Reconstructing and grounding narrated instructional videos in 3D

Dimitri Zhukov, Ignacio Rocco, Ivan Laptev, Josef Sivic, Johannes L Schönberger, Bugra Tekin, Marc Pollefeys

▶ **To cite this version:**

Dimitri Zhukov, Ignacio Rocco, Ivan Laptev, Josef Sivic, Johannes L Schönberger, et al.. Reconstructing and grounding narrated instructional videos in 3D. 2021. hal-03571657

HAL Id: hal-03571657
https://hal.science/hal-03571657

Preprint submitted on 14 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reconstructing and grounding narrated instructional videos in 3D

Dimitri Zhukov[1*] Ignacio Rocco[1,2*] Ivan Laptev[1] Josef Sivic[3]
Johannes L. Schönberger[4] Bugra Tekin[4] Marc Pollefeys[4,5]

Abstract

Narrated instructional videos often show and describe manipulations of similar objects, e.g., repairing a particular model of a car or laptop. In this work we aim to reconstruct such objects and to localize associated narrations in 3D. Contrary to the standard scenario of instance-level 3D reconstruction, where identical *objects or scenes are present in all views, objects in different instructional videos may have large appearance variations given varying conditions and versions of the same product. Narrations may also have large variation in natural language expressions. We address these challenges by three contributions. First, we propose an approach for correspondence estimation combining learnt local features and dense flow. Second, we design a two-step* divide and conquer *reconstruction approach where the initial 3D reconstructions of individual videos are combined into a 3D alignment graph. Finally, we propose an unsupervised approach to ground natural language in obtained 3D reconstructions. We demonstrate the effectiveness of our approach for the domain of car maintenance. Given raw instructional videos and no manual supervision, our method successfully reconstructs engines of different car models and associates textual descriptions with corresponding objects in 3D.*

Figure 1: **Illustration of the proposed method.** Given a set of instructional videos showing the same object (e.g. same car model), we propose a two-step 3D reconstruction approach where, first, each video is independently reconstructed (R) and, second, reconstructions are aligned (A) in a single model. Using the reconstructed 3D model together with the narrations from each individual video and no manual supervision, we learn a language grounding model (G) that can determine the corresponding 3D region for a given text query such as "air filter".

1. Introduction

Imagine you need to perform a task on a product you own, such as replacing the drum unit in your laser printer or re-filling the coolant fluid tank in your car, but you don't know how to proceed. In the past few years, instructional videos have become a popular internet resource for learning how to perform such kind of tasks. However, finding the answer to this particular problem would require searching for the appropriate video, identifying the key steps, and finally, putting it into practice by performing the right sequence of actions involving the correct object parts at the right locations. Wouldn't it be great to have a virtual assistant guide you through the steps in augmented reality showing you *which* tasks to perform and *where* to perform them? In order to produce such an assistant, a method for relating natural language to the actual 3D objects in the world is needed. In this work we show that it is possible to obtain 3D reconstructions and to learn associations between natural language and corresponding 3D locations (3D language grounding) using raw instructional videos and no manual annotations.

In the past, instructional videos have proven to be a

*Equal contribution
[1]Inria, Ecole normale suprieure, CNRS, PSL Research University, Paris, France
[2]Now at Facebook AI Research
[3]CIIRC – Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague
[4]Microsoft
[5]ETH Zurich

useful source of data for learning to temporally localize a set of actions in a video strip, given the set of action labels (i.e. weakly-supervised temporal action localization) [26, 3, 18, 35, 36, 40, 24, 56, 11]. This is useful for downstream tasks such as video summarization or video retrieval. However, with the advent of smartphones and headsets equipped with 3D sensors the problem of spatial action localization in 3D gains relevance [8, 27, 33, 22]. While raw video data is useful to learn visual representations of actions, it is difficult to directly learn about 3D structure from 2D frame data. Therefore, in order to solve the problem of spatial action localization in 3D, a suitable 3D spatial representation of the object and the scene are necessary. However, producing a 3D reconstruction from a set of narrated instructional videos presents several challenges.

In recent years, significant progress has been made in the field of 3D reconstruction. Structure-from-motion (SfM) pipelines [39, 32, 51, 44, 31] leverage robust local features for obtaining correspondences between images despite large illumination and viewpoint changes. This has allowed to perform 3D reconstruction from unstructured photo-collections from the web, where images may be taken months or years apart [43, 14, 1]. However, the task of reconstructing a set of narrated instructional videos poses some additional challenges. First, besides the large illumination and viewpoint changes, these videos may contain significant appearance changes. This is due to the fact that, while the objects that appear in different videos belong to the same brand and model, they might present variations due to different options or sub-versions of the same product. Therefore, obtaining correspondences between such videos is an intermediate problem between traditional instance-level matching (the *same* physical object is imaged) [25, 20, 43] and category-level matching (semantically related objects are imaged, such as two different species of dogs) [16, 37, 38, 30]. Despite the robustness of the local descriptors used in current SfM pipelines, these are not designed to handle such strong appearance changes and therefore are prone to failure. As a second challenge, because the computational cost of 3D reconstruction grows quadratically with the number of total frames, jointly reconstructing a set of many videos may be unfeasible, or may impose a strong limit in the number of frames to be sampled from each video.

In this work we address the above difficulties and propose a method that is (i) capable of finding correspondences across instructional videos despite the large appearance changes, and (ii) reduces the global computational complexity of 3D reconstruction by reconstructing each video independently and then combining the results. Given the obtained 3D reconstructions and corresponding video narrations, we next learn 3D language grounding. Our approach to 3D language grounding leverages both narrations and visual information in instructional videos to associate the input text to different parts of the 3D scene, and doesn't require manual supervision (see Fig. 1).

Contributions. In summary, our contributions are the following. **I.** We develop an approach for correspondence estimation that combines the strengths of local features and dense flow in order to obtain reliable matches across instructional videos. **II.** We propose a two-step 3D reconstruction approach that reduces the overall computational complexity by reconstructing each video independently, and then combining the results into a 3D alignment graph as a proxy to a single aligned 3D reconstruction. We evaluate obtained reconstructions on the task of keypoint transfer. **III.** We propose an unsupervised method, that makes use of the obtained 3D models and narrations in instructional videos to learn language grounding in 3D.

