Learning Action Maps of Large Environments via First-Person Vision

Nicholas Rhinehart, Kris M. Kitani
The Robotics Institute
Carnegie Mellon University
{nrhineha, kkitani}@cs.cmu.edu

Abstract

When people observe and interact with physical spaces, they are able to associate functionality to regions in the environment. Our goal is to automate dense functional understanding of large spaces by leveraging sparse activity demonstrations recorded from an ego-centric viewpoint. The method we describe enables functionality estimation in large scenes where people have behaved, as well as novel scenes where no behaviors are observed. Our method learns and predicts "Action Maps", which encode the ability for a user to perform activities at various locations. With the usage of an egocentric camera to observe human activities, our method scales with the size of the scene without the need for mounting multiple static surveillance cameras and is well-suited to the task of observing activities up-close. We demonstrate that by capturing appearance-based attributes of the environment and associating these attributes with activity demonstrations, our proposed mathematical framework allows for the prediction of Action Maps in new environments. Additionally, we offer a preliminary glance of the applicability of Action Maps by demonstrating a proof-of-concept application in which they are used in concert with activity detections to perform localization.

1. Introduction

The goal of this work is to endow intelligent systems with the ability to understand the functional attributes of their environment. Such functional understanding of spaces is a crucial component of holistic understanding and decision making by any agent, human or robotic. Functional understanding of a scene can range from the immediate environment to the distant. For example, at the scale of a single room, a person can perceive the arrangement of tables, chairs, and computers in an office environment, and reason that they could sit down and type at the computer. People can also reason about the functionality about nearby rooms, for example, the presence of a kitchen down the hall from the office is useful functional and spatial information for when the person decides to prepare a meal. The goal of this
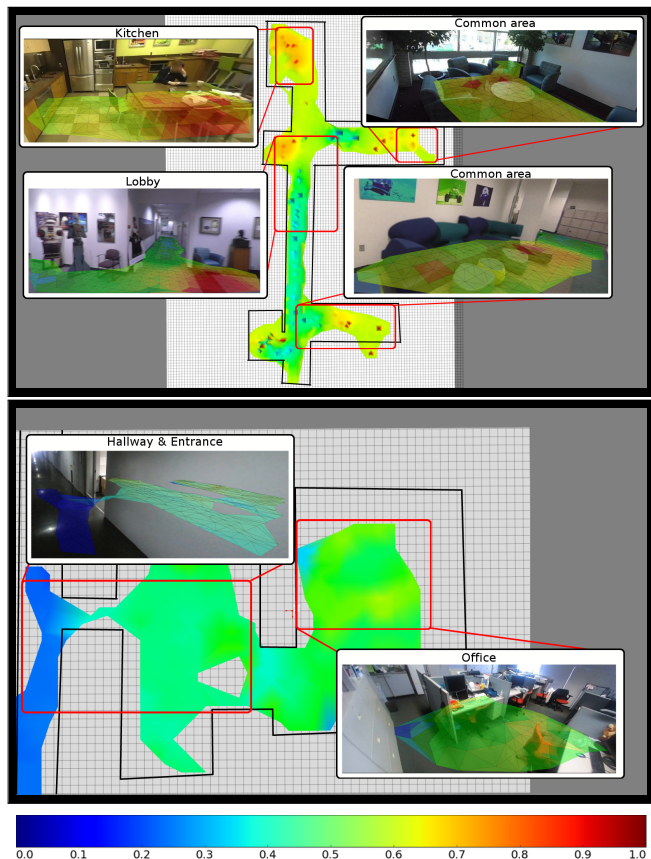

Figure 1: Action Map prediction for the `sit` activity by using our method to combine appearance data and activity observations. Activity and appearance information from the top scene in combination with *only* appearance information (no activity observations) from the bottom scene is used to model the relationship between activities, scene information, and object information to make predictions for both scenes. Areas in the scenes where a person can sit are estimated by our method, such as the chairs and couches in both views.

work is to learn a computational model of the functionality of large environments, called *Action Maps* (AMs), by observing human interactions and the visual context of those

action within a large environment.

There has been significant work in the area of automating the functional understanding of an environment, though much has focused on single scenes [9, 4, 11, 10, 2, 5]. In this work, we aim to extend automated functional understanding to very large spaces (*e.g.*, an entire office building or home). This presents two key technical challenges:

- How can we capture observations of activity across large environments?

- How can we generalize functional understanding to handle the inevitable data sparsity of less explored or new areas?

In order to address the first challenge of observing activity across large environments, we take a departure from the fixed surveillance camera paradigm, and propose an approach that uses a first-person point-of-view camera. By virtue of its placement, its view of the wearer's interactions with the environment is usually unobstructed by the wearer's body and other elements in the scene. An egocentric camera is portable across multiple rooms, whereas fixed cameras are not. An egocentric camera allows for the observation of hand-based activities, such typing or opening doors, as well as the observation of some ego-motion based activities, such as sitting down or standing. The first-person paradigm is well suited for large-scale sensing and allows observation of interactions with many environments.

