate to the challenge is asked to test the model on both splits[1] and for each test segment report the econfidence scores for each verb and noun class. In Figure **??** is reported a qualitative example of the task.

[1]To test the generalizability to novel environments they structured the test set to have a collection of *seen* and *unseen* kitchens.

- **Seen Kitchens (S1):** in this split each kitchen is seen in both training and testing.
- **Unseen Kitchens (S2):** This divides the participants/kitchens so all sequences of tge same kitchen are either in training or testing.

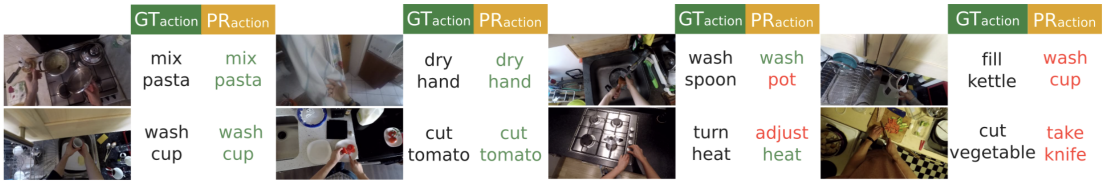

Figure 1.8: Sample qualitative results from the challenge's baseline of the Action Recognition Task

Action Anticipation Challenge

Provided an anticipation time, which is 1s before the action starts, the challenge consists in classifying the future action into its action class composed of the pair of verb and noun classes. To participate to the challenge is asked to test the model on both splits and for each test segment report the econfidence scores for each verb and noun class. In Figure **??** is reported a qualitative example of the task.

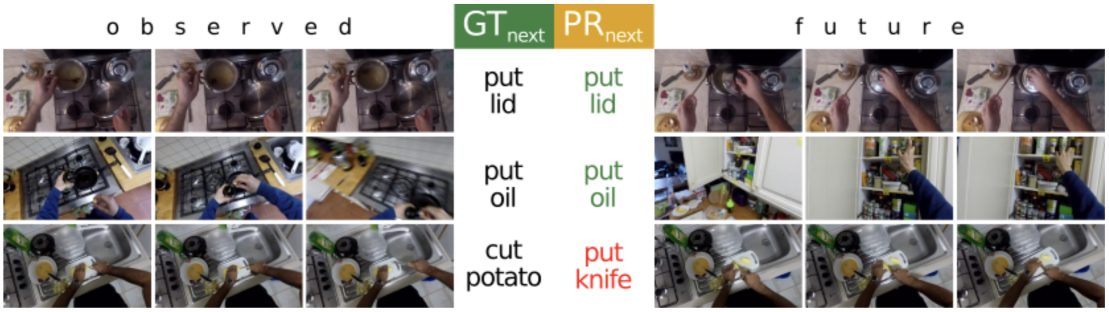

Figure 1.9: Sample qualitative results from the challenge's baseline of the Action Anticipation Task

Object Detection Challenge

In this challenge is required to perform object detectino and localisation. It must be noted that the annotations captured only *active* objects, namely objects involved in the action. To partecipate is required to provide predicted bounding boxes and their confidence scores on both dataset splits. In Figure **??** a qualitative example is reported.

1.1.5 Dataset Release

- Dataset sequences, extracted frames and optical flow are available at:
  `http://dx.doi.org/10.5523/bris.3h91syskeag572hl6tvuovwv4d`

Figure 1.10: Sample qualitative results from the challenge's baseline of the Object Detection Task

- Annotations, challenge leader-board results and updates and news are available at `http://epic-kitchens.github.io`

1.2 EPIC-Kitchens 100

In 2021 with [**EK100**] a new pipeline is introduced to extend EPIC-Kitchens dataset. EPIC-KITCHENS-100 collects 100 hours, 20M frames, 90k actions in 700 variable-length videos, capturing longterm unscripted actions in 45 different environments using headmounted cameras. Due to its novel annotation pipeline, which will be described more in detail later, more complete annotations of fine-grained actions are available, allowing the creation of new challenges such as: action detection[2] ,cross-modal retrieval(e.g.Audio-Based Interaction Recognition) and domain adaptation[3].

1.2.1 Motivation

The introduction of EPIC-KITCHENS has transformed egocentric vision, showcasing the unique potential of first-person views for action recognition and in particular

[2]Action detection involves both recognizing the action and localizing the temporal intervals and spatial regions where the actions occur in a video.

[3]Training on a domain, e.g a specific kitchen, and test on another domain, e.g. a kitchen of a different person.

hand-object interactions. To continue on this previously marked path they decided to enlarge EPIC-KITCHENS, mantaining the *unscripted* and *unedited* object interactions nature. In fact, the unscripted characteristics make the dataset results in a unbalance of data, with novel compositions of actions in new environments, making it a challenging dataset for domain adaptation.

