Combining Embedded Accelerometers with Computer Vision for Recognizing Food Preparation Activities

Sebastian Stein and Stephen J. McKenna
CVIP, School of Computing
University of Dundee
Dundee, United Kingdom
{sstein,stephen}@computing.dundee.ac.uk

ABSTRACT

This paper introduces a publicly available dataset of complex activities that involve manipulative gestures. The dataset captures people preparing mixed salads and contains more than 4.5 hours of accelerometer and RGB-D video data, detailed annotations, and an evaluation protocol for comparison of activity recognition algorithms. Providing baseline results for one possible activity recognition task, this paper further investigates modality fusion methods at different stages of the recognition pipeline: (i) prior to feature extraction through accelerometer localization, (ii) at feature level via feature concatenation, and (iii) at classification level by combining classifier outputs. Empirical evaluation shows that fusing information captured by these sensor types can considerably improve recognition performance.

Author Keywords

Activity recognition, sensor fusion, accelerometers, computer vision, multi-modal dataset

ACM Classification Keywords

I.5.5 Pattern Recognition: Applications; I.4.8 Scene Analysis: Sensor Fusion; I.2.10 Vision and Scene Understanding: Video Analysis; K.4.2 Social Issues: Assistive Technologies for Persons With Disabilities

General Terms

Algorithms, Documentation, Experimentation, Measurement, Performance.

INTRODUCTION

We aim to stimulate research in the area of complex activities that involve manipulative gestures, as occurring frequently in food preparation, manufacturing and assembly tasks. Therefore, we provide a carefully designed dataset taking a novel approach to multi-modal sensing: video data and data from accelerometers attached to objects were recorded simultaneously (see Figure 1).

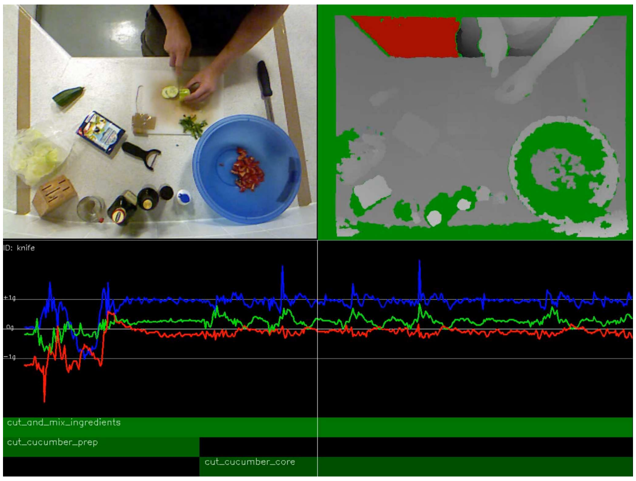


Figure 1. Snapshot from the dataset. Data from an RGB-D camera and from accelerometers attached to kitchen objects were recorded while 25 people prepared two mixed salads each. Activities were split into preparation, core and post-phase, and these phases were annotated as temporal intervals.

One potential application of recognition methods evaluated on this dataset is cognitive situational support. Ubiquitous computing has great potential in addressing the challenges of an aging population by providing automatic situated support for people with cognitive impairments [7, 8, 20]. Enabling people with, e.g., dementia to stay longer in their homes and to perform activities of daily living (ADL) independently of a social carer arguably increases their perceived quality of life [27] and significantly reduces the social cost associated with cognitive impairments. Food preparation is one of the most essential ADL tasks. Automatically recognizing food preparation activities is extremely challenging for various reasons. Usually a recipe involves a large number of complex interactions between hands, utensils, ingredients, etc. in a constrained but non-unique order and with personal variations. In a situational support system actions must be recognized online, imposing strong constraints on computational cost and requiring temporal segmentation of activities, a hard problem in itself. Modelling and tracking activities at a detailed level and issuing sensible prompts to the user are further open research problems. Despite some characteristics that are unique to processing food, many of the challenges faced in this context re-occur in the context of tracking and guiding a person through other complex pro-

To appear in the Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2013).

cesses such as manufacturing and assembly tasks. The study of recognizing food preparation activities has therefore huge potential to push the boundaries of automatic activity recognition in general.

While RFID and embedded accelerometers are inconspicuous and cost-effective they do not provide sufficient data to reason about complex interactions between multiple tagged objects and about their interaction with untaggable objects such as food. Computer vision may be used to establish spatial relationships and enable reasoning about the interaction between visual entities, but recognizing and tracking objects under illumination changes, occlusion, intra-class appearance variations and object deformations are challenging problems. The combination of complementary information from multiple sensor types helps to address some of these issues.

In the ubiquitous computing research community, video is mainly recorded to enable manual annotation of the *actual* sensor data from, e.g., RFID readers, accelerometers and gyroscopes. Often video is not recognized as a valid source of information due to the widespread belief that cameras are too intrusive and would not be acceptable. Over the past few years the number of cameras per household has grown exponentially through the emergence of smart-phones, tablets, and video-based input devices such as Kinect. While users may object to the idea of another person watching them via CCTV-like installations in their homes they are often accepting of visual sensors as part of a closed system that enables desirable services. In the computer vision community some multi-modal activity recognition datasets have been released, including synchronized video, RFID, audio and IMU data. Interestingly, in all of these datasets accelerometers and gyroscopes have been attached to the subject's body, which is practically inconvenient in the context of situational support systems. In pervasive intelligent environments it is therefore common practice to embed sensors in objects involved in an interaction instead [2].

This paper introduces a dataset including video data and data from accelerometers attached to objects. Design decisions, the experimental setup, annotation and evaluation protocols are discussed in detail. We also provide benchmark results for one particular activity recognition problem on this dataset. We propose to combine video and accelerometer data through accelerometer localization and show that fusing features at early stages of the recognition pipeline significantly increases activity recognition performance. In summary, the contributions of this paper are:

- A multi-modal activity recognition dataset including more than 4.5 hours of annotated accelerometer and RGB-D video data, which is the first of its type.
- A novel method for fusing accelerometers and computer vision for activity recognition.
- Experimental evaluation of the proposed method and comparison of various fusion methods on the new dataset, including a protocol for benchmarking.

