Computer Vision and Image Understanding 154 (2017) 82–93

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Recognising complex activities with histograms of relative tracklets

Sebastian Stein, Stephen J. McKenna*

Computer Vision and Image Processing, Computing, School of Science and Engineering, University of Dundee, Dundee DD1 4HN, United Kingdom

ARTICLE INFO

Article history:
Received 10 March 2016
Revised 14 July 2016
Accepted 31 August 2016
Available online 1 September 2016

Keywords:
Activity recognition
Relative tracklets
Sensor fusion
Food preparation

ABSTRACT

One approach to the recognition of complex human activities is to use feature descriptors that encode visual interactions by describing properties of local visual features with respect to trajectories of tracked objects. We explore an example of such an approach in which dense tracklets are described relative to multiple reference trajectories, providing a rich representation of complex interactions between objects of which only a subset can be tracked. Specifically, we report experiments in which reference trajectories are provided by tracking inertial sensors in a food preparation scenario. Additionally, we provide baseline results for HOG, HOF and MBH, and combine these features with others for multi-modal recognition. The proposed histograms of relative tracklets (RETLETS) showed better activity recognition performance than dense tracklets, HOG, HOF, MBH, or their combination. Our comparative evaluation of features from accelerometers and video highlighted a performance gap between visual and accelerometer-based motion features and showed a substantial performance gain when combining features from these sensor modalities. A considerable further performance gain was observed in combination with RETLETS and reference tracklet features.

1. Introduction

Activity recognition research in computer vision has made a remarkable trajectory from distinguishing full-body motion patterns like *running, boxing* and *waving* (Schuldt et al., 2004) through detecting actions of interest in movies (Laptev et al., 2008; Laptev and Prez, 2007; Liu et al., 2009) to reasoning about complex human-human (Ryoo and Aggarwal, 2009) and human-object interactions (Behera et al., 2012; Gupta et al., 2009; Ryoo and Aggarwal, 2007), and tracking through multi-step processes (Hoey et al., 2010b). These challenging problems have gained comparable interest in the ubiquitous computing community (Hoey et al., 2010a; Pham and Oliver, 2009; Plötz et al., 2012; Roggen et al., 2010) but the literature shows few examples of creative cross-fertilization and of methods for integrated activity recognition from video and embedded sensors (Behera et al., 2012; de la Torre et al., 2009; Wu et al., 2007).

We propose to recognise complex human-object interactions with feature descriptors that encode interactions by describing properties of local visual features with respect to trajectories of tracked objects. Such an approach is particularly applicable when only a subset of relevant objects can be tracked reliably. We discuss an example of this approach in detail in which dense tracklets are described relative to reference tracklets in histograms of RElative TrackLETS (RETLETS). Each histogram captures visual motion relative to a reference object. We acquire trajectories of objects to serve as reference tracklets for RETLETS using embedded sensors.

The effectiveness of this method for activity recognition is evaluated on the *50 Salads* (Stein and McKenna, 2013) dataset which is at the time of writing the only publicly available dataset that includes synchronized data from RGB-D video and accelerometers attached to objects. It captures people preparing mixed salads where activities correspond to individual tasks of a recipe and accelerometers are attached to kitchen objects. In a wide range of application areas it would be feasible to create a sensor-rich environment if the benefit of accurate activity recognition outweighed the cost. This includes, for example, augmented reality (Henderson and Feiner, 2011), cognitive situational support (Hoey et al., 2010a; 2010b), supervision of assembly tasks (Behera et al., 2012), skill assessment (Rhienmora et al., 2009), and surgery. In these contexts, activities involve a potentially large number of objects, complex interactions between hands, tools and manipulated objects, and constrained but non-unique orderings in which interactions may be performed. The challenges of recognizing such complex activities, sometimes referred to as manipulation actions (Aksoy et al., 2011; Yang et al., 2013), are well illustrated by food preparation tasks. Kitchen utensils are hard to recognize and track visually as ob-

* Corresponding author.
E-mail addresses: sstein@dundee.ac.uk (S. Stein), stephen@computing.dundee.ac.uk (S.J. McKenna).

http://dx.doi.org/10.1016/j.cviu.2016.08.012

jects are often partially occluded and object categories are defined in terms of affordances. Food preparation activities usually involve transforming one or more ingredients into a target state without specifying a particular technique or utensil that has to be used. As a potentially wide range of techniques and utensils may be employed for each activity, achieving good generalization is particularly challenging.