2. Related work

3D reconstruction. The standard approach to obtain a 3D reconstruction from a set of images is to employ an incremental or global structure-from-motion pipeline such as COLMAP [39], Theia [44], VisualSfM [51] or OpenMVG [32]. In all cases, the first step is to find correspondences between the set of input images. This is typically done using local features such as SIFT [25], which are robust to moderate changes in illumination and viewpoint present in instance-level 3D reconstruction. In this work, we propose a different 3D reconstruction approach targeted to the reconstruction of narrated instructional videos, which have the additional challenge of large appearance changes due to differences in sub-versions of the same object type or due to different conditions of conservation (wear, cleanliness, etc.)

Learning from instructional videos. Learning from instructional videos has been recently addressed in the context of problems such as action discovery [3, 36, 40, 24, 11] and localization [18, 35, 54, 56], modeling of object states [4] and visual reference resolution [19, 17]. A particular attention has been drawn to the *narrated* instructional videos, *i.e*. instructional videos, accompanied by comments in the form of natural language. It was shown, that narrations in such videos generally describe the visual content and can be used as an additional guidance for learning the visual models [26, 3, 56], or for joint modeling of video and language [29, 28, 55]. In this work, we make use of narration for object grounding in 3D, assuming that object mentions in narration coincide with their appearance on the screen.

Related to our work, Damen et al., [9] consider the problem of task-relevant object discovery from videos. Unlike [9] we don't aim for a discovery of objects, and only

attempt to localize objects in the 3D scene. On the other hand we consider a more general setup: our videos depict different object instances in uncontrolled scenarios and are sourced from a mix of egocentric and third-person cameras.

Visual language grounding. Visual language grounding requires associations between the text the most relevant parts of an image. A standard approach to this task involves learning a joint embedding of language and visual data and comparing the similarity between the text and image regions in the joint embedding space [21, 48, 49, 47, 6]. Most works focus on learning to ground language in images in supervised settings, where the regions of images, relevant to the input text are provided at training time. On the contrary, [53, 12, 52] consider weakly-supervised scenario where only the correspondences between text and images are provided during training, while the exact regions, where the text is grounded, are unknown.

Closer to our work, Russell *et al*. [7] attempt to localize text from Wikipedia articles in a 3D scene. This work focuses on museum scenes and the objects, such as paintings and frescos. This is done in two steps. First, an object caption is used as a query to search for related images with Google image search. Retrieved images are then registered to the 3D scene and their extents are used as a clue for object localization in 3D. The authors of this work leverage the *photographer's bias*, *i.e*. the fact that objects of interest are usually perfectly framed by the extents of the image. Our solution for the 3D text grounding is based on a similar observation. However, we focus on instructional videos and unlike [7], we use the same video data both for reconstruction and for language grounding, and don't rely on query expansion via external image search.

3. 3D reconstruction of instructional videos

In this section we present our approach for reconstructing narrated instructional videos in 3D. In order to address the limitations of the standard instance-level 3D reconstruction pipeline (i.e. limited robustness to appearance changes and large computational complexity) we propose an approach based on the *divide and conquer* strategy, where each video is reconstructed independently and then the resulting 3D models are combined into a 3D alignment graph that allows to obtain a final consistent model. In this way we have two advantages over the joint reconstruction approach. First, only a few correspondences are needed *across* different videos, in order to estimate the similarity transformation that aligns the 3D models reconstructed from each video. Second, the computational complexity of the 3D reconstruction is reduced from $O(n^2N^2)$ to $O(N^2)$, where N is the number of frames sampled from each video, and n is the number of videos. In the following sections, we describe in detail our approach for reconstructing instructional videos in 3D. An overview of the approach is presented in Fig. 2.

3.1. Per-video reconstruction

In order to avoid the large computational complexity and matching challenges of joint 3D reconstruction, we reconstruct each video independently. For each video, we proceed in the following way. First, N video frames $\{v_i\}_{i=1,\dots,N}$ are randomly sampled. Then, CNN-based local features $(f_i, p_i) = L(v_i)$ are extracted for every frame using the feature transform L, where $f_i \in \mathbb{R}^{d_L \times N_L}$ is the set of d_L-dimensional feature descriptors, $p_i \in \mathbb{R}^{2 \times N_L}$ is the set of 2D key-point positions, and N_L is the number of extracted local features. Then, exhaustive matching is performed between the features of every video frame v_i. Only matches that are mutually nearest-neighbors are retained. Finally, these matches are input into an SfM pipeline to produce a 3D model represented as a cloud of 3D points $X = \{(X_i, Y_i, Z_i)\}_{i=1,\dots,N_p}$ for the given set of video frames[1]. An illustration of this approach is presented in Fig. 2a.

3.2. Formation of frame-pairs across videos

Once every video is independently reconstructed, the goal is to align them to a common frame of reference, which involves estimating a 3D similarity transformation. Then, given a pair of videos (v^A, v^B), the first step involves estimating a candidate set of matching video frames in the form of $M = \{(v_i^A, v_j^B)\}$, where i and j are the frame indices of matching frames of v^A and v^B respectively. In order to achieve this, we follow the approach that is typically used for image retrieval. This consists of, first, extracting image-level descriptors $d_i^A = D(v_i^A)$ and $d_j^B = D(v_j^B)$ using the image-level description function D for every frame v_i^A in v^A and v_j^B in v^B, where $d_i^A, d_j^B \in \mathbb{R}^{d_I}$ are d_I-dimensional descriptors. Then, the top N_M matches between videos v^A and v^B are selected by nearest-neighbor search between the sets of image level descriptors $\{d_i^A\}$ and $\{d_j^B\}$. An illustration if this approach is presented in Fig. 2b.

3.3. Establishing 2D correspondences between videos

Once candidate matching frames $\{(v_i^A, v_j^B)\}$ between a pair of videos (v^A, v^B) are established, the next step consists of robustly estimating 2D-2D correspondences between them. In order to handle the large appearance differences that may be present in these two frames, we propose to use a combination of learnt local features and dense flow to robustly obtain matches that are simultaneously well localized and compliant with a global transformation. Given a

[1] In some cases, several 3D models will be produced due to the fact that videos might have edits, and show completely different scenes at different times. Small models are discarded.