RELATED WORK

Existing Datasets

Several public datasets for benchmarking activity recognition algorithms exist in the fields of wearable computing [10, 18, 21, 29] and computer vision [14, 15, 17, 22, 23, 25, 4]. One reason for the multiplicity of datasets is that the terms *activity* and *recognition* are used for varied concepts. In many cases *recognition* means offline classification, where data from an entire video clip is used to determine it's activity class (e.g., KTH [23], YouTube [14], Hollywood2 [15] and URADL [17]). In others, however, *recognition* additionally includes identifying the temporal (and spatial) extent of an action, also referred to as activity detection or spotting (e.g., Darmstadt Daily Routines [10], AmbientKitchen [18], TUM Kitchen [25], CMU-MACC [4], Opportunity [21], ICPR-KSCGR[1] and MPII-Cooking [22]). Datasets supporting activity spotting have the benefit that they can also be used purely for classification. The dataset presented in this paper includes continuous sequences of complex interactions involving multiple objects that can be used for classification, activity spotting and progress tracking.

The term activity is used even more broadly and may refer to atomic gestures (e.g., grasp) [21], simple, repetitive, articulated full body motions [23], fine-grained hand gestures [18], or complex inter-actions of multiple objects [10]. Actions are also described with varying level of detail and may contain only a verb (e.g., boxing, waving, cutting, adding) or may additionally include the objects interacted with (e.g, picking up cafeteria food) [10]. Furthermore, the total set of activities considered in a dataset may be very broad (e.g., actions in movies [15] or web video clips [14]) or scenario-specific (e.g., car assembly-line checkpoint [29] or food preparation [18]). This choice affects inter-class variability. A classification problem with high inter-class variability and low intra-class variability is easier than one with low inter-class variability and high intra-class variability. Our dataset contains activities with low inter-class variability and high intra-class variability as well as detailed activity annotations including the identities of objects involved (e.g., *place tomato into bowl*).

Recently several datasets of kitchen activities have been released that combine visual and non-visual sensor types. The CMU-MMAC [4] contains data from multiple cameras and body worn IMUs, BodyMedia and an eWatch. The cameras are placed to overlook large parts of the kitchen as in a surveillance scenario. Participants wear a suit with IMUs placed at joint locations and a helmet with an attached camera. Such a sensor placement is arguably infeasible for practical assisted living solutions as the camera positioning is obtrusive and wearing the suit is strongly disruptive. The TUM Kitchen dataset [25] contains video and RFID data of people dressing a table. Cameras were placed similarly to the CMU-MMAC dataset and RFID sensors were embedded at three locations in the kitchen. Although participants did not have to wear a sensor-suit in this scenario, the camera positioning was intrusive. The activities captured in this dataset

[1] `http://www.murase.m.is.nagoya-u.ac.jp/KSCGR/index.html`

did not involve objects that are untaggable with RFID. One major challenge for recognizing food preparation tasks is, however, that sensors cannot be attached to food. The use of RFID readers in the kitchen is also very limited due to the fact that there is no small number of distinct strategic locations that would suffice to be equipped with RFID readers.

Pham et al. [18] replaced the handles of kitchen utensils with Wii-controllers capturing tri-axial accelerometer data. We extended their approach by attaching accelerometers to various kitchen objects and by combining these sensors with an RGB-D camera facing the work-surface. Such a sensor setup is affordable and feasible to integrate into a home kitchen.

Activity Recognition

In both computer vision and ubiquitous computing communities, researchers have been shifting focus from recognizing simple *actions* that correspond to distinct motion patterns to assessing the quality of execution (e.g., skill) and recognizing complex, broadly defined activities. The dataset provided with this paper facilitates research in the latter direction in that it comprises annotated sequences of complex multi-step activities. Additionally, important factors for activity recognition to find its way into real world applications are recognition at low computational cost, recognition of ongoing activities and activity prediction.

Activity recognition has a long history in computer vision research, recently reviewed by Aggarwal and Ryoo [1]. Currently the most prominent approaches extract local appearance and motion descriptors in the neighborhood of spatio-temporal interest points [13] and point trajectories [26] followed by discriminative classification via support vector machines. While these methods show good performance on after-the-fact (offline) classification of articulated full body motion and actions in movies, Rohrbach et al. [22] show empirically that this approach is not well suited for recognizing food preparation activities. In addition to yielding low recognition performance, extracting local descriptors around densely sampled locations as proposed in [26] is computationally demanding in real-time, e.g. at 30 Hz. It is therefore crucial to investigate methods for activity recognition that are faster and better suited to recognizing food preparation activities for delivering situated guiding prompts.

For an overview of features proposed for accelerometer-based activity recognition we refer the interested reader to the survey by Figo et al. [5]. Pham et al. [18] developed a method for recognizing food preparation actions such as chopping, peeling, stirring and scooping using accelerometer data from accelerometers embedded in knifes and spoons. They extracted statistical features (mean, energy, variance and entropy) in the time domain and computed pitch and roll for encoding device orientation. Recently, Plötz et al. [19] proposed feature learning for activity recognition from accelerometers with deep belief networks. As feature learning is costly and Pham et al. [18] reported good performance with features that are very fast to compute, we use their set of features for our analysis. The conclusions we draw regarding the benefit of combining different sensor types based on these features are expected to be similarly valid for other features.

Activity	#Inst.	#Frames	Core
add oil	55	27813	8161
add vinegar	54	23657	6572
add salt	53	11369	4995
add pepper	55	12912	6123
mix dressing	61	19492	14295
peel cucumber	53	62141	38613
cut cucumber	59	49853	38787
cut cheese	56	50680	26001
cut lettuce	61	53313	28847
cut tomato	63	68347	50768
place cucumber into bowl	59	16800	9071
place cheese into bowl	53	12753	6305
place lettuce into bowl	61	15159	7856
place tomato into bowl	62	13547	6418
mix ingredients	64	22917	16050
serve salad onto plate	53	35230	19110
add dressing	44	22428	12092
Total	**966**	**518411**	**300064**

Table 1. Dataset size in terms of activity instances and video frame counts.