The most important novelty is the new annotation pipeline which allows to obtain denser and more complete actions' annotations in the recorded videos, enabling different task on the same dataset.

1.2.2 Data Collection

The additional videos were obtained by half of the previous participants, 16 persons, half of the 32 previously involved, and 5 new additional subjects. In the end the total participants reached 37 and the different kitchens were 45.

The new request for the subjects was to reecord 2-4 days of their kitchen routine.

1.2.3 Annotation

An overview of the pipeline taken from the paper is reported in Figure **??**.

Narrator

The non-stop audio narration has been replaced with a *pause-and-talk* approach. By pausing the narrator can propose an initial temporal "pointing" but mostly avoids to miss or misspoke some actions due to lack of time. He does not have to narrate past actions while watching future actions, so short and overlapping actions are easier to be annotated.

For this an interface was built for the participants, it can be seen in Figure **??**(a). An important new feature is the possibility to re-record and to delete a narration.

Transcriber

Each narration is first transcribed and then translated in English by a hired translator for correctness and consistency. The transcription process have been facilitated by providing a new transcriber interface showing three images sampled around the time stamp. As a matter of fact, in the old EPIC-Kitchens transcriptor struggled to understand some of the narration wwithout any video context.

Each narration was analyzed by 3 AMT workers using a consensus of 2 or more workers. A transcription was rejected if its Word2Vec **??** embeddings was lower than a threshold of 0.9. In case of consensus failure, the transcription was selected manually.

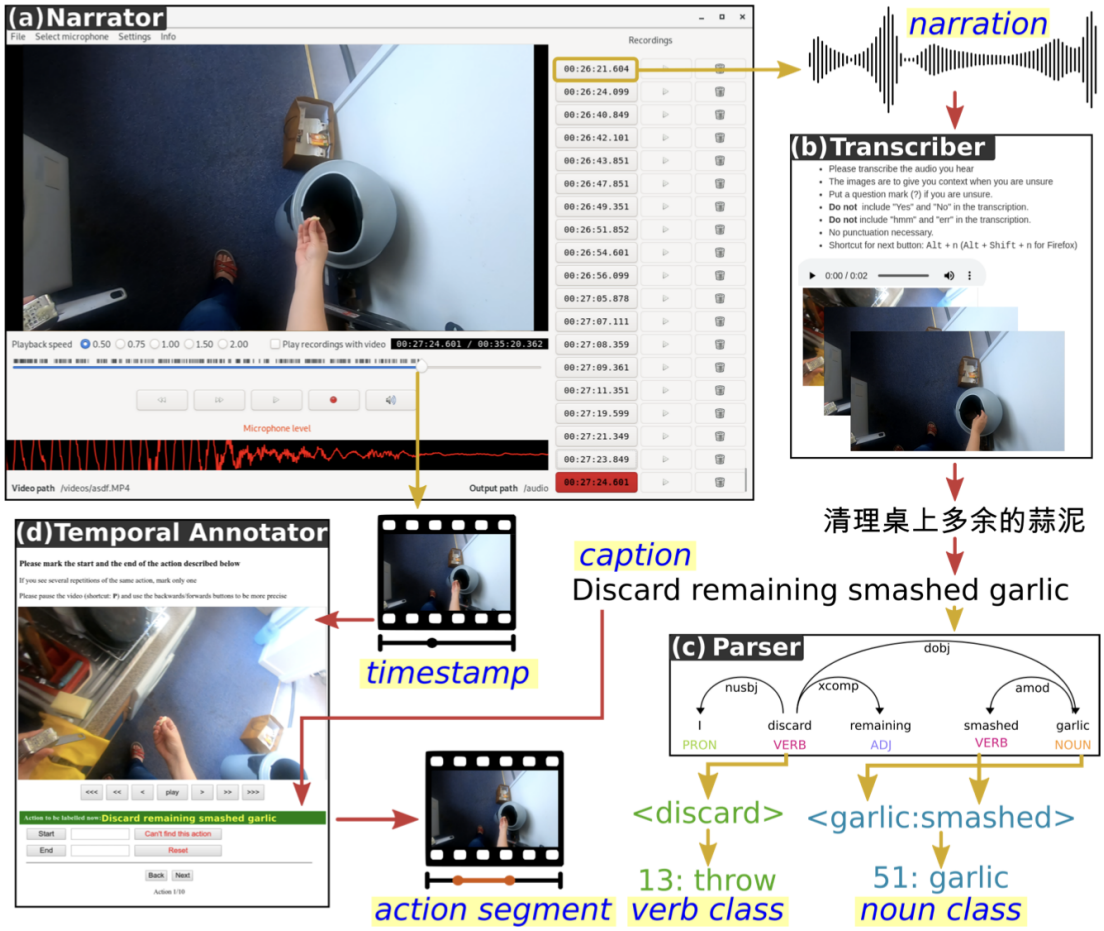

Figure 1.11: Annotation pipeline: **a** narrator, **b** transcriber **c** temporal segment annotator and **d** dependency parser. Red arrows show AMT crowdsourcing of annotations.