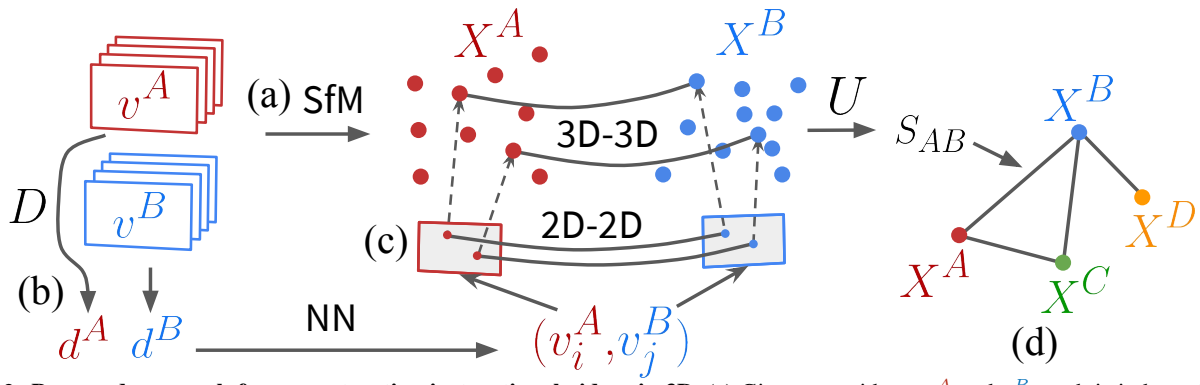

Figure 2: **Proposed approach for reconstructing instructional videos in 3D.** (a) Given two videos, v^A and v^B, each is independently reconstructed, producing the 3D models X^A and X^B. (b) Furthemore, image-level descriptors d^A and d^B are extracted with D, and used to obtain candidate pairs of frames (v_i^A, v_j^B). (c) Then, 2D-2D correspondences between these pairs of frames are obtained by matching local features with global constraints imposed by dense flow estimation, and 3D-3D correspondences are established from the 2D-2D correspondences by leveraging the individually reconstructed 3D models, X^A and X^B, and the camera parameters (extrinsics and intrinsics) for v_i^A and v_j^B. (d) The aggregated 3D-3D correspondences from several image pairs are passed to the solver U that robustly estimates a similarity transformation. Finally, a graph is constructed where each node corresponds to a 3D model, and each edge corresponds to a successful alignment between a pair of point-clouds. By traversing this graph, and composing the transformations along a given path, one can compute the similarities between any pair of connected nodes.

pair of frames (v_i^A, v_j^B), mutual matches between the sets of CNN-based local features (f_i^A, p_i^A) and (f_j^B, p_j^B) are first computed. Using local features allows for a very precise localization of the matches, but does not enforce any global consistency among different matches. In practice, this may lead to a high fraction of incorrect correspondences that may cause the subsequent alignment estimation step to fail. In order to reduce the fraction of incorrect correspondences while retaining the correct correspondences, we filter the matches produced by local features using a dense transformation field $\mathcal{T}_{ij}^{AB} = G(v_i^A, v_j^B)$ obtained from a dense flow model G, where $\mathcal{T}_{ij}^{AB} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a 2D-2D mapping between the pixels of v_i^A and those of v_j^B, and G is a CNN-based dense flow estimation model. Only matches between local features that are consistent with the dense flow mapping $\mathcal{T}_{ij}^{AB}$ up to certain tolerance threshold t are retained. An illustration if this approach is presented in Fig. 2c.

3.4. Estimating 3D transformation between videos

Once 2D-2D correspondences between frames of different videos are established, these can be used to, first, establish 3D-3D correspondences, and, second, determine a 3D similarity transformation between these videos.

Given a pair of videos (v^A, v^B), 2D-2D correspondences are first aggregated from all its matching frames $\{(v_i^A, v_j^B)\}$. Then, these 2D-2D correspondences are converted to 3D-3D correspondences by leveraging the camera parameters (intrinsic and extrinsic) together with the 3D models X^A and X^B obtained from their individual 3D reconstructions (cf. Sec. 3.1). Then, the set of 3D-3D correspondences is input to a solver U that robustly estimates the similarity transformation S_{AB} between 3D models X^A and X^B. However, some models may be difficult to align directly due to very large viewpoint changes, which may cause the matching method presented in Sec. 3.3 to fail. To overcome this issue, we propose to construct a *3D alignment graph* where every node represents a 3D model X^m and every edge represents a successful alignment between the 3D models of the two nodes that constitute the edge. Then, the alignment between any two connected models in the graph can be estimated by computing the shortest path [10] and then composing the pairwise transformations along the path. Note that by traversing the graph it is possible to align 3D models across large changes of viewpoint or appearance which would not be possible by direct matching. In order to register the models to a common reference frame, any node can be selected as reference and the similarities to every other node computed by traversing the graph. An illustration of this approach is presented in Fig. 2d. Appendix A provides additional details.

3.5. Application to keypoint transfer

In this section we describe how the proposed 3D alignment graph can be used to perform keypoint transfer among models. Because manual annotation is costly, annotation transfer has been used in the past as a way to obtain annotation on new examples in an automatic way [15, 50, 23]. In our case, we consider the problem of transferring annotations across several instructional videos showing the same object model. In particular, consider that we have a source video v^s, and a set of target videos $\{v^{t_1}, v^{t_2}, \ldots, v^{t_{N_t}}\}$, where N_t is the number of target videos. Consider as well that we are interested in a set of N_k keypoints. Then, we manually annotate these N_k keypoints on several frames of the source video v^s and triangulate their 3D positions

$K^s \in \mathbb{R}^{3\times N_k}$ in the source video reference frame, where the column of K^s correspond to the 3D coordinates of each different keypoint. Then, we can apply the similarity transformation S_{s,t_i} between the source video v^s and each target video v^{t_i} (obtained by traversing the 3D alignment graph) to transfer the keypoints K^s to each target video v^{t_j}:

$$K^{t_i} = S_{s,t_i} K^s \tag{1}$$

Finally, these transferred 3D keypoint annotations can be projected to any 2D frame $v_j^{t_i}$ of the target video v^{t_i} by using the estimated projection matrix $P_j^{t_i}$ (obtained from the product of the intrinsic and extrinsic camera parameters).

4. Text grounding in 3D

Given a 3D scene reconstruction, our goal is to obtain 3D text grounding, i.e., to identify the parts of the 3D model corresponding to a given text query. We follow [2, 29] and assume video narrations to be approximately synchronized with video frames of corresponding instructional videos and to describe their visual content. We consider alternative methods for localizing narrations in 2D frames and then backproject such location onto the 3D model. While the individual backprojected 3D positions may be noisy, e.g. because of the misalignment of the narration and the video, we obtain accurate positions by accumulating evidence in 3D from multiple videos.

Our approach is summarized in Fig. 3. We first employ a text encoder model B, that produces a $d-$dimensional text feature descriptor $f_s = B(s)$ from a given input text string s. We then use a classifier C to predict the distribution of scores $c = C(f_s) \in \mathbb{R}^n$ over a set of N_V voxels that discretize the 3D model space, for the given text features f_s. In the following, we describe our approach for training such classifier and text encoder in an unsupervised way from narrated instructional videos.