Combining various feature types and classification results has been widely studied. Prominent approaches include multiple kernel learning as a feature combination method for discriminative classification [16], and the sum-rule and product-rule for classifier combination [11]. Wu et al. [28] combine visual features with data from an RFID reader attached to a person's wrist in a dynamic Bayesian network for distinguishing between high-level activities in the kitchen. Video and IMU data from mobile phones are combined in [9] for indoor localization. Combining embedded accelerometers and video has huge potential for activity recognition by providing complementary information. The localization of accelerometers in the visual field as proposed in [24] is one method for efficiently fusing these modalities. In this paper we provide first empirical evidence using this method to support recognition.

FOOD PREPARATION DATASET: 50 SALADS

We collected a dataset[2] comprising annotated video and accelerometer data of people preparing a mixed salad. We first give a description of the experimental setup, participants, sensor synchronization and activity annotation before we motivate the design decisions.

Dataset Details

An RGB-D camera (Kinect) was mounted on the wall to have a top-down view onto the work surface. Accelerometers were embedded in the handles of a knife, a mixing spoon and a peeler. Further accelerometers were attached to a small spoon, a glass, an oil bottle, and a pepper dispenser. We recorded visually aligned RGB and depth data with 640x480

[2] Dataset URL: `http://cvip.computing.dundee.ac.uk/datasets/foodpreparation/50salads/`

pixels resolution at 30Hz. We used Axivity WAX3 wireless accelerometers which are equipped with a rechargeable battery, microprocessor, 3-axis accelerometer, IEEE 802.15.4-2006 radio and a micro-USB port for recharging and configuration. These devices transmit acceleration data at 50Hz with 16-bits per axis resolution. All samples are timestamped upon arrival at the server. We chose not to use gyroscopes as they significantly reduce battery life and their data exhibit strong artifacts resulting from magnetic interference with kitchen equipment. Figure 1 shows an example snapshot from the dataset.

We recruited 27 subjects of varied age, ethnic background and cooking experience. All subjects prepared a mixed salad two times totalling 54 sequences. Two subjects had to be excluded from the final dataset due to data loss. Preparing the mixed salad involved preparing a dressing with salt, pepper, olive oil and balsamic vinegar, cutting ingredients (cucumber, tomato, feta cheese and lettuce) into pieces, mixing ingredients, adding the dressing to the salad and serving the salad onto a plate. Participants were given a specific task order to follow in each run. They were also told to perform all activities within a fixed area on the work surface delimited with tape, that marked the border of the camera's field of view. While no specific quantities for ingredients were given, participants were asked to prepare a single portion of salad for one person.

For sensor synchronization we performed an action that simultaneously produces strong signals in the video and the accelerometer data at the start and the end of all sequences. By establishing correspondences within these signals we estimated two temporal offsets per sequence, one for the start and one for the end. We used linear interpolation for temporal alignment within this interval.

The following activities were annotated in the form of a start time and an end time corresponding to their temporal extent: *add oil*, *add vinegar*, *add salt*, *add pepper*, *mix dressing*, *peel cucumber*, *cut cucumber*, *place cucumber into bowl*, *cut cheese*, *place cheese into bowl*, *cut lettuce*, *place lettuce into bowl*, *cut tomato*, *place tomato into bowl*, *mix ingredients*, *serve salad onto plate* and *add dressing*. Each activity was split into three phases which were annotated individually: *pre-*, *core-* and *post-phase*. Each activity was associated with one of three stages in the recipe which were also annotated: *prepare dressing*, *cut and mix ingredients* and *serve salad*. In total 966 activity instances were annotated. Annotations spanned more than 500k video frames of which more than 300k frames represented the core-phase of an activity. Table 1 lists the numbers of instances and frames for all activities.

Motivation for Design Decisions

Repeated Task Execution

We asked all participants to prepare a salad twice for two reasons. First, people exhibit varying degrees of disorientation while preparing food if they are not within their usual kitchen setting and without access to their own utensils. A laboratory kitchen setup may add to that further through the knowledge of being recorded. Therefore, we expect subjects to behave more naturally in the second session as they had time to get used to the laboratory kitchen in the first run. Secondly, recording subjects multiple times enables the study of idiosyncrasies and the comparison of different learning scenarios, e.g., same subject included in training data against cross-subject generalization.

Task Order Sampling

When observing a person performing multi-step activities interacting with a number of different objects, different orderings in which these steps are carried out induce strong variation on the configuration and appearance of objects in the scene. In the context of preparing a mixed salad, for example, the scene looks different after preparing the dressing, depending on whether the ingredients of the salad have been cut and mixed already. In order to build robust activity models for recognition it is convenient to have a balanced dataset that contains roughly the same number of examples for all likely task-orderings. In practice it is costly to acquire annotated video data of a large number of people performing the same multi-step activity. Additionally, the task-orderings the recorded sample population chooses naturally are potentially highly imbalanced. Therefore, we propose to sample task-orderings from a statistical activity model and ask participants to follow the steps of a recipe in orderings generated by the model. The statistical activity model for preparing a mixed salad we used is illustrated in Figure 2. The model is based on *Activity Diagrams* used in computational process specification and analysis. Every choice-node of the diagram (represented by a horizontal bar with multiple outgoing arcs) is augmented by a probability distribution over all options representing the probability of choosing each option when that choice-node is reached. The probabilities in Figure 2 are set to be uniform to ensure a balanced distribution of task-orderings.

Each participant was given a different ordering of tasks to follow in each session. Surprisingly, few participants precisely followed the task ordering given to them although they were specifically instructed to do so. Our hypothesis is that ordering tasks within a food preparation activity is strongly governed by habit and personal reasoning. In cases where subjects failed to correctly follow the instructions in the second session, this error may also be due to the subject following the memorized task ordering of the previous session. Although this behavior was unintended, the availability of the instructed task ordering together with the annotated activities may be used to experiment with detecting deviations from the given task order.