Parser

They used spaCy (`https://spacy.io`) to parse the transcriptions into verbs and nouns. Then they manually grouped those into minimally overlapping classes.

Temporal Annotator

They built a AMT interface for the start/end times of action segments(see Figure **??**.d). To improve the quality this time the number of workers were increased from 4 to 5.

1.2.4 Quality Improvements

The attentions cared during the annotation process led to denser and more accurate annotations. We can see the results by comparing the same action and their respective annotation from the two different pipelines in Figure **??**.

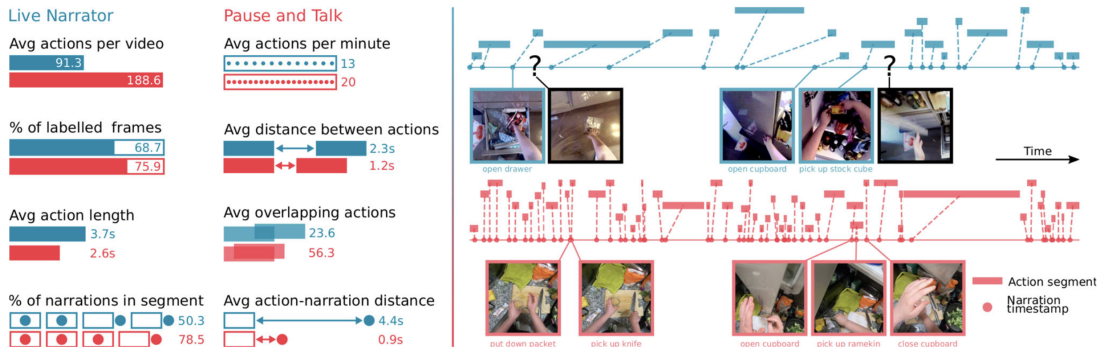

Figure 1.12: Comparing non-stop narrations (blue) to 'pause-and-talk' narrations (red). Right: timestsamps (dots) and segments (bars) for two sample sequences. "pause-and-talk" captures all actions including short ones. Black frames depict missed actions.

1.2.5 Challenges and Baselines

With respect to EPIC-Kitchens, 4 new challenges have been added.Magari non lo metto?

1.3 EPIC-Fields

The necessity of suitable datasets and benchmarks in the unified problem of 3D geometry and video understanding, which has been pushed by Neural Rendering (See Section 1), led to the rise of EPIC Fields. EPIC Fields is a expanded version of EPIC-KITCHENS comprehending 3D camera information. 96% of EPIC-KITCHENS videos were reconstructed, registering 19M frames in 99 hours recorded in 45 kitchens.

EPIC-KITCHENS is suited for studying the unified problem of geometric reconstruction and semantic understanding. As a matter of fact egocentric videos are relevant to mixed and augmented reality applications which are spreading in the last years, and the videos probe dynamic neural reconstruction due to their length (up to one hour) and to their dynamic nature.

Anyway obtaining camera information from EPIC-KITCHENS is difficult due to the complexity of its videos. Removing this step the authors try to ease the research in marrying 3D geometry to video understanding.

In conclusion they made two contributions:

- Intelligent Subsampling of frames for SfM algorithms
- Introduce a set of benchmark tasks:
  - dynamic novel view synthesis: reconstruct the same scene from a different point of view.
  - identifying independently from the camera moving objects
  - segmenting independently from the camera moving objects
  - video object segmentation.

1.3.1 Data

Some of past egocentric datasets [**visor35**, **visor5**] contain static 3D scans of the environment, separately reconstructed from the actions. This additional step is an additional expense both in time and money, since the reconstructions are done with some dedicated costly hardware. In this work they provide a pipeline to extract the geometric reconstruction of the scene by just processing the egocentric video. EPIC Fields extends EPIC-KITCHENS(See Section **??**) to include camera pose information. For each frame camera extrinsics and intrinsics parameterrs are provided, which enable tasks like 3D reconstruction. In total the succesfully processed 671 videos resulting in 18,790,333 registered video frames with estimated camera poses.

Motivation.

The 3D reconstruction could help recognizing different actions. Some actions could be located in the same 3D spot, e.g washing the dishes at the sink. Also the construction of tihs dataset could enable studying the relevance of 3D egocentric trajectories to actions(for anticipation),objects(for understanding object state changes) and hand-object understanding.

Collection

Since EPIC-KITCHENS did not collect videos with 3D reconstruction in mind, its videos are difficult to reconstruct. In fact Structure from Motion algorithms take as assumption that the recorded scene is *static*, meaning that each object will always have the same posiiton in 3D. However kitchen's activities involve the movement

of objects like ingredients or utensils, and above all the presence of the operating hands.