Although we can capture a large number of observations of activity across large environments with wearable cameras, it is still not practical to wait to observe all possible actions in all possible locations. This leads to the second technical challenge of generalizing functional understanding from a *sparse* set of action observations, which requires generalization to new locations. Our method generalizes by using another source of visual observation – which we call *side-information* – that encodes per-location cues relevant to activities. In particular, we propose to extract visual side-information using scene classification [22] and object detection [6] techniques. With this information, our method learns to model the relationship between actions, scenes, and objects. In a scene with no actions, we use scene and object information, coupled with actions in a separate scene, to infer possible actions. We propose to solve the problem of generalizing functional understanding (*i.e.*, generating dense AMs) by formulating the problem as matrix completion. Our method constructs a matrix where each row represents a location and each column represents an action type (*e.g.*, read, sit, type, write, open, wash). The goal of matrix completion is to use the observed entries to fill the missing entries. In this work, we make use of Regularized Weighted Non-Negative Matrix Factorization (RWNMF) [7], allowing us to elegantly leverage *side-information* to model the relationship between activities, scenes, and objects, and predict missing activity affordances.

Figure 2: Projected Action Map examples learned by our method. With global estimates of large Action Maps produced by our method, we use localized images within the scene to show visualizations of the Action Maps by projecting them to the images.

1.1. Contributions

To the best of our knowledge, this is the first work to generate *Action Maps*, such as those in Figures 1 and 2, over large spaces using a wearable camera. The first-person vision paradigm is an essential tool for this problem, as it can capture a wide range of visual information across a large environment. Our approach unifies scene functionality information via a regularized matrix completion framework that appropriately addresses the issue of sparse observations and provides a vehicle to leverage visual side information.

We demonstrate the efficacy of our proposed approach on five different multi-room scenes: one home and four office environments. Our experiments in real large-scale environments show how first-person sensing can be used to efficiently observe human activity along with visual side-information across large spaces. **1)** We show that our method can be used to model visual information from both single and multiple scenes simultaneously, and makes efficient use of all available activity information. **2)** We show that our method's power increases as the set of performed activity increases. **3)** Furthermore, we demonstrate how our proposed matrix factorization framework can be used to leverage sparse observations of human actions along with visual side-information to perform functionality estimation of large *novel* scenes in which *no activities* have been demonstrated. We compare our proposed method against natural baselines such as object-detection-based Action Maps and scene classification, and show that our approach outperforms them in nearly all of our experiments. **4)** Additionally, as a proof-of-concept application of the rich information in an Action Map, we present an application of our Action Maps as priors for localization.

1.2. Background

Human actions are deeply connected to the scene. Scene context (*e.g.*, a chair or common room) can be a strong in-

dicator of actions (*e.g.*, sitting). Likewise, observing an action like sitting, is a strong indicator that there must be a sittable surface in the scene. In the context of time lapse video, Fouhey *et al.* [4] used detection of sitting, standing, and walking actions to obtain better estimates of 3D geometry for a single densely explored room. Gupta *et al.* [9] addressed the inverse problem of inferring actions from estimated 3D scene geometry using a single image of a room. Their approach synthetically inserted skeleton models into the 3D scene to reason about possible functional attributes of the scene. Delaitre *et al.* [2] also used time lapse video of human actions to learn the functional attributes of objects in a single scene. The work of Savva *et al.* [16] obtains a dense 3D representation of small workspace (*e.g.* desk and chair space) and learns the functional attributes of the scene by observing human interactions. Similar to previous work, this work seeks to understand the functionality of scenes. However, limitations of previous work include the reduced size of the physical space and the presumed density of interactions. In contrast, our approach attempts to infers the dense functionality over an entire building (*e.g.*, office floor or house), and reasons about multiple large scenes simultaneously by modeling the relationship between scene information, object information, and sparse activities.

Another flavor of approaches reason in the joint space of activities and objects. In Moore *et al.* [13], human actions are recognized by using information about objects in the scene. Gall *et al.* [5] uses human interaction information to perform unsupervised categorization of objects. Other approaches have capitalized on the interplay between actions and objects: Gupta *et al.* [8] demonstrate an approach to use object information for pose detection, and Yao *et al.* [20] jointly model objects and poses to perform recognition of both objects and actions. The approach of [14] performs object recognition by observing human activities, and notes an important idea that our approach also uses: whereas object information may sometimes be too small in detail, human activities usually are not. We capitalize on this observation close-up observation capability of an egocentric camera.

The egocentric paradigm is an excellent method for understanding human activities at close range [18, 3, 15, 12]. Our work builds on such egocentric action recognition techniques by associating actions with physical locations in a single holistic framework. By bringing together ideas from single image functional scene understanding, object functionality understanding and egocentric action analysis, we propose a computational model that enables cross-building level functional understanding of scenes.

2. Constructing Action Maps

Our goal is to build *Action Map*s that associate possible actions for every spatial location on a map over a large environment. We decompose the process into three steps. We first build a physical map of the environment by using egocentric videos to obtain a 3D reconstruction of the scene using structure from motion. Second, we use a collection of recorded human activity videos recorded with an egocentric camera to detect and spatially localize actions. This collection of videos is also used to learn the visual context of actions (*i.e.*, scene appearance and object detections) which is later used as a source of side information for modeling and inference. Third, we aggregate the localized action detection and visual context data using a matrix completion framework to generate the final Action Map. The focus of our method is the third step, which we describe next. We mention how we obtain the visual context in Section 2.1.1, and describe the first two steps in detail in Section 3.2.