4.1. Generation of training data

We propose to automatically generate training samples for learning text grounding in 3D scenes. Our objective is to obtain training pairs (s_i, k_i) formed by the text s_i and the corresponding location in the scene represented by the 3D voxel label $k_i \in [0, N_V]$. Towards this goal we first split an instructional video v and its corresponding transcribed narration s into a set of temporal segments $\{(v_i, s_i)\}_{i=1,\dots,n}$. To obtain k_i for each segment, we first select a 2D position $p_i = (x_i, y_i)$ in video v_i, and then use the camera parameters (intrinsics and extrinsics) together with the reconstructed 3D model to obtain the corresponding 3D point P_i of the scene. We build a pair (s_i, k_i) by choosing a 3D voxel k_i containing P_i.

To obtain 2D image locations p_i, we ground narrations s_i in video frames by following three alternative approaches.

Center of frame. The first approach is to assign p_i to the frame center. This simple strategy turns out to be effective in instructional videos where narrated objects and actions are typically centered in the frame by the cameraman.

Hand detector. Our second approach follows the assumption that narrated objects are being manipulated by the user. We hence assign $p_i = H(v_i)$, where H is a position of a hand in the video obtained with a hand detector [41].

MIL-NCE. Our third approach builds on a pre-trained model for local text-video similarity. We use a joint text-video embedding model trained on a large set of narrated instructional videos with MIL-NCE [28]. This model projects video and text to a joint embedding space, which allows us to compute similarity between a local spatio-temporal video region and a text fragment. To ground the text narration within the frame, we select a position in a feature map with the highest text-video similarity score.

In all three cases described above, the selected 2D locations p_i within frames are reprojected to the surface of the reconstructed 3D model to obtain P_i and corresponding voxels k_i.

4.2. Learning 3D grounding model

Given a set of training pairs with narrations and corresponding voxels $\{(s_i, k_i)\}_{i=1,\dots,n}$, the classifier C is trained to predict the correct voxel label k_i given input text s_i. We train C by minimizing the cross-entropy loss over all training pairs:

$$-\sum_{i=1}^{n} \log\left(C(B(s_i))[k_i]\right), \tag{2}$$

where $C(B(X))[k]$ is the score of voxel k, given input text X, according to our model C. This loss is minimized when C assigns the maximum score of one to the correct voxel.

In practice, we are interested in training language grounding models for different 3D reconstructions. For example, for car maintenance videos, we build a separate 3D reconstruction for each specific car model, such as Ford Focus or Honda Civic, as the 3D shape of the car and the engine can be substantially different. However, while the classifier C is different for all 3D reconstructions, we choose to share the same text encoder B for all reconstructions. This allows us to benefit from all available text data, when training the text representation module. The entire model trained in a multi-task setting is shown in Fig. 3. Additional implementation details are provided in Sec. 5.

5. Experimental results

To evaluate our proposed method for 3D reconstruction and 3D grounding, we collect a dataset of narrated instructional videos related to car maintenance tasks. The tasks we

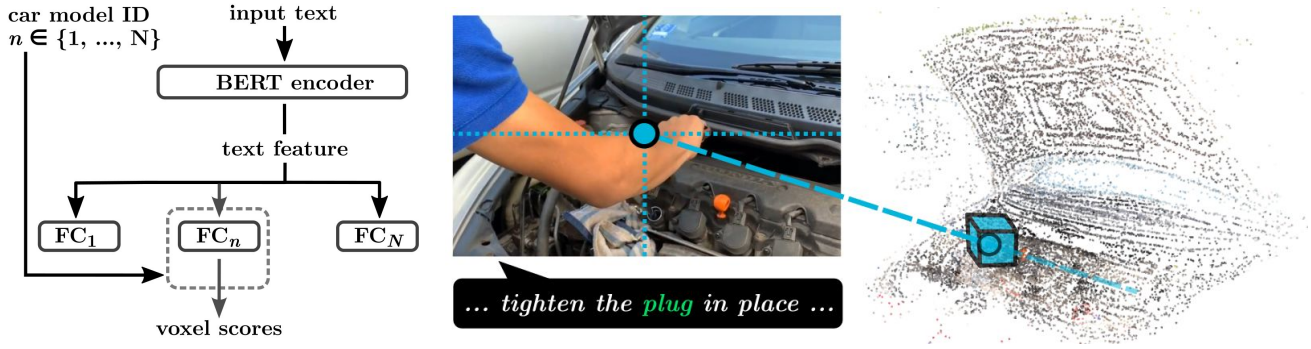

Figure 3: **Left.** Illustration of our 3D text grounding model. We train a separate model for each car make. All models share the same text encoder (BERT) and differ in the last grounding layer. The grounding layer is a single fully connected layer with the output dimension equal to the number of voxels. Given the input text, the model produces the confidence scores for all voxels in the 3D model. **Middle and Right.** Illustration of our training procedure. Given a word sequence from subtitles (bottom), and a 2D point in the frame coordinates (blue dot), we backproject the point onto the surface of the 3D model and find the corresponding voxel (blue cube). The pair (text, voxel) is then used as a training pair. In this example, we use the center of the frame as the approximate supervision to ground the text narration on the 3D model.

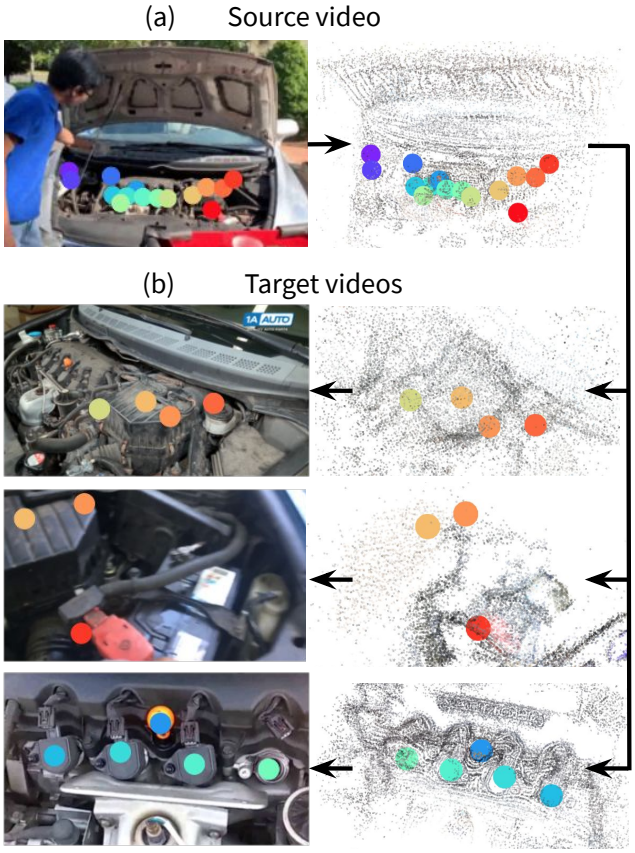

Figure 4: **Example of keypoint transfer results.** (a) 2D keypoints are annotated in several frames of the source video and triangulated to 3D keypoints in the 3D reference frame of source video. (b) These source keypoints are automatically transferred to the 3D reference frames of several target videos using the proposed alignment graph. The 3D keypoints can be finally projected to 2D keypoints for every registered frame in the target videos.