Annotation

The pre- and post-phases of an activity include grabbing, moving and placing utensils and ingredients. The core phase captures actions that are essential. Taking *add oil* as an example, the pre-phase might consist of grabbing the oil bottle, moving it over the dressing glass and screwing it's lid off. The core phase represents tilting the bottle and pouring oil into the glass. Screwing the lid back on, moving the bottle and placing it on the work surface would be annotated as the

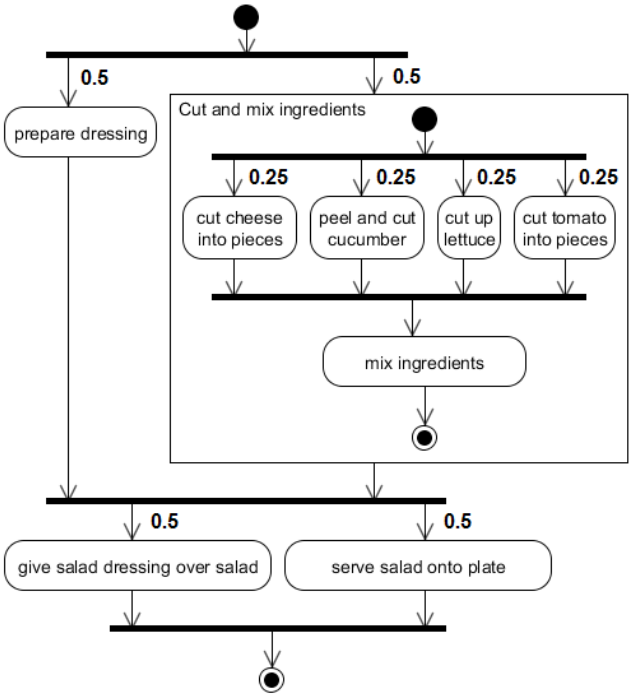


Figure 2. Activity Diagram: Task orderings were sampled from this model and given to participants in order to increase variability of task orderings in the dataset.

post-phase of the *add oil* activity. The annotated temporal extent of a phase of an activity is delimited by distinct events at the start and end of these phases. The start of the pre-phase of the *add oil* activity, for example, would be marked as the frame in which the oil bottle was first touched as opposed to the frame in which the hand reaches out in order to grab the bottle. Annotations are therefore unambiguous and repeatable.

Use Cases

This dataset can be split in various ways into training and test data in order to investigate different generalization problems. Cross-subject generalization is a common problem investigated by the activity recognition community, where all data of any subject is used either for training or for testing. This is a hard problem in the context of food preparation due to strong personal preferences regarding how activities are executed (idiosyncrasies). Intra-subject generalization is a comparably easier problem given the same amount of training data. Here, only very limited data of a single subject is available. Nevertheless it would be interesting to evaluate intra-subject generalization on this dataset as having very limited training data of the target subject might be a good representation of real-world conditions for, e.g., a situated prompting system.

While inferring the full specification of an activity (verb, phase and objects involved) is our long-term objective, we recognize that current activity recognition methods are not powerful enough to do this. In order to gradually approach this goal, however, the available detailed annotations may be used to automatically formulate easier classification problems. For example, parts of the description such as the ingredients or the activity-phase may be ignored, mapping distinct annotations onto the same label. One such simplified recognition problem is discussed in the following Section.

MULTI-MODAL ACTIVITY RECOGNITION

Task Description and Evaluation Protocol

The choice of ontology by which activities are categorized can substantially influence recognition performance. Consider, for example, the two activities mixing the salad dressing and mixing the completed salad. They may be regarded as belonging to the same general activity of mixing ingredients. The exhibited motion pattern when performing these activities is similar. Based on low-level motion features alone it would therefore be difficult to differentiate between mixing the dressing and mixing the completed salad, and combining these activities in the general activity *mixing ingredients* simplifies the recognition task. However, the kitchen utensils used while performing these two activities are different. While participants tend to move the dressing glass and the small spoon when mixing the dressing, the large spoon and the salad bowl are moved when the final salad is being mixed. As the sets of utensils moved in these two activities are mutually exclusive, the task of differentiating these activities based on object use is comparably easy. Therefore, assigning a common class label to these activities renders the recognition task more difficult. With these considerations in mind we investigate the problem of recognizing the following activity classes based on various combinations of feature types: *add_oil, give_pepper, mix_dressing, mix_ingredients, cut_into_pieces, place_into_bowl, peel_cucumber, serve_salad, dress_salad* and *NULL*, where *NULL* indicates that none of the activities of interest is happening. This is of course only one of many recognition problems that could be investigated using this dataset.

For activity recognition we assume that no two activities occur simultaneously. In rare cases the annotated temporal intervals of subsequent activities overlap. As the frames in which this situation occurs only account for 0.09% of the data, we skip these samples for both training and testing, rendering the recognition task a pure classification problem.

In order to test cross-subject generalization, we evaluate algorithms by 5-fold cross-validation. Although 10-folds are often used (as argued in [12]), we chose 5-fold cross-validation to keep the computational cost manageable. We split the dataset into 5 partitions each containing both sessions of 5 subjects. Each of these partitions is used for testing an algorithm that has been trained on the remaining 20 subjects. Model selection for each tested algorithm is performed via 5-fold cross-validation on each training set. The training set for each test partition is split into 5 partitions containing both sessions of 4 subjects. Each of these partitions is used for validating a model that has been trained on the remaining 16 subjects.

For comparatively evaluating different algorithms, recognition performance is measured as mean precision and mean recall. Mean precision and mean recall are calculated across five cross-validation partitions for each activity class and the arithmetic mean across all activity classes is taken to produce the final result. Given the number of true positive (TP), false positive (FP) and false negative (FN) classification results, precision is defined as $\frac{TP}{TP+FP}$ and recall is defined as $\frac{TP}{TP+FN}$. Mean precision and recall over all classes and cross-validation partitions are computed by first summing TP, FP and FN over all partitions for each class separately, then applying the formulas for precision and recall on the sums and finally estimating the arithmetic mean over all classes, as argued in [6]. The arithmetic mean over all classes assigns equal importance to all classes regardless of their prevalence in the test data.