2.1. Action Map Prediction as Matrix Factorization

We now describe our method for integrating the sparse set of localized actions and visual side-information to generate a dense Action Map (AM) using regularized matrix completion. Our goal is to recover an AM in matrix form $\mathbf{R} \in \mathbb{R}_+^{M \times A}$, where M is the number of locations on the discretized ground plane and A is the number of possible actions. Each row of the AM matrix $\mathbf{R}$ contains the action scores $\mathbf{r}_m$, where m is a location index, and each entry r_{ma} describes the extent to which an activity a can be performed at location m. To complete the missing entries of $\mathbf{R}$, we design a similarity metric for our side-information, enabling the method to model the relationship between activities, scenes, and objects.

We impose structure on the rows and columns of the AM matrix by computing similarity scores with the side-information. Examples of this side information are shown in Figure 3, where two features from scene classification, plus one feature from object detection are shown in the same physical space as the AM. Figure 3 serves to further motivate the idea of exploiting scene and object information between two different scenes to relate the functionality of the scenes. We define three kernel functions based on scene appearance, object detections and spatial continuity. This structure is integrated as regularization in the RWNMF objective function (Equation 2).

2.1.1 Integrating Side-Information

To integrate side-information into our formulation, we build two weighted graphs that describe the cross-location (row) similarities, and cross-action (column) similarities. We are primarily interested in the cross-location similarities, and discuss how we handle the cross-action similarities in Section 2.2. To build the cross-location graph, we aggregate the spatial proximity, scene-classification, and object detection information as a linear combination of kernel-based similarities, as shown in Equation 1.

For every location a in the AM, we compute the scene classification score $\mathbf{p}_a = [p_{1a} \dots p_{Ca}]$ for each image as
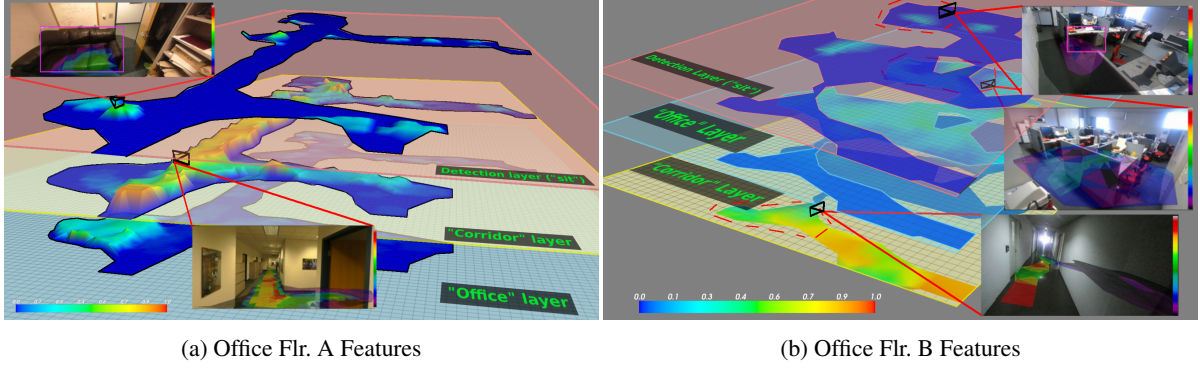
(a) Office Flr. A Features　　　　　　　(b) Office Flr. B Features

Figure 3: Several Office Flr. A and Office Flr. B Features. The "office" and "corridor" layers correspond to the features from the scene classification CNN, and the "sit" layer corresponds to the object detection CNN features aggregated across all `sit`-able objects, which is also one of the baselines as described in Section 3.2. This figure demonstrates our idea that object information and scene information can be used to relate scenes to each other. This relationship is the basis for transferring and sharing activity functionality between scenes. Heatmaps from several layers are shown projected into localized images from the scene. Note that the "office" portion of Office Flr. A also contains sittable regions, and that the much larger "office" area in Office Flr. B contains a select few sittable regions. The corridors in both scenes are described well by the features, and these areas strongly correlate with an *an absence* of functionality, as scene in Figure 1.

the average of the C-dimensional outputs from the Places-CNN of images within a small radius.

We use Structure-from-Motion (SFM) keypoints inside each detection to estimate the back-projected 3D location of the detected object in the environment by taking the mean of their 3D locations, which are then projected to the ground plane to form a set $\mathcal{D}_f$ for each object category $f \in [1 \dots F]$. The SFM reconstruction is also used to localize images and described further in Section 3.2. We calculate the object detection scores $\mathbf{o}_a = [o_{1a} \dots o_{Fa}]$ for each location a as the max score of object detection of the nearby back-projected object detections $d \in \mathcal{D}_f$ within a $r = \sqrt{2}$ grid-cell radius, exponentially weighted by its distance along the floor from the object z_d:

$$o_{fa} = \max_{d \in \mathcal{D}_f} \frac{1}{\sqrt{2r^2\pi}} \exp \frac{-z_d^2}{2r^2}.$$

We wish to enforce similarity of activities between nearby locations, as well as between locations that have similar object detections and scene classification description. Between any two locations a, b, and given associated scene classification scores $\mathbf{p}_a, \mathbf{p}_b$, object detection scores $\mathbf{o}_a, \mathbf{o}_b$, and 2D grid locations $\mathbf{x}_a, \mathbf{x}_b$ the kernel is of the form:

$$k(a,b) = (1 - \alpha)k_s(\mathbf{x}_a, \mathbf{x}_b) + \frac{\alpha}{2}k_p(\mathbf{p}_a, \mathbf{p}_b) + \frac{\alpha}{2}k_o(\mathbf{o}_a, \mathbf{o}_b'), \quad (1)$$

where k_s is an RBF kernel between the spatial coordinates of each location, k_p and k_o as χ^2 kernels on scene classification scores and object detection scores, and k_o has 0 similarity between locations with no object score.