Some other difficulties are introduced by:

- the **length of videos**, which on average last 9 mins
- the **skewed distribution of viewpoints**: the time spent in different part of the scene is different. In particular we have alternating phases of small motion around hot-spots,e.g. washing dishes, and of fast motions, like taking something to finish some task.

The solutions to these problems were given by:

- **Intelligent subsampling** of video frames.
- Using **SfM** for reconstructing the filtered frames.
- **Registering remaining frames** to the reconstruction.

Filtering

The aim of this step is to reduce the number of frames while keeping enough overlapping viewpoints for accurate reconstruction while diminishing the viewpoint skew. Overlap is measured by estimating homographies on matched SIFT features. Given a homography H, we define visual overlap r as the fraction of image area covered by the quadrilateral formed by warping the image corners by H. Windows are formed greedily, finding runs of frames (i+1,...,i+k) with overlap $r \geq 0.9$ to the first frame i. Filtering discards about 81.8 % of frames

Sparse reconstruction

Once filtered, frames are fed to COLMAP(Its functioning is reported in Section **??**).

Dense reconstruction, automated verification, and restart

The remaining frames are fed to COLMAP with initial reconstruction. The final reconstruciton is accepted if over 70% of frames are registered succesfully. In the end 631 videos were obtained.

In case of failure the threshold r is increase,e.g. r$\geq$0.95. This usually results in doubling the frames, but increasing success rate to 96%.

1.3.2 Benchmarks, Experiments and Results

The authors defined three new benchmarks to explore the combination of 3D and video understanding.

New-View Synthesis(NVS)

Given a reconstruction based on a subsample of frames, the goal is to predict new video frames based on their timestamps and camera parameters. The quality of the reconstruction is evaluated as proposed in [2],measuring Peak Signal-to-Noise Ratio (PSNR) of the reconstructed frames compared to the real ones, making the lack of a 3D ground-truth irrelevant.

Video and Frame Selection. Due to the computational expensive cost of Neural Reconstruction, they provided a benchmark of limited selection of videos, namely 50 for a duration of 14.7 hours and 2.86M registered frames.

Frame selection instead is needed to divided the data in train and evaluation splits. The evaluation frames were divided into three tiers of difficulty:

- **In-Action (Hard):** frames belonging to an annotated action segment. During an action it is likely that object in the scene are moved making it difficult to reconstruct.

- **Out-of-Action:** frames NOT belonging to an annotated action segment. These frames can be further divided in:

  - **Easy**: Frames for which exists a neighbouring frame in the training set. The temporal proximity should ease the process of reconstruction.
  - **Medium**: Frames for which do not exists a neighbouring frame in the training set.

Benchmark methods. Three different neural rendering techniques were used to illustrate their possibilities and limits in such challenging scenarios like EPIC Fields. The methods used were:

- **NeuralDiff [27]:** consists of three different NeRFs, each one tailored to a part of the scene: static background, moving foreground and the actor. See Section 1 for more details.

- **NeRF-W [nerfw]:** extend NeRF abilities by learning a low latent space that can modulate scene appearance and geometry. As a results it can separates static and transient components.

- **T-NeRF+ [Tnerf]:** time conditioned NeRF at which was added another NeRF to model the static background.

In Table **??** I report the authors' resulting experiments, while in Figure **??** we can see the comparison of the output of the three methods.

Method	Easy	Medium	Hard		
			All	BG	FG
NeRF-W [**nerfw**]	21.13	19.3	17.93	18.99	13.54
T-NeRF+ [**Tnerf**]	21.58	19.81	18.44	19.73	13.74
NeuralDiff [27]	22.14	19.88	18.36	19.54	13.37

Table 1.2: Dynamic New View Synthesis. Comparison of different neural rendering methods on varying difficult frames. The values reported corresponds to PSNR considering all pixels in each test frame.

Unsupervised Dynamic Object Segmentation(UDOS)

In Unsupervised Dynamic Object Segmentation(UDOS) the objective is to find those regions in each frames that correspond to dynamic objects. The lack of a 3D ground-truth make 2D segmentation accuracy the only way to assess the model. In particular mean average precision (mAP) was used, as proposed in [27].

Video and Frame selection. The used videos were the same of NVS, with the difference that only In-Action frames where considered, with VISOR annotations as ground-truth. Actually VISOR annotations were processed in the following way: the original masks were converted into foreground-background masks in three different ways, depending on the type of objects present.

- **Dynamic objects only** setting:a dynamic object is an object that is currently being moved by visible hands.

- **Dynamic and semi-static objects** setting: objects that moved, not necessarily in the current frame, are semi-static objects.

- **Dynamic and semi-static excluding body parts** setting: active hands are excluded, as some methods overfit to predicting hands solely as dynamic objects, ignoring other moving objects.