Features

We consider features extracted from accelerometer data (*Object Use* and *Acceleration Statistics*) and features constructed by visually localizing accelerometers (*Device Locations* and *Visual Displacement Statistics*).

Accelerometer Localization

In order to use visual accelerometer localization for activity recognition we adapt the approach proposed in [24]. Accelerometers are localized in the visual field of a camera by matching a device's measured acceleration $\mathcal{A}_{dev} : (\mathbf{a}_{dev}^{(0)}, \ldots, \mathbf{a}_{dev}^{(t)})$ to the acceleration $\mathcal{A}_{vis}^{s} : (\mathbf{a}_{s}^{(0)}, \ldots, \mathbf{a}_{s}^{(t)})$ estimated along visual point trajectories $\mathcal{P}_s : ((x_s^{(0)}, y_s^{(0)}), \ldots, (x_s^{(t)}, y_s^{(t)}))$. The location in the most recent frame of the visual trajectory $\mathcal{P}_{\hat{s}}$ with strongest similarity is taken to be the device's location estimate: $(\hat{x}_{dev}^{(t)}, \hat{y}_{dev}^{(t)}) = (x_{\hat{s}}^{(t)}, y_{\hat{s}}^{(t)})$. The similarity between $\mathcal{A}_{dev}$ and $\mathcal{A}_{vis}^{s}$ is estimated incrementally with a temporal decay α:

$$\begin{aligned} S_t(\mathcal{A}_{dev}, \mathcal{A}_{vis}^{s}) = & 1\left[|\mathbf{a}_{s}^{(t)}| \geq T_{loc} \wedge |\mathbf{a}_{dev}^{(t)}| \geq T_{loc}\right] \\ & + \alpha \cdot S_{t-1}(\mathcal{A}_{dev}, \mathcal{A}_{vis}^{s}) \end{aligned} \quad (1)$$

This method does not require any learning and provides good localization results for devices that exhibit strong acceleration. For S_t to differ significantly from S_{t-1} the device acceleration $|\mathbf{a}_{dev}^{(t)}|$ has to exceed T_{loc}. Therefore, the location estimates $(\hat{x}_{dev}^{(t)}, \hat{y}_{dev}^{(t)})$ drifted away from the target when the device measured no acceleration and the point trajectory $P_{\hat{s}}$ changed its location. This becomes problematic when the device does not move and $P_{\hat{s}}$ tracks the motion of other objects in the scene.

The modification we propose here detects when a device is stationary and temporarily stores the locations $(x_s^{(t)}, y_s^{(t)})$ of all point trajectories P_s together with the similarity values $S_t(\mathcal{A}_{dev}, \mathcal{A}_{vis}^{s})$. When the device is detected to move again the similarity of a point trajectory is initialized with the value $S_t(\mathcal{A}_{dev}, \mathcal{A}_{vis}^{s})$ corresponding to the closest location $(x_s^{(t)}, y_s^{(t)})$ in the current frame.

Object Use

We assume that an object is in use if and only if it is moving. The wireless accelerometers used in our experiments stop data transmission if the measured magnitude of acceleration does not exceed a threshold over a fixed number of consecutive samples N^*. Although this method only detects constant velocity, it is unlikely that a human performs a movement with constant velocity over an extended period of time. We therefore consider an object to be not moving if the accelerometer attached to it does not send any data. With this approach *not moving* is detected with a delay of $N^* - 1$ samples. Using a generalized formulation we can estimate whether an object is moving with shorter delay. Let $\mathcal{A} : (\mathbf{a}^{(0)}, \ldots, \mathbf{a}^{(t)})$ be a sequence of accelerometer data up to time t, N' the number of considered consecutive acceleration samples, $\mathbf{g}$ the gravitational acceleration and T_{mov} a threshold. Whether an accelerometer is moving at time t can then be formally expressed as

$$moving(\mathcal{A}, t) = \neg\left(\bigwedge_{j=t-N'+1}^{t} (|\mathbf{a}^{(j)}| - |\mathbf{g}| \leq T_{mov})\right) \quad (2)$$

$N' = N^*$ in the experiments reported here. Preliminary evaluation results have shown that recognition performance does not change significantly with $N' \neq N^*$.

Acceleration Statistics

Following the approach of Pham et al. [18] who experimented with various classifiers in the context of recognizing food preparation actions involving four utensils, we extract the statistical features mean, energy, standard deviation and entropy for each of the three axes and estimate pitch and roll from four temporal subwindows. These subwindows have a length of 32 samples each and are evenly spaced within the temporal window. Pitch and roll encode the device's orientation relative to the direction of gravity and can be estimated from accelerometer data because the data represents proper acceleration (relative to free fall). The yaw angle cannot be recovered from accelerometer data because the yaw angle describes rotation around the axis that is aligned with the direction of gravity. The concatenation of these features results in a vector of 20 dimensions for each device. We concatenate the feature vectors extracted from all devices, resulting in a feature vector with 140 dimensions.

Device Locations

Assuming that accelerometers are attached to objects that participate in an activity of interest their visual trajectories are likely to be distinctive for the activity that is performed with those objects. Therefore, we propose to visually localize accelerometers and use accelerometer locations and trajectories as features for activity recognition. In contrast to visual object tracking which is an extremely challenging problem in itself, localizing accelerometers attached to objects enables object tracking without making assumptions about an object's appearance. Using the accelerometer localization algorithm, we construct a feature vector for each video frame containing the estimated 2D location of each device in camera coordinates.

Visual Displacement Statistics

We estimate statistical features (mean, energy, standard deviation, entropy) for the visual displacement components, $(\Delta x, \Delta y)$, of the point trajectory that is matched to an accelerometer using the accelerometer localization algorithm. These features, extracted from a fixed temporal window of 16 video frames along all accelerometer trajectories, are concatenated to form a single feature vector characterising the visual motion of all devices. The resulting feature vector has 56 dimensions (8 features per device for 7 devices).