Thus, there is a tradeoff between the k_s, k_p and k_o kernels, controlled by α. When $\alpha = 0$, only spatial smooth-

ness is considered, and when $\alpha = 1$, only scene classification and object detection terms are considered, ignoring spatial smoothness. When a location in one scene is compared to a location in a new scene or the same scene, $k(\cdot, \cdot)$ returns higher scores for locations with similar objects and places, and as shown Section 2.2, places more regularization constraint on the objective function, rewarding solutions that predict similar functionalities for *both* locations.

2.2. Completing the Action Map Matrix

To build our model, we seek to minimize the RWNMF objective function in Equation 2:

$$J(\mathbf{U}, \mathbf{V}) = \left\| \mathbf{W} \circ (\mathbf{R} - \mathbf{U}\mathbf{V}^T) \right\|_F^2 + \frac{\lambda}{2} \sum_{i,j}^{M} \|\mathbf{u}_i - \mathbf{u}_j\| \mathbf{K}_{ij}^U$$
$$+ \frac{\mu}{2} \sum_{i,j}^{A} \|\mathbf{v}_i - \mathbf{v}_j\| \mathbf{K}_{ij}^V$$
$$(2)$$

where $\mathbf{U} \in \mathbb{R}_+^{M \times D}$, $\mathbf{V} \in \mathbb{R}_+^{A \times D}$, together form the decomposition, $\mathbf{W} \in \mathbb{R}_+^{M \times A}$ is the weight matrix with 0s for unexplored locations, and $\mathbf{K}^U$ the kernel Gram matrix of the side information defined by its elements: $\mathbf{K}_{ij}^U = k(i,j)$. The squared-loss term penalizes decompositions with values different from the observed values in $\mathbf{R}$. The term involving $\mathbf{K}^U$ penalizes decompositions in which highly similar locations have different decompositions in the rows $(\mathbf{u}_i^T)$ of $\mathbf{U}$. Roughly, locations with high similarity in scene appearance, object presence, or position impose penalty on the resulting decomposition for predicting different affordance values in the AM. The term involving $\mathbf{K}_V$ corresponds to the cross-action smoothing, which we take as the

	# GT locs.	# Actions	Length	r_e	r_a
Office Flr. A	40	90	53.3 min.	0.59	0.03
Office Flr. D	15	44	32.8 min.	0.23	0.03
Office Flr. C	44	14	12.2 min.	0.16	0.01
Office Flr. B	50	13	3.3 min.	0.67	0.04
Home A	15	17	13.4 min.	0.75	0.04

Table 1: Scene stats. The number of GT locations is the number of distinct places a specific activity can be performed. The number of activity demonstrations is the total number of demonstrations collected in each environment. $r_e = \frac{\#\text{cells explored}}{\#\text{total cells}}$, $r_a = \frac{\#\text{cells with non-empty actions}}{\#\text{total cells}}$.

identity matrix, enforcing no penalty for differences across per-location action labels.

To minimize the objective function, we use the regularized multiplicative update rules following [7]. Multiplicative update schemes for NMF are generally constructed such that their iterative application yields a non-increasing update to the objective function; [7] showed that these update rules yield non-increasing updates to the objective function. Thus, after enough iterations, a local minima in the objective function is found, and the resulting decomposition and its predictions are returned.

Values in $\mathbf{W}$ are set to counteract class imbalance. The number of observed values for each activity is computed as n_c, and assigned to each nonempty location i's corresponding entry as $w_{ic} = 1/n_c$, and the zeros from observed cameras associated with no activities as $w = 1/n_z$.

3. Experiments

Our dataset consists of 5 large, multi-room scenes from various locations. Three scenes, Office Flr. A, Office Flr. D, and Office Flr. C, are taken from three *distinct* office buildings in the United States, and another scene, Office Flr. B, comes from an office building in Japan. Each office scene has standard office rooms, common rooms, and a small kitchen area. A final scene, Home A, consists a kitchen, a living room, and a dining room. See Table 1 for scene activity and sparsity statistics. Our goal is to predict dense Action Maps from sparse activity demonstrations.

The first experiments (Section 3.3) measure our method's performance when supplied with all observed action data that covers on average about half of all locations and some actions (See Table 1 for the coverage statistics). Additionally, this experiments compares against performance of the spatial kernel-only approach, which serves to illustrate the utility of including side-information. However, as it takes some time to collect the observations of each scene, we demonstrate a second set of experiments (Section 3.4), to showcase our method handling fractions of the already sparse observation data while still maintaining reasonable performance. In Section 3.5, our third set of experiments shows that if our method is presented with novel scenes for which there is *zero* activity demonstrations,