As Baseline methods the authors used 4 methods: three based on 3D neural rendering techniques(NeRF-W,T-NeRF,NeuralDiff) and one based on 2D optical flow(Motion

Grouping(MG) [**MG**]. The results are shown in Figure **??** and Table **??**.It is worth noting how 3D methods are better discovering semi-static objects. However none of the 3D methods explicitly consider motion. and this can be seen as MG performs better on purely dynamic motion, due to the input being the optical flow.

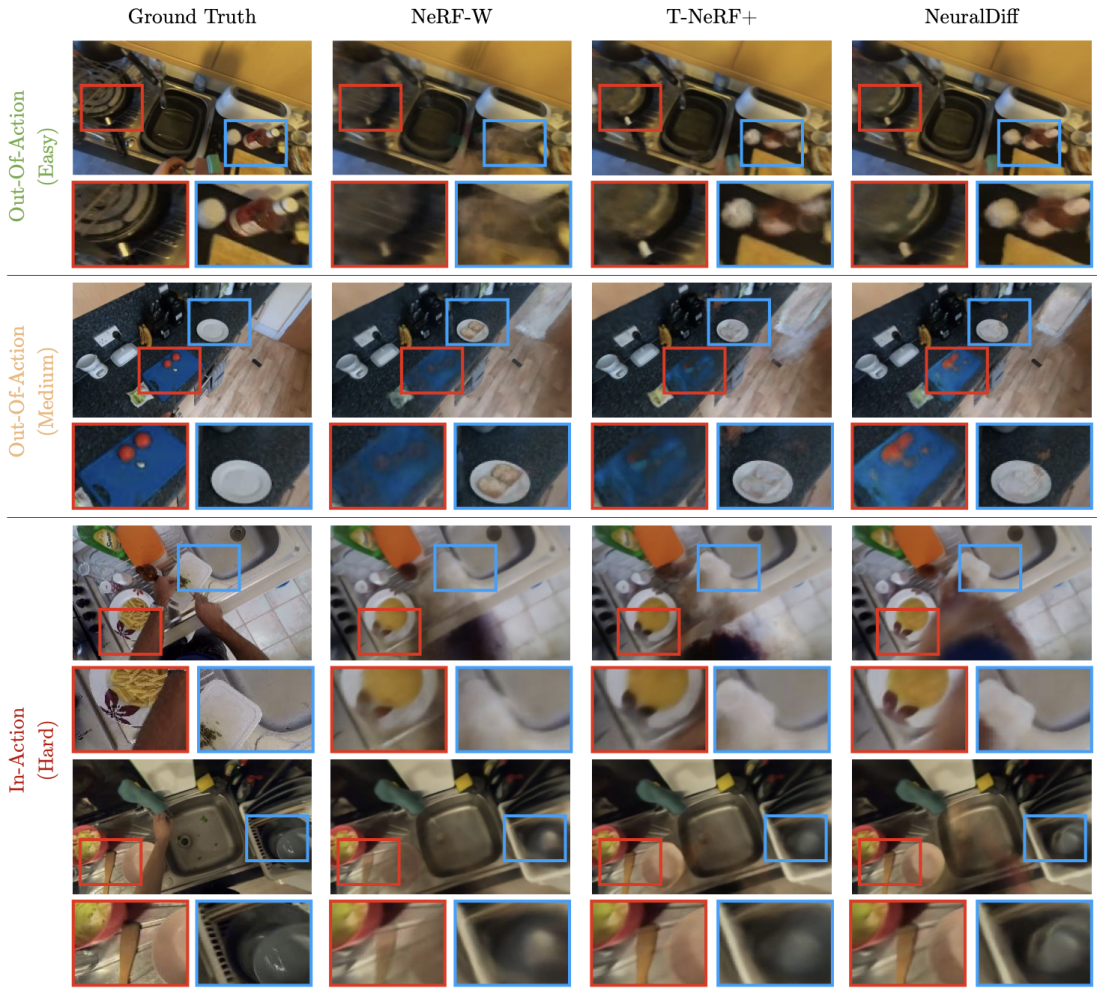

Figure 1.13: Dynamic New View Synthesis. I report an example of the output for the three different methods used: NeRF-W, T-NeRF+ and NeuralDiff. We can see how the initial labelling of difficulty for frames was actually accurate as the reconstructions struggle with Hard frames.

Semi-Supervised Video Object Segmentation(VOS)

Semi-Supervised Video Object Segmentation is a standard video understanding task which consists in propagating some given masks for one or more objects in a reference frame to the subsequent ones. Usually this task is performed by 2D models but here the authors show how integrating the third dimension could be beneficial. The idea is to project the 2D mask in 3D, fixing its position in the 3D scene and reproject it depending on the new camera position.