Whereas recognition and tracking of objects from video is challenging, embedded sensors such as accelerometers attached to objects provide information about object identity and object motion by design; they capture subtleties in object motion and continuous miniaturisation allows them to be inconspicuously integrated into a wide variety of objects. However, reasoning about interactions between objects solely based on accelerometers would require that each participating object has a sensor attached to it. Clearly it is not always practical to equip objects with sensors or tags. On the other hand, visual data effectively capture spatial relations and interactions between visual entities, assuming that they can be identified and localized. The complementarity of these sensing modalities suggests that methods for effectively combining visual data with data from embedded accelerometers have the potential to significantly improve recognition of complex activities and, importantly, to increase the range of activities that recognition systems can address. Traditionally, features from different sensor modalities are either combined for classification by concatenating feature vectors (*early fusion*), by combining semantic concept classifiers (*mid-level fusion*), or by merging classification results obtained separately from each modality (*late fusion*). Extracting features from each sensor modality independently may, however, discard important *cross-modal* relational properties. In order to reason about complex interactions from video, it is useful to relate motion captured by object-embedded sensors to locations in the image space. We present an accelerometer localization and tracking algorithm and use it to track objects in the visual field of a camera without relying on their visual appearance.

We compare quantitatively the performance of computer vision motion features and accelerometer features for activity recognition; this experiment can inform future decisions on sensor selection, how these sensors are used, and where they are placed. Since accelerometer tracking and dense tracklets are both based on dense optical flow, the proposed multi-modal features can be extracted with little additional computational cost. We focus mainly on motion features as opposed to appearance features because manipulation of objects (such as food ingredients) can severely change their appearance; appearance-based activity models are likely to capture the comparably stable appearance properties of tools and utensils. Unless training data with a wide variety of such objects were available, which is hard to achieve for practical reasons, appearance-based activity models would be likely to learn the appearance of particular object instances, and their generalization performance could not be assessed reliably. In any case, we note that the performance improvement obtained by including the well-established appearance descriptor, histograms of oriented gradients (HOG), by concatenation with motion features from both video and accelerometers, was negligible in our experiments.

This paper builds on our previously published conference papers (Stein and McKenna, 2012; 2013) in several ways. A feature descriptor is proposed that encodes relations between tracked objects and local visual features. The accelerometer localization algorithm presented in Stein and McKenna (2012) is extended to enable long-term tracking and new experiments comparing multiple tracking methods are presented. New results are reported comparing features from accelerometers and video, and evaluating modality fusion at different stages of the recognition pipeline. The contributions of this paper include the following.

- A family of feature descriptors encoding relational properties between tracked objects and local visual features.
- A method for online activity recognition based on multi-modal features from video and embedded sensor data.
- An algorithm for accelerometer tracking and a comparative evaluation of features from accelerometers and video for activity recognition.

2. Related work

This section briefly reviews related work on visual and accelerometer-derived features for activity recognition, and methods for fusing vision with inertial sensors.

2.1. Visual features for activity recognition

Features for visual activity recognition can be broadly categorized as *object-based* (Albanese et al., 2010; Behera et al., 2012; Fathi et al., 2011a; 2011b; Hoey et al., 2010a; Lei et al., 2012) or *generic* (Laptev, 2005; Matikainen et al., 2009; Messing et al., 2009; Wang et al., 2011) descriptors.

Object-based methods identify and track objects in the scene and recognize activities by reasoning about spatiotemporal relationships between them (*high-level features*). This approach usually assumes that *all* objects of interest can be detected and tracked reliably. The necessity of training reliable object detectors for all relevant objects is a major practical limitation; issues include dealing with detector uncertainty, modelling hard-to-detect deformable objects, and scaling to large numbers of different objects. Fathi et al. (2011a, 2011b) proposed to train object detectors from weak (image-level) annotations in a multiple instance learning framework and used a probabilistic graphical model for activity recognition in which nodes represented super-pixel regions, object labels, activities and a complex activity. Lei et al. (2012) recognized activities in RGB-D video based on hand-object interaction events and hand trajectory features, tracking hands using skin color and modelling objects via local color, texture, and depth descriptors of foreground regions. In these methods (Fathi et al., 2011b; Lei et al., 2012; Rohrbach et al., 2015), object detectors were trained on the specific object instances to be used at test time. Therefore, it is questionable how well these methods generalize. Rohrbach et al. (2015) proposed modelling fine-grained hand-object interactions using trajectories of tracked hands and encoding gradient and color descriptors extracted from within hands' local image neighborhoods.