	W. Max F1	W. Mean F1	Max F1	Mean F1
Of. Flr. A S sng	0.73	0.72 ± 0.01	0.44	0.43 ± 0.02
Of. Flr. A SOP_D sng	0.63	0.61 ± 0.01	0.34	0.32 ± 0.01
Of. Flr. A SOP sng	0.74	0.69 ± 0.04	**0.56**	0.5 ± 0.04
Of. Flr. A SOP_D all	0.75	0.71 ± 0.02	0.44	0.43 ± 0.01
Of. Flr. A SOP all	**0.76**	**0.73** ± 0.02	0.54	**0.51** ± 0.02
Of. Flr. B S sng	0.56	0.55 ± 0.01	0.38	0.38 ± 0.01
Of. Flr. B SOP sng	0.56	0.55 ± 0.01	0.44	0.38 ± 0.03
Of. Flr. B SOP_D all	**0.58**	**0.56** ± 0.01	0.39	0.37 ± 0.03
Of. Flr. B SOP all	**0.58**	**0.56** ± 0.01	**0.53**	**0.44** ± 0.04
Of. Flr. C S sng	0.74	**0.66** ± 0.1	0.48	0.42 ± 0.06
Of. Flr. C SOP_D sng	0.67	0.46 ± 0.08	0.41	0.29 ± 0.05
Of. Flr. C SOP sng	0.68	0.53 ± 0.1	0.53	0.44 ± 0.06
Of. Flr. C SOP_D all	0.67	0.55 ± 0.06	0.45	0.38 ± 0.03
Of. Flr. C SOP all	**0.77**	0.58 ± 0.07	**0.56**	**0.46** ± 0.04
Of. Flr. D S sng	0.68	0.57 ± 0.11	0.57	0.45 ± 0.12
Of. Flr. D SOP_D sng	0.56	0.49 ± 0.05	0.37	0.32 ± 0.04
Of. Flr. D SOP sng	0.69	0.55 ± 0.08	0.68	0.54 ± 0.07
Of. Flr. D SOP_D all	0.81	0.68 ± 0.07	0.59	0.46 ± 0.08
Of. Flr. D SOP all	**0.82**	**0.73** ± 0.08	**0.77**	**0.61** ± 0.09
Home A S sng	0.57	0.53 ± 0.04	0.35	0.34 ± 0.02
Home A SOP_D sng	0.5	0.48 ± 0.01	0.26	0.24 ± 0.02
Home A SOP sng	**0.62**	**0.6** ± 0.01	0.43	**0.4** ± 0.02
Home A SOP_D all	0.52	0.49 ± 0.03	0.27	0.25 ± 0.02
Home A SOP all	**0.62**	0.55 ± 0.03	**0.45**	**0.4**±0.02

Table 2: Prediction results by using the activity observations for each scene ("sng"), and, as separate results, by simultaneously fitting data from all scenes ("all"). By using observations from all scenes, the performance of our method on each scene improves over using each scene's observation data alone. Additionally, our method is able to integrate activity detections without much performance loss: a D suffix indicates activity detection predictions were used, otherwise, labelled activities were used. "S" stands for spatial kernel only, and "SOP" stands for "Spatial+Object Detection+Scene Classification" kernels. The spatial kernel only is useful yet outperformed by the full model. Side information from multiple scenes generally improves the performance.

our method can still make predictions in these new environments. This final set of experiments also investigates which side-information is most helpful for our task.

3.1. Performance scoring

To evaluate an AM, we perform binary classification across all activities and compute mean $F1$ scores. We collect the ground truth activity classes for every image in the scene by retrieving them from labelled grid cells, as shown in Figure 5, in a small triangle in front of each camera, which represents the viewable space. We collect the predicted AM scores from the same grid cells and average the scores to produce per-image AM scores. We used 100 evenly-spaced thresholds to evaluate binary classification performance by averaging $F1$ scores across the thresholds. We report $F1$ scores as opposed to the overall accuracy, as the overall accuracy of our method is very high due to the large amount of space in each scene with no labelled functionality (a large amount of "true negatives"). The activity classes we use are sit, type, open-door, read,

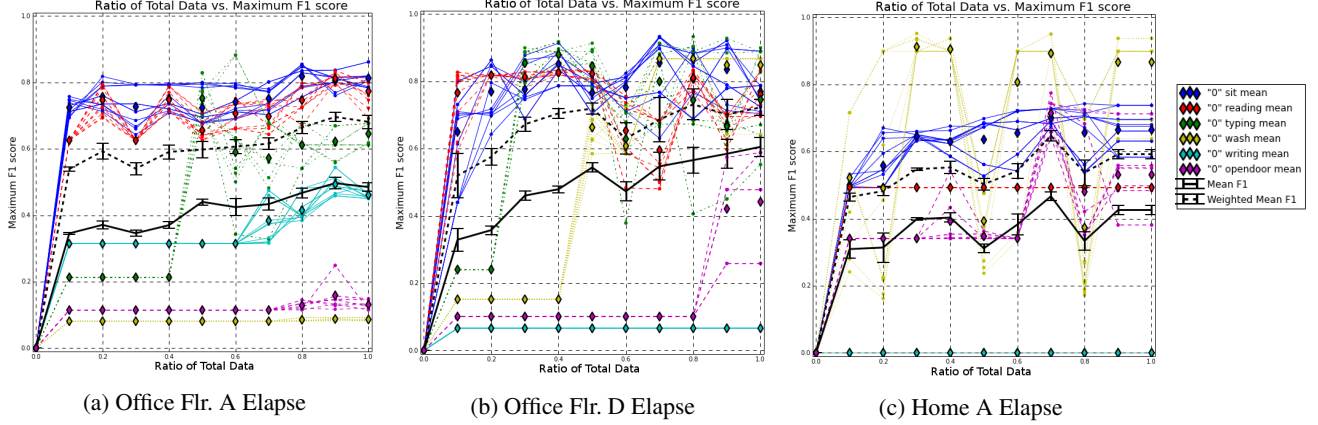(a) Office Flr. A Elapse (b) Office Flr. D Elapse (c) Home A Elapse

Figure 4: Performance improves a function of available data. For each parameter setting, we show the $F1$ scores for each activity label, as well as the mean and weighted mean of the $F1$ scores across all parameter settings and activity labels. Some variations in performance are observed as new activities are introduced, as the correlations between an established activities and newly introduced activities are initially sparse. As more data is collected, erroneous correlations are unlearnt, and correct ones are reinforced.