consider are the following: replacing the air filter, changing the spark plugs, replacing the car battery, measuring the oil level, adding coolant fluid and checking the break fluid. The car models we consider are: Honda Civic (8th gen), Honda Accord (7th gen), Ford Focus (Mk3), Ford Explorer (5th gen), Toyota Corolla (E140) and Toyota Prius (XW20). We select these car models as they are common in North America. We also focus our data collection on English-spoken videos. We have collected a total of 174 videos, with the number of videos per car model between 23 and 36. The number of videos per task varies between 14 (measure oil level) and 53 (change spark plugs).

In the following we first describe implementation details and then present results obtained for the keypoint transfer and 3D text grounding tasks. Our dataset, pre-trained models and code will become publicly available.

Implementation details. We use R2D2 [34] learnt local features as the feature extraction function L and GLU-Net [45] as the learnt dense flow estimation function G. The image level description-function D is done using a NetVLAD [5] model with a VGG16 backbone [42]. The solver U is composed by a RANSAC loop [13] that performs a transformation fitting using the least-squares method proposed by Umeyama [46].

For the text grounding, we use a pre-trained BERT model as the text encoder B. The output dimension d of the text encoder is set to 1024. A linear classifier is used for the grounding layer C, where the output dimensionality corresponds to the number N_v of voxels. The text enconder B is finetuned together with the grounding classifier C in an end-to-end manner. For the hand detector H, we use [41]. When using MIL-NCE [28] we obtain local similarity scores for any location of the frame by removing the last average pooling layer from the S3D backbone of the visual branch of the model and obtain a features map of size 14×8 for each

selected frame of the video.

We define a set of voxels as follows. First, consider a rectangular region, that encompasses the 3D model. This region is divided into 20 equal parts by each axis, resulting in a set of 8000 voxels. Since the shape of a car engine is relatively flat, we may expect that most of these voxels don't intersect with any parts of the model. In order to reduce the total number of voxels, and only retain the voxels, that intersect with objects of interest in the 3D scene, we select top N_v voxels, that contain most of the training points. In practice N_v is set to 500.

5.1. Keypoint transfer

For evaluating the task of keypoint transfer we annotate multiple frames for each Honda Civic video. We select one video as source (v^s), and the rest as target ($\{v^{t_i}\}_{i=1,\dots,N_t}$). Then, the annotations are used to triangulate the 3D keypoint positions K^s for the source video and for each of the target videos K^t_i, where $i = 1, \dots, N_t$. Then, we estimate a ground-truth similarity transformation between the source video and the target videos by fitting a similarity transformation using the solver U on the triangulated keypoints:

$$S^{GT}_{s,t_i} = U(K^s, K^{t_i}). \tag{3}$$

Finally, we can assess the quality of the similarity transformations S_{s,t_i} that we estimated using the proposed 3D alignment graph by computing the percentage of correctly transferred keypoints (PCK) in 3D coordinates. For evaluation, we use 89 pairs of videos for the Honda Civic car model, where the source and target keypoints K^s and K^t have been verified to have consistent relative distances and angles (a consistent overall 3D shape). Our results are presented in Fig. 5, where we plot the mean PCK over these 89 evaluation pairs vs. a varying tolerance threshold calibrated in centimeters. Results show that our proposed matching approach using a combination of R2D2 learnt local features and GLU-Net dense flow is superior to these two methods employed individually. Nevertheless, some video pairs are very difficult to match directly due to extreme viewpoint changes and our direct matching approach can only align about 50% of the keypoints for a 10cm tolerance value. In this cases the 3D alignment graph can be used to compute the similarity transformation via other intermediate videos. Results show that by using the 3D alignment graph, 80% of keypoints are correctly localized within a tolerance of 8cm. Examples of keypoint transfer using the 3D alignment graph are shown in Fig. 4 and, thanks to the 3D alignment graph, demonstrate transfer across an impressive range of viewpoints and appearance variation across the different car instances.

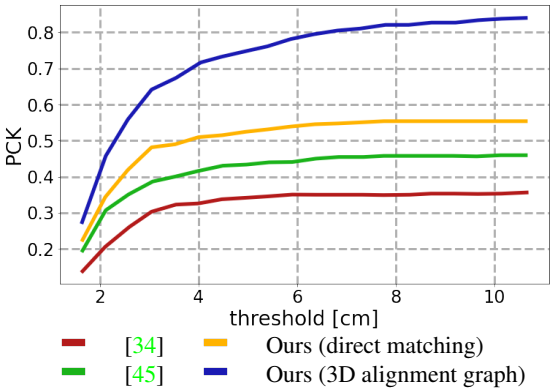

Figure 5: **Quantitative evaluation of keypoint transfer.** We plot the mean PCK curve over a set of 89 video pairs. The proposed 3D alignment graph correctly aligns 80% of keypoints within a 8cm tolerance.

5.2. Text grounding in 3D

Evaluation setup. To demonstrate the effectiveness of 3D text grounding, we select six different objects: *air filter*, *brake fluid reservoir*, *negative battery cable*, *positive battery cable*, *dipstick* and *spark plugs*, and annotate their positions within each 3D model. Each object is annotated with a single 3D point. We manually parse the WikiHow.com car repair articles and select a set of textual instructions, related to each object. These instructions are object-centric, each refering to a specific location within the engine, where one of the objects is located. Examples of such instructions include "remove the dipstick" (*dipstick*), "use a wrench to loosen the positive cable clamp and take the cable off of the terminal" (*positive battery cable*) and "lift the filter out of the housing" *(air filter*). At test time, we use these instructions as queries, with the goal to ground each query to the correct 3D location, defined by the corresponding point of interest. Since our model returns the scores over a set of voxels, instead of a single 3D point, we define the grounding by the model as the center of a highest scoring voxel.