Classification

In this paper we consider the naive Bayes and the random decision forest classifiers. Naive Bayes is used to generate baseline recognition results to which more complicated classifiers can be compared. Random decision forests are nonlinear classifiers that naturally extend to multi-class classification and produce well calibrated posterior probabilities. These characteristics make this classifier favorable compared to the popular support vector machine (SVM) [3].

Naive Bayes Classifier

Naive Bayes assumes independence of the observations $\mathbf{o} : (o_1, \ldots, o_K)^T$ conditioned on the class. MAP classification selects the class that maximizes the posterior:

$$P(c|\mathbf{o}) \propto P(\mathbf{o}|c)P(c) \approx P(c)\prod_{k=1}^{K} P(o_k|c) \qquad (3)$$

This assumption is often poor, e.g., a glass and a spoon are likely to move simultaneously when mixing a salad dressing. Probabilities are modeled with binomial distributions for *Object Use* and with Gaussian distributions for all other features and feature combinations.

Random Decision Forest

A random forest is an ensemble of random decision trees, where each tree is trained in isolation [3]. Each internal node of a decision tree represents a weak classifier in the form of a binary decision function. Starting at the root node, a random subset of the set of weak learners is selected. This random subset of features is evaluated in combination with a small number of randomly selected thresholds against the information gain criterion. The weak classifier with highest information gain on the training data for this node is selected. The weak classifier divides the training data into two partitions. The left and right child node are subsequently trained based on their respective training data partitions. Leaf nodes store the distribution of training samples arriving in a given node over classes. We use axis-aligned weak classifiers, which simply compare the value of a single dimension of a feature vector with a threshold. Using more complex classifiers in the tree nodes is associated with a significantly higher computational cost for training. Preliminary evaluation results showed that more complex classifiers did not improve recognition performance. In order to deal with unbalanced training data, the contribution of samples from different classes to (i) information gain and (ii) class distributions in the leaf nodes are weighted differently. The weight of a sample from class c is set to be inversely proportional to the number of samples n_c from that class in the training set:

$$w_c = max_u(n_u)/n_c \qquad (4)$$

In the limit of an infinite number of training samples this approach is equal to stratification, i.e., selecting an equal number of samples per class.

In the inference stage a feature vector traverses all trees starting at the root node, descending to the next node depending on the evaluation of the weak classifiers in the current node on the test sample. The class distributions of the destination leaf nodes are summed and normalized yielding a posterior distribution over activity classes given the test sample.

The meta-parameters of a random forest specifying (i) the number of decision trees, (ii) the maximum depth of each tree, (iii) the number of randomly selected features tested in each node, and (iv) the number of thresholds tested per feature need to be set prior to forest training. We attempt to find good values for these parameters through model selection. Automatic model selection involves choosing the model that minimizes a loss-function. As our random forests select features for each node to maximize information gain, the cross-entropy error is used as the loss-function for model comparison. The cross-entropy error for a single datapoint is defined as

$$H(p, q) = -\sum_{c} p(c)log_2(q(c)) = -log_2(\hat{p}(c_{gt}|o)), \qquad (5)$$

where $p(c)$ is the ground-truth class distribution (delta-function with peak at true class label c_{gt}) and $q(c)$ is the class posterior $\hat{p}(c|o)$ estimated by the recognition algorithm. In this special case the sum only contains a single non-zero element, which is the log of the estimated probability for the ground-truth class. We compute the per class cross-entropy error and sum over all classes in order to obtain a single performance measure given in (6).

$$H_m = \sum_{c} \frac{1}{n_c} \sum_{i:p_i(c)=1} H(p_i, q_i) \qquad (6)$$

Since cross-entropy is estimated from cross-validation it can be regarded as a random variable with fluctuation around the mean. Treating model-selection as a regression problem with cross-entropy as its error function, we handle the bias-variance trade-off by selecting the model that minimizes $mean(H_m)^2 + variance(H_m)$.

Fusion Methods

In addition to combining information from accelerometers and video data through accelerometer localization to extract *Device Locations* and *Visual Displacement Statistics*, we consider fusion at feature level by concatenating feature vectors (early fusion) and fusion at classifier level (late fusion). For classifier fusion we tried the sum-rule and the product-rule [11] as well as Random Decision Forests as a non-linear combination method.

Evaluation

Following the quantitative evaluation in Pham et al. [18] we use a temporal window of 256 accelerometer samples for estimating *Acceleration Statistics*. For *Visual Displacement Statistics* a temporal window of 16 video frames was used, a trajectory length commonly used for visual action recognition [26].

For random forests model selection we searched for a good model with 20 trees, the values $\{8, 10, 12\}$ for the maximum number of tree levels, $\{\frac{1}{3}d, \frac{2}{3}d, d\}$ for the number of weak classifiers tested per node (where d is the number of feature dimensions) and 10 thresholds per weak classifier.

Results

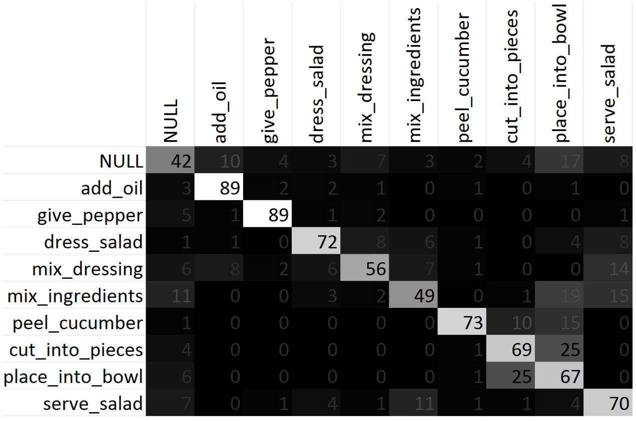
The naive Bayes classifier did not perform better than chance on features other than *Object Use* and *Device Locations*. Using *Object Use* gave a respectable recognition performance of 0.42 ± 0.01 precision and 0.48 ± 0.02 recall, intervals representing one standard deviation. This result indicates that the involvement of objects in an activity regardless of their interactions is already a strong cue for recognizing food preparation activities. Scaling a recognition system beyond a single recipe, however, would drastically increase the number of possible activities involving any single object, rendering this type of feature less informative. With *Device Locations* the Naive Bayes classifier achieved 0.14 ± 0.02 precision at 0.14 ± 0.01 recall. The low performance may indicate that the naive Bayes assumption is particularly poor for this type of feature or that device locations are not discriminative for distinguishing food preparation activities.