		W. Max F1	W. Mean F1	Max F1	Mean F1		W. Max F1	W. Mean F1	Max F1	Mean F1
	RFC	0.38	0.38	**0.62**	**0.62**		0.27	0.27	0.41	**0.41**
	Det.	0.59	0.59	0.33	0.33		0.44	0.44	0.28	0.28
	NMF	0.35	0.35	0.24	0.24	Office Flr. D	0.65	0.65	0.40	0.40
Office Flr. B	**SO**	0.69	0.67 ± 0.02	0.44	0.42 ± 0.01		0.65	0.51 ± 0.12	0.46	0.36 ± 0.09
	SP	**0.74**	**0.69** ± 0.02	0.46	0.43 ± 0.02		**0.68**	**0.55** ± 0.12	**0.51**	0.38 ± 0.09
	SOP	0.57	0.54 ± 0.03	0.28	0.26 ± 0.02		0.42	0.36 ± 0.02	0.28	0.25 ± 0.01
	RFC	0.24	0.24	0.37	0.37		0.28	0.28	0.35	0.35
	Det.	0.54	0.54	0.31	0.31		0.53	0.53	0.25	0.25
	NMF	0.39	0.39	0.27	0.27	Home A	0.43	0.43	0.25	0.25
Office Flr. C	**SO**	0.67	0.55 ± 0.1	0.47	0.39 ± 0.07		0.59	0.51 ± 0.07	0.41	0.33
	SP	0.61	0.56 ± 0.08	0.47	0.39 ± 0.06		**0.61**	**0.58** ± 0.01	**0.45**	**0.42** ± 0.03
	SOP	**0.74**	**0.63** ± 0.05	**0.64**	**0.54** ± 0.05		0.54	0.45 ± 0.03	0.3	0.26 ± 0.01

Table 3: Performance of our algorithm by using activity observations from Office Flr. A to make predictions in novel scenes. Each baseline method is run with a single parameter setting, and thus their maxes and means are equivalent. The baseline methods "RFC", "Det.", and "NMF" correspond to the Random Forest Classification, Object Detection AMs, and non-regularized NMF augmented matrix approaches, respectively. Variants of our approach, **SO**, **SP**, and **SOP** correspond to using "Spatial+Object Detection" kernels, "Spatial+Scene Classification" kernels, and "Spatial+Object Detection+Scene Classification" kernels. Multiple metrics are considered to observe the effects of ground-truth class imbalance, and means are used to quantify performance across a variety of parameter settings.

`write-whiteboard` and `wash`. This set of activities provides good coverage of common activities that a person can do in an office or home setting. To summarize results, we compute the unweighted and weighted averages of per-class $F1$ scores, where the weighted average is computed by using the normalized counts of the GT classes in the images.

3.2. Preprocessing and parameters

The first step to build the AM is to build a physical map of the environment. We use Structure-From-Motion (SFM) [19] with egocentric videos of a walk through of the environment to obtain a 3D reconstruction of the scene. Next, we consider two important categories of detectable actions: (1) those that involve the user's hands (gesture-based activities), and (2) those that involve significant motion of the user's head, or egomotion-based activities. We used the deep network architecture inspired by [17] to perform ac-

tivity detection, as the two stream network takes into account both appearance (*e.g.*, hands and objects) as well as motion (*e.g.*, optical flow induced by ego-motion and local hand-object manipulations). When actions are detected by our action recognition module, we need a method for estimating the location of this action. We use the SFM model to compute the 3D camera pose of new images.

As we define an AM over a 2D ground plane (floor layout), we project the 3D camera pose associated to an action to the ground plane. To obtain a ground plane estimate, we fit a plane to a collection of localized cameras using SFM. We assume that the egocentric camera lies approximately at eye level, thus this *height plane* is tangent to the top of the camera wearer's head. We then translate this plane along its normal, while iteratively refitting planes with RANSAC to points in the SFM model. Once we have an estimate of the 2D ground plane in 3D space, we can use it to project