Generic descriptors represent video as sets of local *low-level features* or higher-order statistics over those (*mid-level features*) (Matikainen et al., 2009), without making strong assumptions about the presence of specific objects. These methods have in common that local features are described relative to the image's frame of reference. In comparison to features extracted at spatio-temporal interest points, dense tracklets (dense fixed length point trajectories) have shown superior performance on several standard action recognition datasets (Wang et al., 2011; 2009), highlighting their discriminative power. Additional local appearance and motion features, i.e. HOG, histograms of optical flow (HOF) and motion boundary histograms (MBH), extracted along dense tracklets also outperformed the same descriptors extracted densely on a spatio-temporal grid (Wang et al., 2011), suggesting higher repeatability. Matikainen et al. (2009) proposed to model pairwise spatio-temporal relations among tracklets using a relative location probability table. As pairwise relations grow exponentially with codebook size, heuristics to populate multiple cells based on a single data point need to be applied, which severely weakens exhaustive relational models among generic features. While generic features

do not rely on object detection, it is unclear to what extent they are able to differentiate activities exhibiting similar motion but involving different objects.

This paper explores some of the middle-ground between object-based and generic descriptors by proposing a family of feature descriptors that relates generic visual features to properties of *some* detectable objects. Bilinski et al. (2013) present a degenerate case of this family, tracking a single object, a person's head, in order to extract tracklets that are invariant to the person's translational motion in the image plane. Our generalization of this approach relates local motion to multiple reference trajectories providing a rich representation of complex interactions between objects of which only a subset can be tracked. Additionally, we provide baseline results for HOG, HOF and MBH, and combine these features with others for multi-modal recognition.

2.2. Accelerometer-based activity recognition

Whereas carefully engineering or learning discriminative features are research foci in the computer vision community, activity recognition from accelerometers commonly involves standard statistical features in the temporal or frequency domain as surveyed by Figo et al. (2010). Recently, Plötz et al. (2012) applied deep belief networks to learning features from accelerometer data. Hammerla et al. (2013) reported state-of-the-art performance on a wide variety of datasets by sampling the quantile functions of acceleration magnitudes along orthogonal axes. Pham and Oliver (2009) reported promising results for recognition of food preparation actions such as scooping, stirring, peeling and chopping using statistical features in the temporal domain as well as estimates of accelerometer pitch and roll. The experiments we report incorporate the features of Pham and Oliver (2009) and Hammerla et al. (2013).

2.3. Fusing vision with inertial sensors

Fusing vision with other sensor modalities has previously been investigated for tasks including activity recognition (Behera et al., 2012; Wu et al., 2007), people tracking (Hsu and Yu, 2009) and object tracking (Stein and McKenna, 2013). Chen et al. (2015) give an overview of research combining depth and inertial sensors for action recognition. Behera et al. (2012) recognized assembly tasks by concatenating histograms of visual and inertial sensor features in an early fusion approach. Specifically, pairwise distances and changes of distance between objects recognized from a body-worn camera were encoded in a histogram as were pairwise body-part relations estimated from inertial data. The problem of localizing inertial sensors in a camera view has been primarily investigated in the context of tracking people (Maki et al., 2010; Shigeta et al., 2008; Teixeira et al., 2010; Wilson and Benko, 2014). Wilson and Benko (2014) proposed tracking peoples' phones in video using dense scene flow and Kalman filters. Teixeira et al. (2010) identified multiple people in CCTV footage based on data from magnetometers and accelerometers in mobile phones. Their method strongly relied on the person's appearance for resolving ambiguities, e.g., when people cross each other or enter and exit the scene. Shigeta et al. (2008) made similar appearance assumptions by tracking hands and jackets, and matching their trajectories to accelerometer data using normalized cross correlation (NCC). Maki et al. (2010) proposed replacing trajectories of tracked objects by trajectories of salient points tracked via KLT (Tomasi and Kanade, 1991), also using NCC for matching. In a previous paper (Stein and McKenna, 2013) we investigated accelerometer localization based on dense point trajectories and proposed a more robust similarity measure.