Two baselines were provided, a 2D and a 3D one. In Figure **??** we can see the

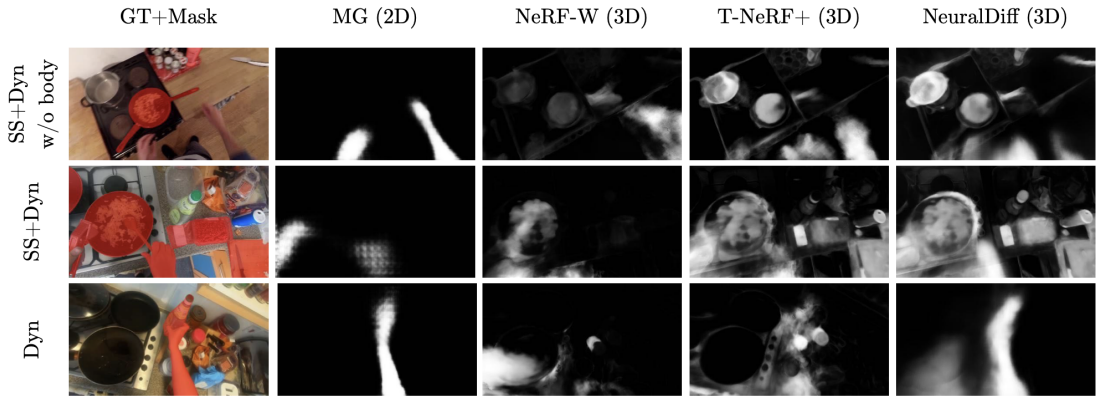

Figure 1.14: UDOS. Here is reported the comparison of the different methods' output. The 2D based perform very good on dynamic objects, while 3D methods struggle a bit but can detect even semi-static objects.

Method	3D	SS+Dyn	SS+Dyn (w/o body)	Dynamic
MG [MG]	-	60.19	21.65	69.26
NeRF-W [nerfw]	✓	37.26	22.96	27.41
T-NeRF+ [Tnerf]	✓	54.23	31.23	42.68
NeuralDiff [27]	✓	62.30	31.11	55.10

Table 1.3: UDOS. Here I reported the author results for UDOS. The values visible corresponds to the mAP on segmenting semi-static(SS) and dynamic(Dyn) components of the scene.

results and how the intuition previously described led to a good improvement.

1.4 VISOR

VISOR [**visor**] is an extension of the EPIC-KITCHEN dataset introducing pixel annotations and a benchmark suite for segmenting hands and active objects. In particular segmentation annotations are provided in a *sparse* way, meaning that not every frame is segmented at pixel level.

The authors proposed a pipeline to obtain the annotations. Namely it consists of: (i) identifying *active* objects that are of relevance to the current action;(ii)annotating pixel-level segments via an AI-powered interface;(iii) relating objects spatially and temporally for short-term consistency. Some examples of annotations are reported in Figure **??**.

ACTIVE OBJECT: each object that is involved in the current action. It does not

Figure 1.15: VOS. Here is reported the comparison of the two different methods. The 3D method is clearly beter having as output something really close to the groundtruth. The 2D method instead is performing poorly.