(a) Office Flr. A GT

(b) Office Flr. B GT

(c) Office Flr. D GT
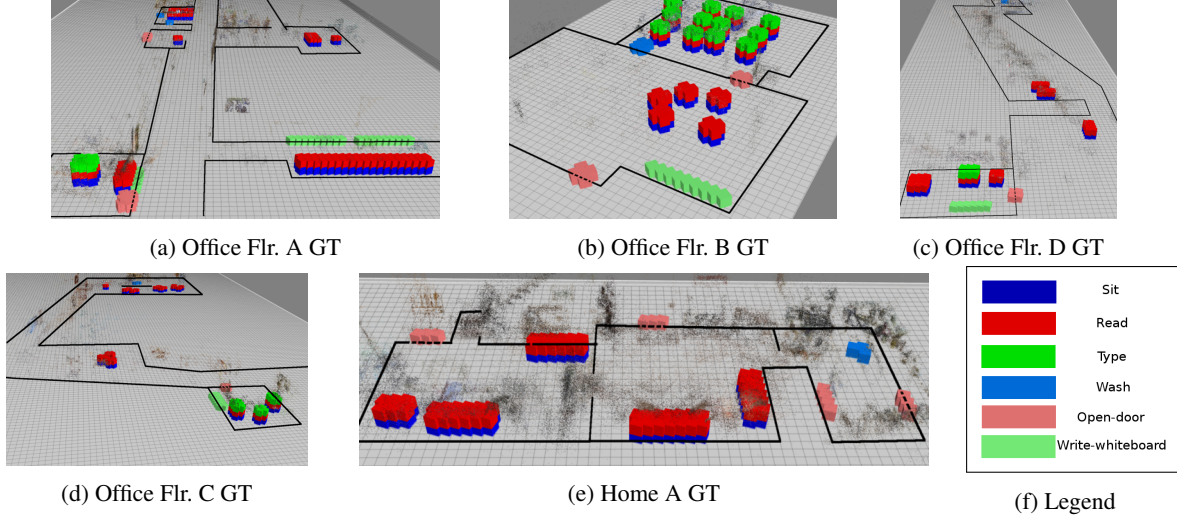
(d) Office Flr. C GT

(e) Home A GT

(f) Legend

Figure 5: Ground truth labels and SFM points in each scene. Dotted lines indicate a doorway, solid lines indicate walls.

the localized actions onto the ground plane. When dealing with multiple scenes, distances must be calibrated between them. We use prior knowledge of the user's height to form estimates of the absolute scale of each scene. Specifically, we use the distance between the ground plane and the user height plane, along with a known user height, to convert distances in the reconstruction to meters. Finally, we grid each scene with cells of size 0.25 meters. (we use a radius of 2 grid cells, which is ~ 0.5 meters after metric estimation).

Since actions are often strongly correlated with the surrounding area and objects, as shown in Figure 3, we also extract the visual context of each action as a source of side-information. For every image obtained with the wearable camera, we run scene classification and object detection with [22] and [6]. We use the pre-trained "Places205-GoogLeNet" model for scene-classification, which yields 205 features per image, one per each scene type, and a radius of 2 grid cells inside which to average the classification scores. For object detection, we use the pretrained "Bvlc_reference_rcnn_ilsvrc13" model, which performs object detection for 205 different object categories, and use NMS with overlap ratio 0.3, and min detection score 0.5.

We use a small grid of parameters for our method ($\alpha \in [0, .1, .3, .5, .7, .9, 1]$, $\lambda \in [10^{-3}, 10^{-2}]$, $\gamma \in [100, 1000]$), where each γ is used for the χ^2 kernels, and evaluate performance of multiple runs as the cross-run maximum and cross-run average of each of the various scores. In a scenario with many additional test scenes, a single choice of parameters could be selected via cross-validation. We also consider variations of our kernel that use different combinations of side-information: Spatial+object detection (SO), Spatial+scene classification (SP), and Spatial+object detection+scene classification (SOP). In the first two cases, the $\frac{\alpha}{2}$ weight of Equation 1 becomes α for the object detection or scene classification kernel that is on, and 0 for the other.

3.3. Full observation experiments

When all activity observations are available, our method is able to perform quite well. The dominant source of error is that of camera localization, which reduces the spatial precision of the AM. In Table 2, we evaluate the performance of our method run on each scene separately, as well as running once with all of the scenes in a single matrix. When multiple scenes are used, side-information is crucial: without it, there is no similarity enforced across scenes. In single scene case, we find that using a spatial kernel only can perform well, yet is generally outperformed by using all side information, especially when side information and activity demonstrations are present from other scenes. By using the data from all scenes simultaneously in a global factorization, performance increases globally over using each single scene's data alone. This is expected and desirable: simultaneous understanding of multiple scenes can improve as the set of available scenes with observation data grows.

3.4. Partial observation experiments

We expose our algorithm to various fractions of the total activity demonstrations to simulate an increasing amount of observed actions. We find that performance is high even with only a few demonstrations and steadily increases as the amount of activity demonstrations increases. The Office Flr. A, Office Flr. D, and Home A scenes have enough activity demonstration data to illustrate the performance gains of our method as a function of the available data. We show quantitative per-class results for these in Figure 4. Sharp increases can be observed in the per-class trends, which correspond to the increase of coverage of each activity class. In Figure 6, we show the overhead view of the AM for the sit and type labels for the Office Flr. A as a function of the available data, where it can be seen how the AM quali-

tatively improves over time as observations are collected.

3.5. Novel scene experiments

Another scenario is the task of predicting AMs for novel scenes containing *zero* activity observation data. Our method leverages the appearance and activity observation data in one scene, and only appearance data in the novel scene to make predictions. We now introduce three baselines we consider. The first baseline is to perform per-image classification with the object detection and scene classification features, which serves to estimate image-wise performance of using the object detection and scene classification information. This baseline requires observations in a labelled scene for training. We use Random Forests [1] as the classification method, trained on images from the source scene. The second baseline we consider is non-regularized Weighted Nonnegative Matrix Factorization by augmenting the target matrix $\mathbf{R}$ with the object detection and scene classification features for each location. This baseline does not explicitly enforce the similarity that the regularized framework does, thus, we expect it to not perform as well as our framework. The third baseline we consider is to build AMs from the back-projected object detections by directly associating each detection category with an activity category.