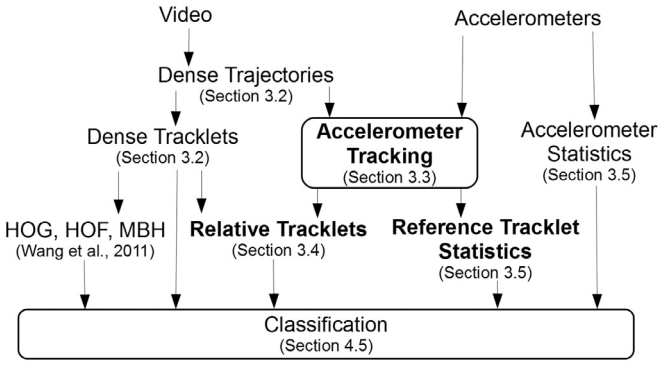

Fig. 1. Overview of data flow in the proposed method. Stages involved in encoding cross-modal properties are highlighted in bold. See Section 3.1 for details.

2.4. Activity datasets

Several public datasets for benchmarking activity recognition algorithms exist in the fields of ubiquitous computing (Huynh et al., 2008; Pham and Oliver, 2009; Roggen et al., 2010; Zappi et al., 2008) and computer vision (Liu et al., 2009; Marszałek et al., 2009; Messing et al., 2009; Rohrbach et al., 2015; Schuldt et al., 2004; Tenorth et al., 2009; de la Torre et al., 2009). We identified two key reasons for this multiplicity of datasets. Firstly, 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 its activity class (e.g., KTH (Schuldt et al., 2004), YouTube (Liu et al., 2009), Hollywood2 (Marszałek et al., 2009) and URADL (Messing et al., 2009)). 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 (Huynh et al., 2008), AmbientKitchen (Pham and Oliver, 2009), TUM Kitchen (Tenorth et al., 2009), CMU-MACC (de la Torre et al., 2009), Opportunity (Roggen et al., 2010) and MPII 2 (Rohrbach et al., 2015)). Datasets supporting activity spotting have the benefit that they can also be used purely for classification. Secondly, methods for activity recognition make varied assumptions about availability and positioning of different sensors. This poses a major challenge, particularly for research into multi-modal activity recognition, to evaluate new methods across a wide range of application scenarios and datasets. As it is extremely time-consuming to record, annotate, document, and curate a large, challenging dataset, creating datasets across a wide range of applications is a long-term community effort. By publishing the *50 Salads* dataset, we make a contribution towards this joint effort.

3. Methodology

3.1. Overview

This section introduces a family of feature descriptors called object-generic relational histograms and describes a method for multi-modal recognition of activities from accelerometers and video data. Central to the proposed recognition method is one instance from the family of relational histograms - histograms of relative tracklets (RETLETS) – that encodes interactions between tracked objects and generic motion descriptors (dense tracklets) extracted from video (Fig. 1). Specifically, the trajectories of certain objects are estimated by localizing and tracking accelerometers in video (Section 3.3). These trajectories are subsequently used as reference frames for dense tracklets (Section 3.2), which are encoded as histograms of relative tracklets with respect to each reference frame (Section 3.4). This feature descriptor capturing cross-modal

relational properties is then combined with (i) features extracted from each sensor modality independently (Sections 3.2 and 3.5), and with (ii) statistical features from the visual trajectories of localized accelerometers (Section 3.5).

3.2. Dense tracklets

The methods presented in this paper are based on dense point tracking, specifically dense tracklets as proposed for visual activity recognition by Wang et al. (2011). Point trajectories are initialized at locations $g \in G$ on a regular grid (with horizontal and vertical displacement d between grid locations) in each frame if and only if two conditions are satisfied:

1. none of the locations of active trajectories are within a $d \times d$ neighborhood around the grid point g, and
2. the minimum eigenvalue, $min(\lambda_1^{(g)}, \lambda_2^{(g)})$, of the auto-correlation matrix of the image at location g is larger than the threshold $\tau_\lambda = 0.001 \cdot \ max_{h \in G} \min(\lambda_1^{(h)}, \lambda_2^{(h)})$.

The displacement of a point from one frame to the next is estimated as the median-filtered dense optical flow field in a 3×3 neighborhood around the point's location in the previous frame.