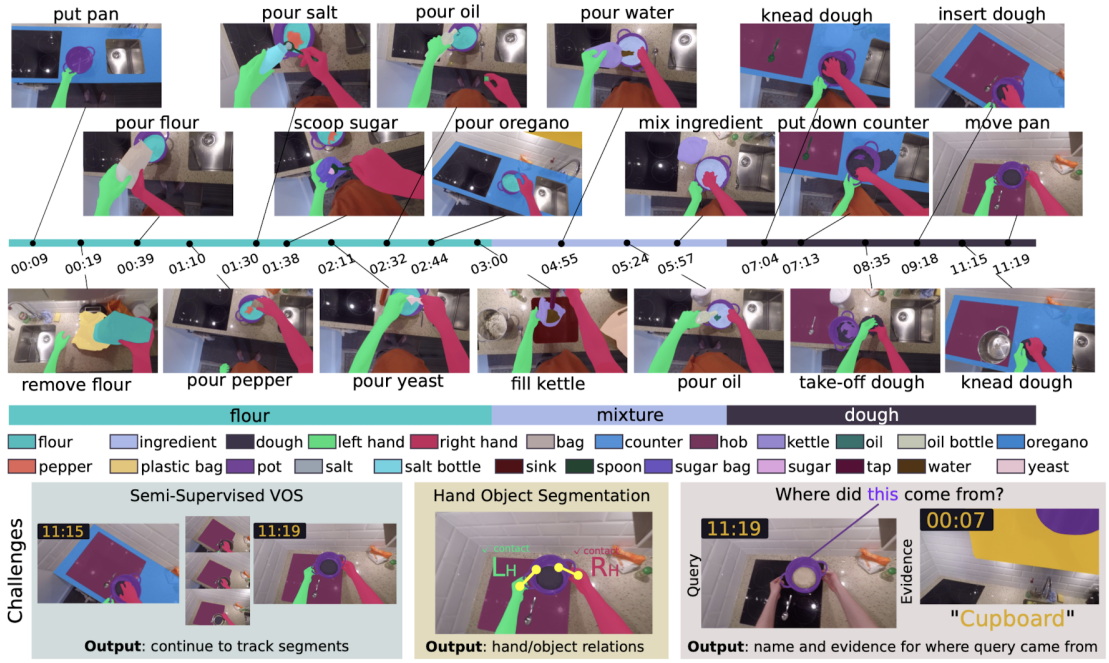

Figure 1.16: VISOR Annotations and Benchmarks. Sparse annotations of the P06-03 scene, where flour becomes dough in the end.Each color represents a different object. In the bottom part the three proposed challenges are reported for the current scene. Namely: Semi-Supervised Video Object Semgentation(VOS), Hand Object Segmentation(HOS) and Where Did This Come From(WDTCF).

mean it is dynamic, since as shown in Figure **??** the table top is usually segmented as active. Non va bene perchè gli oggeti che vengono segmentati sono solo quelli rilevanti all azione descritta, quindi se sposto un bicchiere mentre sto pelando le carote non viene segmentato il bicchiere -> non va bene per noi. Oppure oggetti attivi possono essere il lavandino, il tosta pane che sono però statici. LOC SCIVO QUA O DA QUALCHE ALTRA PARTE?

1.5 Other Egocentric Datasets

1.5.1 Ego4D

Part II

Da Sistemare

1.6 Metrics

CHIEDERE COME MOTIVARE LA SCELTA DELLE NOSTRE METRICHE
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 [27]. Namely these are:

- **PSNR**
- **AP**

1.6.1 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.

1.6.2 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:

$$Precision = \frac{TP}{TP + FP} \tag{1.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{1.2}$$

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 1.8. 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.

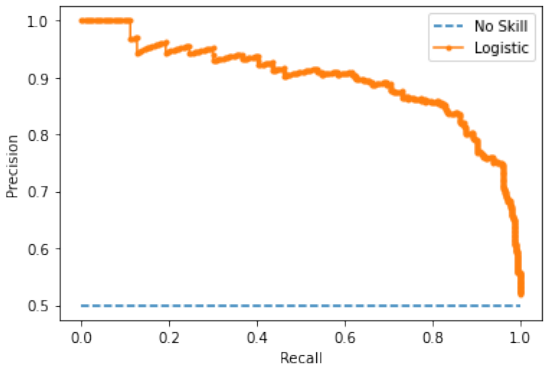

Figure 1.17: Example of Precision-recall curve.We can see how the bottom line model represents the worst a model can perform, e.g. predict every sample as it is coming from the same class,if the dataset is balanced. A better model would *tend* to the upper-right corner, which instead represents the best possible model, a model that have maximum precision and recall.

1.7 Sampling

Per il sampling abbiamo prima seguito Epic fields. QUindi abbiamo variato soglia per ottener circa 5k frames,seguendo indicazioni di EpicFields. Dopo di

che, siccome NeuralDiff è molto lento, non possiamo dargli un sacco di frames → ulteriore sampling. Per questo sampling abbiamo fatto delle prove per cercare di ottenere un metodo che consentisse sia di ridurre i frame ma di mantenere una ricostruzionoe adeguata. Perciò abbiamo provato diversi metodi:

- Uniform
- Altro
- Intelligent

estraendo per ognuno circa 1k frames. Il test e val set sono uguali per tutti e 3.

Scene	**P01-01**	**P03-04**	**P04-01**	**P05-01**	**P06-03**	**P08-01**	**P09-02**	**P13-03**	**P16-01**	**P21-01**	Avg.Time
Succesful Reconstruction	✓	✓	✓	×	×	×	✓	×	✓	✓	-
Reconstructed Frames	693	831	873	-	-	-	817	-	936	725	-
Feature Extractor	0:00:02	0:00:05	00:00:02	-	-	-	0:00:02	×	0:00:02	0:00:02	
Exhaustive Matcher	0:00:18	0:00:19	00:00:23	-	-	-	0:00:21	×	0:00:31	0:00:14	
Mapper	0:25:58	0:21:03	00:46:37	-	-	-	0:31:53	×	1:02:12	0:46:34	
Image Undistorter	0:00:00	0:00:01	0:00:00	-	-	-	0:00:01	×	0:00:00	0:00:01	
Patch Match Stereo	0:20:30	0:24:40	0:25:21	-	-	-	0:23:37	×	0:27:22	0:21:18	
Stereo Fusion	0:00:12	0:00:12	0:00:17	-	-	-	0:00:15	×	0:00:21	0:00:11	
Total Time				-	-	-		-			

Table 1.4: COLMAP execution times for each scene.

Appendix A

Galileo

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

$\mathcal{O}\left(n \log n\right)$
`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] URL: https://justusthies.github.io/posts/neuralrenderingtutorial_cvpr/#:~:text=Neural%20rendering%20is%20a%20new,%2C%20appearance%2C%20and%20semantic%20structure (cit. on p. 3).