Evaluation metric. We measure the performance of the grounding by the PCK score. This score is computed as follows. For each car model and for each query, we compute the distance between the predicted 3D point and the ground truth 3D point that corresponds to the object within the query. Then for each threshold value from 1cm to 100cm we compute PCK as the number of queries grounded correctly (*i.e.* the distance between the predicted point and the ground truth point is below the given threshold value), divided by a total amount of queries for all models.

Results and discussion. We evaluate our models, trained with three different types of supervision, as described in Sec. 4, and compare their performance against a chance baseline ("Chance"), which implies grounding each query

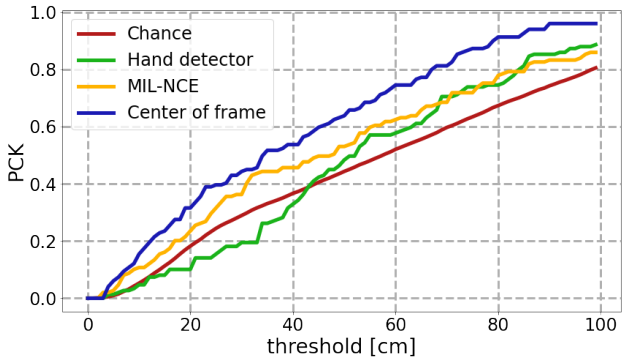

Figure 6: **Quantitative evaluation of 3D text grounding.** We plot PCK curve over a set of 6 different car models and 6 different objects of interest. For each object we consider a set of text queries, related to the corresponding location in the 3D model, and predict the 3D location of the object based only on the the language query. We compare our model trained with different types of supervision against a Chance baseline, where each query is grounded in a random point on the 3D model.

to a random point in the 3D model. Results are shown in Fig. 6. Interestingly, the best results are provided by the center of frame method ("Center of frame") confirming the previously observed photographer bias [7]. Supervision from the Hand detector ("Hand detector") suffers from a large amount of noise when the hands are visible in the video but are not near the object. The hand detector also provides less supervision than other methods, since hands may not appear within the frame at the time when the object is mentioned by the narration. Finally, in many cases, the appearance of hands in the videos coincide with an object mention in narration, when an action over the object is being performed. However, an action often involves using tools, such as a wrench or a screwdriver. Therefore, the position of the hand is often shifted with respect to the object, contributing to the noise in the supervision. We found MIL-NCE feature ("MIL-NCE") not well suited for the task. While, in some cases, this feature is able to provide good localization, it is usually very coarse, as MIL-NCE is not designed to accurately localize queries within the image. Results for different object classes are shown in Tab. 1. These results indicate that our model (here the best performing variant "Center of frame" is used) consistently outperforms the chance baseline for 5 object classes out of 6, and only fails on the dipstick object. Fig. 7 demonstrates qualitative results for different queries and different car models. Additional qualitative results are provided in Appendix B.

6. Conclusion

We present a method for 3D reconstruction of instructional videos and localizing the associated narrations in 3D. Our method is resistant to the differences in appearance of objects depicted in the videos and computationally efficient.

Object	Chance	Our method
Air filter	0.26	**0.55**
Brake fluid reservoir	0.19	**0.23**
Negative battery cable	0.07	**0.42**
Positive battery cable	0.05	**0.20**
Dipstick	**0.40**	0.35
Spark plugs	0.54	**0.69**
Average	0.29	**0.41**

Table 1: PCK grouped by object classes. We compare our model, trained on centers of frames, against the Chance baseline. Here the PCK error threshold is set to 30cm.

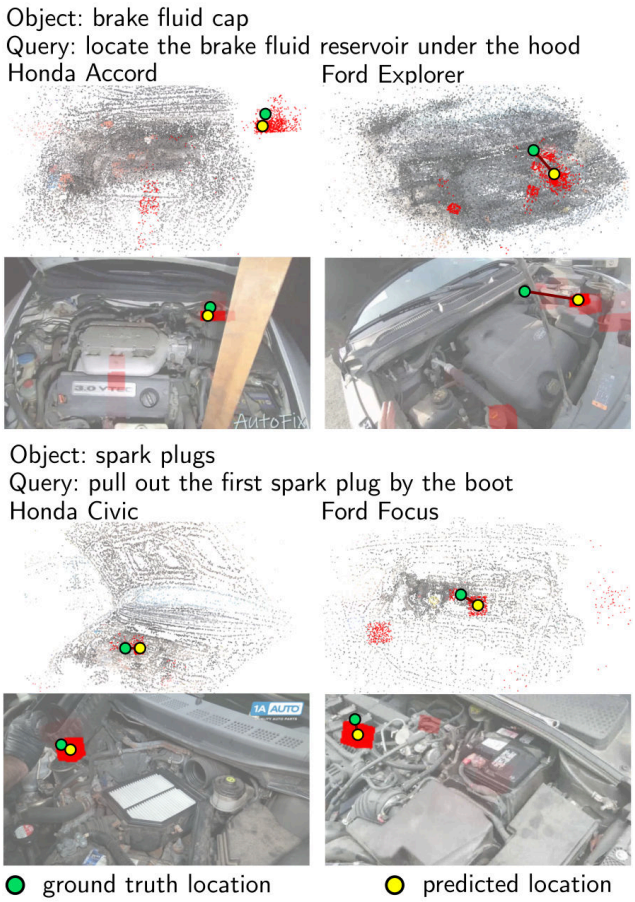

Figure 7: **Examples of 3D text grounding with our model.** Our model predicts a heatmap over a set of voxels with higher values assigned to locations that correspond to the query text. For clarity, we visualize the predicted voxels in random frames from the videos. The confidence that a given text query corresponds to a given location is shown in red with higher opacity corresponding to higher confidence. The ground truth 3D location is shown by a green dot. Please see additional results in Fig. 9.

Building on our 3D reconstruction, we propose an unsupervised approach to 3D language grounding. Our work opens-up the possibility of providing visual hints, based on textual instructions for augmented reality applications.