Tracklets encode point trajectories $P : (\mathbf{x}_0, \ldots, \mathbf{x}_{L-1})$ of image coordinates $\mathbf{x}$: (x, y) with fixed length L as $L-1$ displacements $\Delta\mathbf{x}_j : (x_{j+1} - x_j, y_{j+1} - y_j)$ that are normalized by the total length of displacements (1).

$$T = \frac{(\Delta\mathbf{x}_0, \cdots, \Delta\mathbf{x}_{L-2})}{\sum_{j=0}^{L-2} ||\Delta\mathbf{x}_j||_2} \tag{1}$$

Normalizing a trajectory by its total length emphasizes the trajectory's shape. Tracklets are extracted at multiple spatial scales. Optionally, HOG, HOF and MBH descriptors (Wang et al., 2011) are extracted from the local $32 \times 32 \times L$ neighborhood around each tracklet. Features extracted from a spatio-temporal video window are encoded as a histogram over codebook features (bag-of-words) for classification. Codebooks are obtained via k-means clustering of features from a training set.

3.3. Accelerometer localization and tracking

Localizing accelerometers in the visual field of a camera is non-trivial for a number of reasons. Firstly, accelerometers are usually visually occluded. An accelerometer may be occluded by the object it is attached to or embedded into, in which case the motion observed at the visible location of the object is likely to be similar to the motion captured by the accelerometer. It may, however, also be occluded by a different visual entity in which case the visual motion at the accelerometer's location and the accelerometer's motion projected in the image plane are likely to differ. Secondly, accelerometers capture tri-axial translational acceleration with respect to a local reference frame; in general an accelerometer's orientation is unknown and changes over time, making alignment with the camera's frame of reference problematic. Thirdly, accelerometers measure proper acceleration (relative to free fall) whereas acceleration estimated from visual motion represents coordinate acceleration (relative to the camera's frame of reference). Further issues include sensor synchronization and dealing with differences in sensor frequencies.

The proposed method for accelerometer localisation involves generating location proposals in videos, estimating local visual accelerations at these locations and matching acceleration estimates to accelerometer data. Location proposals are generated by sampling points in the video. Tracked point sequences, i.e., point trajectories, are used to estimate acceleration. Section 3.3.1 describes two methods for generating these point trajectories. Locations along point trajectories are transformed into world coordinates and a scoring function is applied to determine point trajectories that best match the accelerometer data. The location of the best matching point trajectory in the most recent frame is regarded as the estimated accelerometer location. An example similarity map representing scores for all location proposals is shown in Fig. 2.

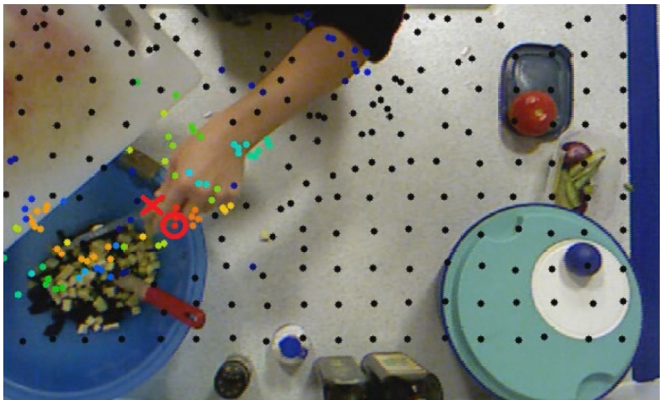

Fig. 2. Accelerometer localization (best viewed in color). By measuring similarity of accelerations along trajectories of point features (colored dots) with accelerometer data (black indicates weakest, red indicates strongest similarity), the algorithm estimates the accelerometer location (red circle). A red cross marks the ground-truth. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.3.1. Generating location proposals

For sampling location proposals and point tracking the initial steps involved in extracting dense tracklets as introduced in Section 3.2 are followed. Points are sampled on a regular grid and updated based on frame-by-frame dense optical flow (Farnebäck, 2003). A new sample is initialized at a grid location if no existing samples lie in a $d \times d$ neighborhood centred at that location. Whereas the dense tracklets described in Section 3.2 last for some fixed, pre-specified number of frames, the point trajectories generated here using optical flow are not of fixed length. Instead they are terminated if they pass too close to an older trajectory. In effect this imposes an upper bound on the number of trajectories. Specifically, a trajectory is terminated if its location in the most recent frame becomes closer than some threshold τ_d to another track's location and it is younger than that other track.