We use the Office Flr. A demonstration and appearance data as input and evaluate the performance by applying the learned model to each of the other scenes. These results (Table 3) illustrate that our method's AM predictions outperform the baselines in $\frac{13}{16}$ cases, and that the appearance information is capitalized upon the most by our method. We find that scene classification is particularly beneficial to performance, a phenomenon for which we present two hypothesized factors: 1) as shown in [21] "object detectors emerge in deep scene CNNs", suggesting that the Scene Classification features subsume the cues present in the object detector features, and 2) due to localization noise, correlations between localized activities and localized objects are not as strong, and can serve to introduce noise to the Spatial+Scene Classification kernel combination when this object information is integrated.

Overall, we find that our model harnesses the power of activity observations in concert with the availability of rich scene classification and object detection information to estimate the functionality of environments both with and without activity observations. See the Supplementary Material for additional novel scene experiments.

4. Action Maps for Localization

We demonstrate a proof-of-concept application of Action Maps to the task of localization. Intuitively, by leveraging the "where an activity can be done" functional-spatial information from Action Maps, along with "what activity has been done" functional information from activity detection, the user's spatial location is constrained to be in one
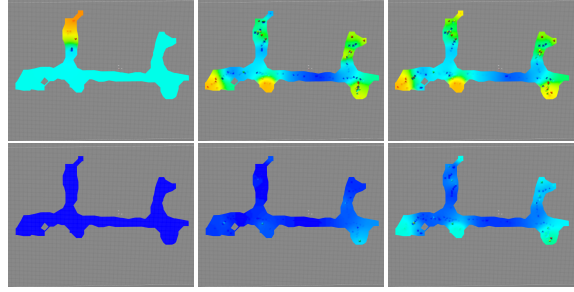

Figure 6: 'Sit' (top row) and 'Type' (bottom row) AMs as the amount of observed data increases on Office Flr. A. The columns stand for 10%, 80%, and 100% of the data.
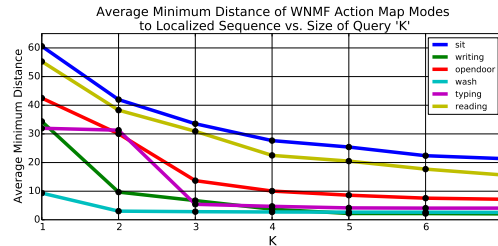

Figure 7: Localizing with an Action Map and observed activities. Activities that are more specialized are localized with less guesses.

of several areas. We localize activity sequences in each 2D map based on the combination of predicted action locations from the Action Map, and observed actions in each frame. In Figure 7, we show the spatial discrepancy in grid cells between the K-best AM location guesses decreases. Thus, an Action Map can be used to localize a person with observations of their activity.

5. Conclusion

We have demonstrated a novel method for generating functional maps of uninstrumented common environments. Our model jointly considers scene appearance and functionality while consolidating evidence from the natural vantage point of the user, and is able to learn from a user's demonstrations to make predictions of functionality of less explored and completely novel areas. Finally, our proof-of-concept application hints at the breadth of future work that can exploit the rich spatial and functional information present in Action Maps.

References

[1] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 8

[2] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *Computer Vision–ECCV 2012*, pages 284–298. Springer, 2012. 2, 3

[3] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 407–414. IEEE, 2011. 3

[4] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: Human actions as a cue for single-view geometry. In *Proc. 12th European Conference on Computer Vision*, 2012. 2, 3

[5] J. Gall, A. Fossati, and L. Van Gool. Functional categorization of objects using real-time markerless motion capture. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1969–1976. IEEE, 2011. 2, 3

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014. 2, 7

[7] Q. Gu, J. Zhou, and C. H. Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210. SIAM, 2010. 2, 5

[8] A. Gupta, T. Chen, F. Chen, D. Kimber, and L. S. Davis. Context and observation driven latent variable model for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 3

[9] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *Computer Vision and Pattern Recognition(CVPR)*, 2011. 2, 3

[10] Y. Jiang, H. Koppula, and A. Saxena. Hallucinated humans as the hidden context for labeling 3d scenes. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2993–3000. IEEE, 2013. 2

[11] H. S. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32(8):951–970, 2013. 2

[12] Y. Li, Z. Ye, and J. M. Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295, 2015. 3

[13] D. J. Moore, I. Essa, M. H. Hayes III, et al. Exploiting human actions and object context for recognition tasks. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 80–86. IEEE, 1999. 3

[14] P. Peursum, G. West, and S. Venkatesh. Combining image regions and human activity for indirect object recognition in indoor wide-angle views. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 82–89. IEEE, 2005. 3

[15] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012. 3

[16] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, and M. Nießner. Scenegrok: Inferring action maps in 3d environments. *ACM Transactions on Graphics (TOG)*, 33(6), 2014. 3

[17] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 6

[18] E. H. Spriggs, F. De la Torre Frade, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *IEEE Workshop on Egocentric Vision, CVPR 2009*, June 2009. 3

[19] C. Wu. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 127–134. IEEE, 2013. 6

[20] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 17–24. IEEE, 2010. 3

[21] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 8

[22] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014. 2, 7