Acknowledgements. This work was supported in part by the MSR-Inria joint lab, the Louis Vuitton ENS Chair on Artificial Intelligence, the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15003/0000468), and the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). This work was granted access to the HPC resources of IDRIS under the allocation 20XX-AD011011329.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 2

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Ivan Laptev, Josef Sivic, and Simon Lacoste Julien. Unsupervised learning from narrated instruction videos. 2016. 5

[3] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Learning from narrated instruction videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, XX, Sept. 2017. 2

[4] Jean-Baptiste Alayrac, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. 2017. 2

[5] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. 2016. 6

[6] Plummer B.A., Kordas P., Kiapour M.H., Zheng S., Piramuthu R., and Lazebnik S. Conditional image-text embedding networks. In *Computer Vision – ECCV 2018*, pages 258–274. Springer International Publishing, 2018. 3

[7] Russell Bryan C., Martin-Brualla Ricardo, Butler Daniel J., Seitz Steven Maxwell, and Zettlemoyer Luke S. 3d wikipedia: using online text to automatically label and navigate reconstructed geometry. In *ACM*, 2013. 3, 8

[8] Henry Chen, Austin S Lee, Mark Swift, and John C Tang. 3d collaboration method over hololens™ and skype™ end points. In *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, pages 27–30, 2015. 2

[9] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVA*, 2014. 2

[10] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. 4

[11] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. 2020. 2

[12] Martin Engilberge, Louis Chevallier, Patrick Pérez, and Matthieu Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[13] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 6

[14] Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. 2010. 2

[15] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. 110(3):328–348, 2014. 4

[16] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. 2016. 2

[17] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding "It": Weakly-Supervised Reference-Aware Visual Grounding in Instructional Videos. June 2018. 2

[18] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. 2016. 2

[19] De-An Huang, Joseph J. Lim, Li Fei-Fei, and Juan Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. 2017. 2

[20] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. 2008. 2

[21] Andrej Karpathy, Armand Joulin, and Fei Fei F Li. Deep fragment embeddings for bidirectional image sentence mapping. pages 1889–1897, 2014. 3

[22] Jonghyun Kim, Youngmo Jeong, Michael Stengel, Kaan Akşit, Rachel Albert, Ben Boudaoud, Trey Greer, Joohwan Kim, Ward Lopes, Zander Majercik, et al. Foveated ar: dynamically-foveated augmented reality display. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019. 2

[23] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. 2017. 4

[24] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. 2019. 2

[25] David G Lowe. Object recognition from local scale-invariant features. 1999. 2

[26] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's cookin'? Interpreting cooking videos using text, speech and vision. In *NAACL*, 2015. 2

[27] Jorge Martín-Gutiérrez, Carlos Efrén Mora, Beatriz Añorbe-Díaz, and Antonio González-Marrero. Virtual technologies trends in education. *EURASIA Journal of Mathematics, Science and Technology Education*, 13(2):469–486, 2017. 2

[28] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos, 2020. 2, 5, 6

[29] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. 2019. 2, 5

[30] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. 2019. 2

[31] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Adaptive structure from motion with a contrario model estimation. 2012. 2

[32] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 2

[33] Philip Pratt, Matthew Ives, Graham Lawton, Jonathan Simmons, Nasko Radev, Liana Spyropoulou, and Dimitri Amiras. Through the hololens^{TM} looking glass: augmented reality for extremity reconstruction surgery using 3d vascular models with perforating vessels. *European radiology experimental*, 2(1):1–7, 2018. 2

[34] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. R2D2: repeatable and reliable detector and descriptor. 2019. 6, 7

[35] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. 2017. 2

[36] Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. 2018. 2

[37] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. 2018. 2

[38] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. 2018. 2

[39] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. 2016. 2

[40] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. 2018. 2

[41] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale. June 2020. 5, 6

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014. 6

[43] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006. 2

[44] Chris Sweeney. Theia multiview geometry library: Tutorial & reference. http://theia-sfm.org. 2

[45] Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and correspondences. 2020. 6, 7

[46] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Computer Architecture Letters*, 13(04):376–380, 1991. 6

[47] L. Wang, Y. Li, J. Huang, and S. Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):394–407, 2019. 3

[48] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3

[49] Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. Structured matching for phrase localization. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV 2016*, pages 696–711, Cham, 2016. Springer International Publishing. 3

[50] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. 2019. 4

[51] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 2

[52] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[53] Fang Zhao, Jianshu Li, Jian Zhao, and Jiashi Feng. Weakly supervised phrase localization with multi-scale anchored transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[54] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 2

[55] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. 2020. 2

[56] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. June 2019. 2

Appendix

This appendix presents additional results of our method. In Sec. A, we provide additional qualitative results demonstrating the benefits of using the 3D alignment graph. In Sec. B we provide additional qualitative results for text grounding in 3D.

A. Estimating 3D transformation between videos

In this section we further illustrate details for our approach to 3D registration. In particular, Fig. 8 presents the procedure for building and using the 3D alignment graph. See Fig. 8 caption for explanations.

B. Text grounding in 3D

Fig. 9 demonstrates additional results of our text grounding similar to the results shown in Sec. 5 of the main paper. Note that our method relies on the underlying 3D model, instead of image appearance. As result, our method is capable to ground queries even when the corresponding object is partially or fully occluded.