The recognition results obtained with various features, fusion methods and Random Forest classifiers are shown in Table 2. The top section shows the recognition performance for individual feature types, including two types that rely on multi-modal fusion using accelerometer localization. The middle section (Combination: Early) shows recognition performance with fusion at feature level, i.e. concatenation of feature vectors. The bottom section shows recognition performance achieved through combining posterior class distributions obtained by classification based on individual feature types.

Applying a random forest classifier to *Object Use* features did not improve recognition performance over the baseline results obtained with the naive Bayesian model (not shown in the Table). This indicates that the naive Bayes assumption is good for this type of feature. Recognition accuracy for *Device Locations* increased compared to naive Bayes classification but was still lowest among all configurations tested with random forest classifiers. This illustrates that objects are positioned quite freely on the work surface and that their locations do not provide strong cues for activity recognition even in this dataset where all sequences were recorded in a single (confined) kitchen setup involving the same utensils. The performance using *Acceleration Statistics* of 0.62 precision and 0.64 recall was the best among all individual

Feature Type	Comb.	Precision	Recall
OU	-	0.41 ± 0.03	0.48 ± 0.02
DL	AL	0.26 ± 0.02	0.22 ± 0.03
VS	AL	0.52 ± 0.05	0.49 ± 0.04
AS	-	**0.62 ± 0.05**	**0.64 ± 0.04**
OU + DL	Early	0.51 ± 0.03	0.51 ± 0.02
OU + VS	Early	0.54 ± 0.02	0.53 ± 0.04
OU + AS	Early	0.63 ± 0.05	0.66 ± 0.03
DL + VS	Early	0.57 ± 0.04	0.54 ± 0.03
DL + AS	Early	0.61 ± 0.05	0.64 ± 0.03
AS + VS	Early	0.67 ± 0.05	0.67 ± 0.03
OU + AS + VS	Early	**0.67 ± 0.05**	**0.68 ± 0.03**
OU + DL	Sum	0.43 ± 0.02	0.49 ± 0.03
	Product	0.45 ± 0.02	0.50 ± 0.02
OU + VS	Sum	0.51 ± 0.03	0.53 ± 0.03
	Product	0.52 ± 0.03	0.53 ± 0.03
OU + AS	Sum	0.43 ± 0.07	0.49 ± 0.03
	Product	0.44 ± 0.02	0.50 ± 0.03
DL + VS	Sum	0.49 ± 0.05	0.51 ± 0.03
	Product	0.52 ± 0.04	0.51 ± 0.04
DL + AS	Sum	0.59 ± 0.05	0.64 ± 0.03
	Product	0.61 ± 0.05	0.64 ± 0.04
AS + VS	Sum	0.63 ± 0.04	0.67 ± 0.03
	Product	0.65 ± 0.03	0.67 ± 0.03
OU + AS + VS	Sum	0.62 ± 0.04	0.65 ± 0.02
	Product	0.64 ± 0.03	0.66 ± 0.03
	RF	**0.65 ± 0.05**	**0.67 ± 0.03**

Table 2. Activity recognition performance (mean precision and mean recall) achieved with various features, fusion methods and Random Forest classifiers. Intervals represent ± one standard deviation. Device Locations (DL) and Visual Displacement Statistics (VS) use multi-modal fusion by accelerometer localization. Early fusion refers to concatenating feature vectors. Sum, product and RF (random forest) indicate classifier combinations by aggregating class posterior distributions. AL: accelerometer localization, AS: Acceleration Statistics, DL: Device Locations, OU: Object Use, RF: Random Forest, VS: Visual Displacement Statistics.



	NULL	add_oil	give_pepper	dress_salad	mix_dressing	mix_ingredients	peel_cucumber	cut_into_pieces	place_into_bowl	serve_salad
NULL	42	10	4	3	7	3	2	4	17	8
add_oil	3	89	2	2	1	0	1	0	1	0
give_pepper	5	1	89	1	2	0	0	0	0	1
dress_salad	1	1	0	72	8	6	1	0	4	8
mix_dressing	6	8	2	6	56	7	1	0	0	14
mix_ingredients	11	0	0	3	2	49	0	1	19	15
peel_cucumber	1	0	0	0	0	0	73	10	15	0
cut_into_pieces	4	0	0	0	0	0	1	69	25	0
place_into_bowl	6	0	0	0	0	0	1	25	67	0
serve_salad	7	0	1	4	1	11	1	1	4	70

Figure 3. Confusion Matrix for the method that achieved highest activity recognition accuracy among all configurations considered in Table 2. A random forest classifier was trained on concatenated Object Use, Acceleration Statistics and Visual Displacement Statistics features (early fusion). Rows and columns represent ground-truth and predicted class labels, respectively. Numbers represent frequencies in percent and cell gray-levels linearly encode frequencies from 0% (black) to 100% (white).

feature types. This result confirms that acceleromter-based activity recognition can yield good performance using features that are fast to extract from the temporal domain and require no learning as proposed in [19]. The lower performance of *Visual Displacement Statistics* compared to *Acceleration Statistics* can be attributed to the shorter temporal window used for feature extraction (0.53s compared to 5.12s for *Acceleration Statistics*) and imperfect localization.