We compare this method to sparsely tracking keypoints in the image at which the Hessian has two large eigenvalues (Bouguet, 1999) (the smaller eigenvalue being the *cornerness*). Using sparse tracking we maintain a fixed number of tracks, N_t, at all times. In the first frame, N_t points are initialized at locations with highest cornerness under the constraint that no two points lie within d pixels of each other. In every subsequent frame, points that cannot be tracked reliably get replaced by new keypoints.

The set of location proposals at frame t consists of locations $\mathbf{x}_t^{(i)}$ along all point trajectories $P^{(i)}$ that are tracked until that frame.

3.3.2. Transformation to world coordinates

Two transformations are needed in order to match accelerations estimated along point trajectories with data captured by accelerometers: (i) point trajectories in image coordinates need to be transformed into world coordinates (i.e., metric values with the center of the image plane at $\mathbf{0}$), and (ii) gravitational effects need to be simulated to transform coordinate acceleration to proper acceleration (relative to free fall).

The relationship between image and world coordinates is governed by the distance of objects from the camera, and the camera's intrinsic parameters. Assuming a pinhole camera model, intrinsic parameters and distortion coefficients were determined as proposed by Zhang (2000) and Brown (1966), respectively, from multiple views of a chessboard pattern. Given an undistorted image location (x, y), estimated depth z, focal length f, imaging element dimensions s_x and s_y, and principal point (c_x, c_y), image locations are transformed to world coordinates using (2).

$$x' = \frac{(x - c_x)z}{fs_x} \qquad y' = \frac{(y - c_y)z}{fs_y} \qquad z' = z \tag{2}$$

We investigate two methods for assigning depth values z to pixels. First, we assign each pixel the value from the depth map provided by the structured light sensor. In this case depth values of some pixels frequently become unavailable due to shadows of the structured light pattern and transparent or specular reflective surfaces, for example. When this situation occurs we extrapolate from previous depth values and estimate velocity along the point trajectory. Second, we assume a constant depth for all pixels in the image. Surprisingly, we observed lower localization accuracy using depth maps than using a constant fixed depth.

The estimation accuracies of x', y' and z' in (2) depend linearly on the estimated depth z, and errors in z are exacerbated when estimating accelerations from sequences of locations. With z estimated using a depth sensor, acceleration estimates are particularly sensitive to location estimates crossing depth discontinuities as these induce erroneous instantaneous spikes in acceleration. Assuming a constant fixed depth avoids these strong errors, but introduces noise as an object moves away from the pre-set depth and fails to capture acceleration along the z-axis. Both of these types of errors are relatively small if the chosen fixed depth is set to a reasonable value, and if the motion along z is small compared the distance to the camera or small compared to motion in x and y.

Let $P' : (\mathbf{x}'_0, \ldots, \mathbf{x}'_t)$ denote a point trajectory represented as a sequence of locations in world coordinates, $\mathbf{x}'_j : (x'_j, y'_j, z'_j)$. Velocities $\mathbf{v}'_j$ and accelerations $\mathbf{a}'_j$ are approximated using discrete differences $\mathbf{v}'_j = f_{vid}(\mathbf{x}'_j - \mathbf{x}'_{j-1})$ and $\mathbf{a}'_j = f_{vid}(\mathbf{v}'_j - \mathbf{v}'_{j-1})$, respectively, where f_{vid} is the video frame rate. Locations $\mathbf{x}'_j$ are smoothed with a zero-mean Gaussian with some small standard deviation to avoid instabilities in the approximation (Rao et al., 2002).