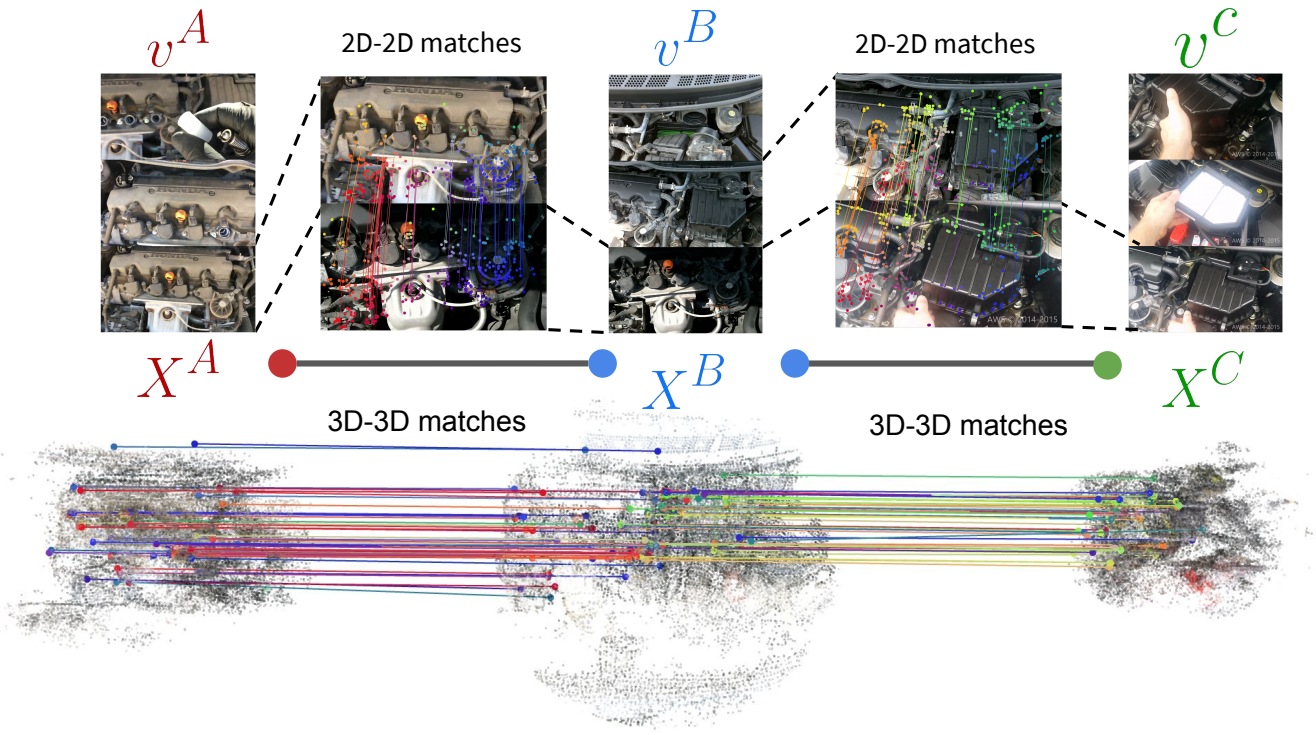

Figure 8: **Illustration of the proposed alignment method and the 3D alignment graph.** Given two videos v^A and v^B, pairs of matching frames are found using global image descriptors, and are then matched using our approach to obtain 2D-2D matches (top). These 2D-2D matches are then translated into 3D-3D matches between the 3D point clouds X^A and X^B corresponding to v^A and v^B, respectively (bottom). These 3D-3D matches are used to estimate the similarity transformation S_{AB} between X^A and X^B, using the function U. Furthermore, the constructed 3D alignment graph can be used to align two videos v^A and v^C which have little overlap (i.e. they focus on different parts of the object), by traversing the edges of the graph using one or more intermediate videos and 3D models (v^B and X^B in this example).

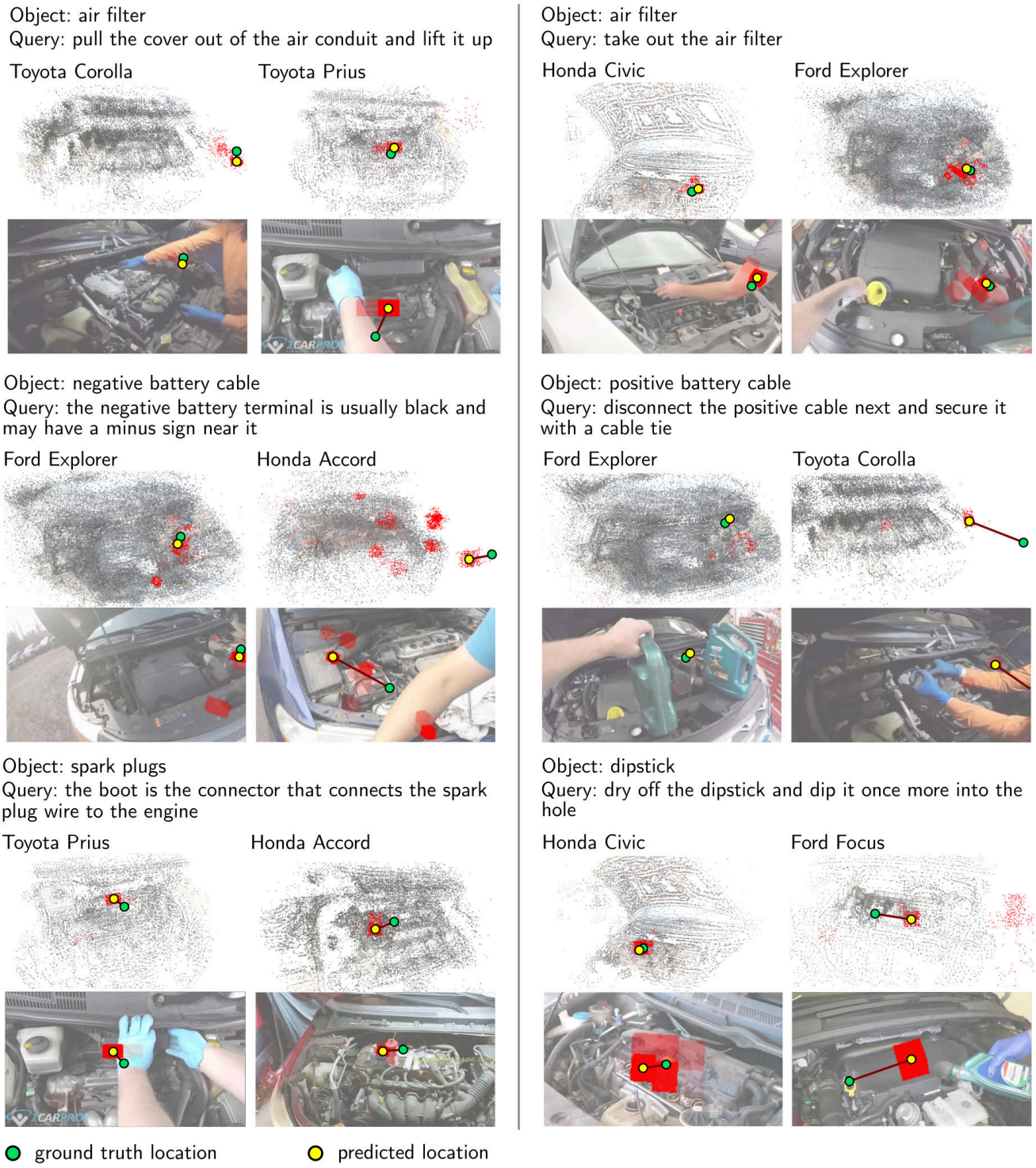

Figure 9: **Additional examples of 3D text grounding with our model.** Our model predicts a heatmap over a set of voxels with higher values assigned to locations that correspond to the query text. For clarity, we visualize the predicted voxels in random frames from the videos. The confidence that a given text query corresponds to a given location is shown in red with higher opacity corresponding to higher confidence. The ground truth 3D locations are shown by green dots. Predicted points are shown by yellow dots and correspond to the centers of voxels with the maximum score.