[2] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020. arXiv: 2003.08934 [cs.CV] (cit. on p. 3).

[3] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. *Local Implicit Grid Representations for 3D Scenes*. 2020. arXiv: 2003.08981 [cs.CV] (cit. on p. 4).

[4] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. *DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation*. 2019. arXiv: 1901.05103 [cs.CV] (cit. on p. 4).

[5] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. *Local Deep Implicit Functions for 3D Shape*. 2020. arXiv: 1912.06126 [cs.CV] (cit. on p. 4).

[6] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. *Occupancy Networks: Learning 3D Reconstruction in Function Space*. 2019. arXiv: 1812.03828 [cs.CV] (cit. on p. 4).

[7] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. *Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision*. 2020. arXiv: 1912.07372 [cs.CV] (cit. on p. 5).

[8] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. *Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations*. 2020. arXiv: 1906.01618 [cs.CV] (cit. on p. 5).

[9] Marc Levoy and Pat Hanrahan. «Light field rendering». In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH96. ACM, Aug. 1996. DOI: 10.1145/237170.237199. URL: http://dx.doi.org/10.1145/237170.237199 (cit. on p. 5).

[10] Steven Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael Cohen. «The Lumigraph». In: *Proc. of SIGGRAPH 96* 96 (Aug. 2001). DOI: 10.1145/237170.237200 (cit. on p. 5).

[11] Michael Waechter, Nils Moehrle, and Michael Goesele. «Let There Be Color! Large-Scale Texturing of 3D Reconstructions». In: *European Conference on Computer Vision*. 2014. URL: https://api.semanticscholar.org/CorpusID:6085476 (cit. on p. 5).

[12] Chris Buehler, Michael Bosse, Leonard McMillan, Steven J. Gortler, and Michael F. Cohen. «Unstructured lumigraph rendering». In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001). URL: https://api.semanticscholar.org/CorpusID:215780580 (cit. on p. 5).

[13] Wenzheng Chen, Jun Gao, Huan Ling, Edward J. Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. *Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer*. 2019. arXiv: 1908.01210 [cs.CV] (cit. on p. 5).

[14] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. *Unsupervised Training for 3D Morphable Model Regression*. 2018. arXiv: 1806.06098 [cs.CV] (cit. on p. 5).

[15] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. *Soft Rasterizer: Differentiable Rendering for Unsupervised Single-View Mesh Reconstruction*. 2019. arXiv: 1901.05567 [cs.CV] (cit. on p. 5).

[16] K.N. Kutulakos and S.M. Seitz. «A theory of shape by space carving». In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 1. 1999, 307–314 vol.1. DOI: 10.1109/ICCV.1999.791235 (cit. on p. 5).

[17] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. *DeepView: View Synthesis with Learned Gradient Descent*. 2019. arXiv: 1906.07316 [cs.CV] (cit. on p. 5).

[18] Philipp Henzler, Volker Rasche, Timo Ropinski, and Tobias Ritschel. *Single-image Tomography: 3D Volumes from 2D Cranial X-Rays*. 2018. arXiv: 1710.04867 [cs.GR] (cit. on p. 5).

[19] Abhishek Kar, Christian Häne, and Jitendra Malik. «Learning a Multi-View Stereo Machine». In: *ArXiv* abs/1708.05375 (2017). URL: https://api.semanticscholar.org/CorpusID:19285959 (cit. on p. 5).

[20] James T. Kajiya and Brian Von Herzen. «Ray tracing volume densities». In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques* (1984). URL: https://api.semanticscholar.org/CorpusID:6722621 (cit. on p. 7).

[21] N. Max. «Optical models for direct volume rendering». In: *IEEE Transactions on Visualization and Computer Graphics* 1.2 (1995), pp. 99–108. DOI: 10.1109/2945.468400 (cit. on p. 7).

[22] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. *On the Spectral Bias of Neural Networks*. 2019. arXiv: 1806.08734 [stat.ML] (cit. on p. 8).

[23] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. *DeepVoxels: Learning Persistent 3D Feature Embeddings*. 2019. arXiv: 1812.01024 [cs.CV] (cit. on p. 10).

[24] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. *Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines*. 2019. arXiv: 1905.00889 [cs.CV] (cit. on pp. 10, 11).

[25] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. «Neural volumes: learning dynamic renderable volumes from images». In: *ACM Transactions on Graphics* 38.4 (July 2019), pp. 1–14. ISSN: 1557-7368. DOI: 10.1145/3306346.3323020. URL: http://dx.doi.org/10.1145/3306346.3323020 (cit. on p. 10).

[26] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. *Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations*. 2020. arXiv: 1906.01618 [cs.CV] (cit. on p. 10).

[27] 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 p. 15).