The combination of different features prior to classification (Combination: Early) in Table 2 consistently improved recognition performance compared to the individual feature types, and the observed performance increase was statistically significant in all cases except for the combination of *Device Locations* with *Acceleration Statistics*. These observations strongly support the hypothesis that robust activity recognition benefits from integrating multiple types of cues. Combining classifier outputs using the sum-rule or the product-rule only showed minor improvement compared to the individual feature types. Note, however, that inference in decision forests trained on combined features is faster by a factor K than combining classifier outputs of K feature types.

We also experimented with a non-linear combination of classifier outputs using random forests. Due to the computationally demanding model selection, we only ran these experiments on the combination of features that achieved highest recognition accuracy with early fusion. Here, recognition accuracy is not significantly different from that obtained with early fusion of the same features.

The confusion matrix of test results obtained with a random forest classifier trained on concatenated *Object Use*, *Acceleration Statistics* and *Visual Displacement Statistics* features is illustrated in Figure 3. This configuration achieved highest activity recognition accuracy among all configurations considered in Table 2. For almost all activities recall was above 50%, except for the $NULL$-activity (42%) and $mix\_ingredients$ (49%). Considering the high intra-class variability and the fact that no temporal activity modelling was used these results are very promising. As the pre- and post-phases of activities involve re-organizing objects on the work surface, there is significant confusion between $NULL$ and all other activities. The large spoon was often used to carry out the $mix\_ingredients$ and $serve\_salad$ activities. As the way the large spoon was moved during these activities was also very similar they were frequently confused. The noticeable confusion between cut_into_pieces and $place\_into\_bowl$ may be due to the knife often being used to scrape chopped ingredients off the chopping board into the bowl. Stronger motion features or a representation of spatial relations between objects might help distinguish these activities.

DISCUSSION & CONCLUSION

In this paper we introduced a challenging dataset of food preparation activities with a novel combination of video and accelerometers attached to kitchen objects. The dataset contains complex interactions of multiple objects and may be used to investigate a wide range of recognition problems.

We proposed a new method for combining video and accelerometer data for activity recognition through accelerometer localization. This approach and other methods for fusing these modalities were comparatively evaluated on the new dataset. Features encoding object use showed considerable discriminative power. Similar information might have been obtained using RFID tags as an alternative to accelerometers. However, with a recognition accuracy below 50% it is clearly insufficient to solely rely on this type of feature. Motion features extracted from accelerometer data provided the strongest cues among individual feature types investigated. However, by fusing data from different sensor types via accelerometer localization and by combining features prior to classification we were able to significantly improve recognition performance. These results highlight the potential for multi-modal recognition approaches. Note that the choice of activities we used for evaluation here was deliberately made to exclude important factors such as the manipulated ingredients. If we had set out the task to differentiate between different ingredients being cut into pieces or placed into the bowl, additional (visual) features would be necessary to robustly recognize such activities. We expect that the integration of visual information will be even more beneficial for reasoning about interactions between multiple entities such as *moving the chopped tomato from the chopping board into the bowl*.

ACKNOWLEDGEMENTS

This research is funded by RCUK Digital Economy Research Hub EP/G066019/1 SIDE: Social Inclusion through the Digital Economy, from December 2010 until May 2014. We would like to thank Patrick Olivier and Axivity for providing accelerometers and for useful discussions.

REFERENCES

1. J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1 – 16:43, 2011.

2. L. Chen, C. D. Nugent, J. Biswas, and J. Hoey, editors. *Activity Recognition in Pervasive Intelligent Environments*. Springer/Atlantis Press, 2011.

3. A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3):81–227, 2012.

4. F. de la Torre, J. Hodgins, J. Montano, S. Valcarcel, R. Forcada, and J. Macey. Guide to the Carnegie Mellon University multimodal activity (CMU-MMAC) database. 2009.

5. D. Figo, P. C. Diniz, and D. R. Ferreira. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645 – 662, 2010.

6. G. Forman and M. Scholz. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *ACM SIGKDD Explorations*, 12(1):49–57, 2010.

7. J. Hoey, T. Ploetz, D. Jackson, A. Monk, C. Pham, and P. Oliver. Rapid specification and automated generation of prompting systems to assist people with dementia. *Pervasive and Mobile Computing*, 7(3):299–318, 2010.

8. J. Hoey, P. Poupart, A. v. Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

9. C.-H. Hsu and C.-H. Yu. An accelerometer based approach for indoor localization. In *Proc. UIC-ATC*, pages 223–227, 2009.

10. T. Huynh, M. Fritz, and B. Schiele. Discovery of activity patterns using topic models. In *Proc. UbiComp*, 2008.

11. J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:226–239, 1998.

12. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. IJCAI*, 1995.

13. I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2/3):107–123, 2005.

14. J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos “in the wild”. In *Proc. CVPR*, 2009.

15. M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *Proc. CVPR*, 2009.

16. B. McFee and G. Lanckriet. Learning multi-modal similarity. *Journal of Machine Learning Research*, 12:491–523, 2011.

17. R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Proc. ICCV*, 2009.

18. C. Pham and P. Oliver. Slice&Dice: recognizing food preparation activities using embedded accelerometers. *Ambient Intelligence, LNCS*, 5859:34–43, 2009.

19. T. Plötz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Proc. IJCAI*, pages 1729 – 1734, 2012.

20. M. E. Pollack. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI Magazine*, 26(2), 2005.

21. D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, M. Creatura, and J. del R. Milln. Collecting complex activity data sets in highly rich networked sensor environments. In *Proc. INSS*, 2010.

22. M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Proc. CVPR*, 2012.

23. C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proc. ICPR*, 2004.

24. S. Stein and S. J. McKenna. Accelerometer localization in the view of a stationary camera. In *Proc. CRV*, pages 109 – 116, 2012.

25. M. Tenorth, J. Bandouch, and M. Beetz. The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In *Proc. ICCV*, 2009.

26. H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc CVPR*, 2011.

27. J. P. Wherton and A. F. Monk. Problems people with dementia have with kitchen tasks: The challenge for pervasive computing. *Interacting with Computers*, 22(4):253–266, 2010.

28. J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *Proc. ICCV*, pages 1–8, 2007.

29. P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *Proc. EWSN*, 2008.