Ideally, one would transform accelerometer data to coordinate acceleration by subtracting acceleration measured due to gravity, but as accelerometer orientation relative to the direction of gravity is unknown and changing over time this is not possible. Fortunately, acceleration due to gravity can be simulated and added to acceleration estimated along point trajectories if the direction of gravity in video can be estimated. We propose to determine the direction of gravity in the camera's field of view by estimating surface normals from depth maps. Assuming there is a planar surface in the scene that is aligned with gravity (e.g., a floor, a ceiling, a work surface or a tabletop) we take a pragmatic approach and estimate the normal from a set of at least three manually marked points on the surface. Given three such points in world coordinates $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, the direction of gravity is given by the cross-product of the co-planar vectors $\mathbf{u} = \mathbf{p}_1 - \mathbf{p}_0$ and $\mathbf{v} = \mathbf{p}_2 - \mathbf{p}_0$, and the magnitude follows from standard gravity (3).

$$\hat{\mathbf{g}} = 9.80665 \cdot \frac{\mathbf{u} \times \mathbf{v}}{||\mathbf{u} \times \mathbf{v}||} \tag{3}$$

The estimated proper acceleration $\mathbf{a}_t$ of a point trajectory at time t is given by $\mathbf{a}_t = \mathbf{a}'_t + \hat{\mathbf{g}}$.

3.3.3. Similarity scoring and localization

Accelerometers are localized by estimating similarity scores between acceleration sequences $A^{(i)}_{vid} : (\mathbf{a}^{(i)}_{t-j}, \ldots, \mathbf{a}^{(i)}_t)$ estimated at all location proposals $\mathbf{x}^{(i)}_t$ and the sequence of accelerometer data $A_{acc} : (\mathbf{a}^{acc}_{t-j}, \ldots, \mathbf{a}^{acc}_t)$. The video frame rate f_{vid} is usually lower than the accelerometer sampling rate f_{acc}. For a one-to-one association the accelerometer data needs to be sub-sampled. Acceleration as measured by an accelerometer corresponds to instantaneous properties of the sensor. Because acceleration experienced by the sensor *in between* subsequent samples is unknown, it is advisable to match visual acceleration estimates to the temporally closest accelerometer sample rather than using interpolation. Thereby accelerometer data is implicitly sub-sampled as some samples remain unmatched. We confirmed this preference by comparative empirical evaluation.

Since accelerometer orientation is unknown and changing over time, a similarity score between acceleration *norms* is established. A moving object's visual trajectory is most easily discriminated from those of other objects during periods when its velocity changes frequently. Unfortunately, the similarity of raw acceleration sequences during such periods is sensitive to synchronization errors and to differences between instantaneous acceleration measured by accelerometers and mean acceleration between frames as estimated from video. In order to address this issue we take the radical step of transforming sequences $A^{(i)}_{vid}$ and A_{acc} into binary sequences $B : (b_{t-j}, \ldots, b_t)$, where each element b_{t-j} is non-zero if and only if the absolute difference between the acceleration norm and the magnitude of standard gravity exceeds some noise threshold τ_a, as in (4) where $1[]$ is the indicator function. While this transformation discards most information on acceleration magnitudes it preserves local extrema and saddle points in the corresponding velocity sequences.

$$b_{t-j} = 1[|\mathbf{a}|_{t-j} - |\mathbf{g}| \geq \tau_a] \tag{4}$$

We define an efficient, recursive similarity score S between pairs of binary sequences which gives higher weight to recent frames using a multiplicative temporal decay $\alpha \in [0, 1)$ in (5) and (6).

$$S^{(i)}_{t-j-1}(B^{(i)}_{vid}, B_{acc}) = 0 \tag{5}$$

$$S^{(i)}_t(B^{(i)}_{vid}, B_{acc}) = \alpha S^{(i)}_{t-1} + b^{(i)}_t b^{acc}_t \tag{6}$$

The similarity score is thereby defined as the number of frames in which both sensors capture significant acceleration, reducing the impact of samples in the past through temporal decay. As new location proposals get initialized others have already accumulated a potentially high similarity score over time. Through empirical evaluation we found that the algorithm becomes more effective if after the first frame the similarity score of new location proposals is initialized to the similarity score of the closest location proposal. For each accelerometer, the similarity score is estimated between the corresponding sequence B_{acc} and $B^{(i)}_{vid}$ corresponding to all location proposals in frame t. Finally, accelerometer location is estimated as the location proposal corresponding to the binary sequence with highest similarity score (7).

$$\mathbf{x}^{acc}_t = \mathbf{x}^{(\hat{i})}_t, \text{ where } \hat{i} = argmax_{(i)}\ S^{(i)}_t \tag{7}$$

3.3.4. Long-term accelerometer tracking

If an occluding object's motion differs from the accelerometer motion, a previously correctly matched trajectory is likely to drift away from the correct location as it tracks the occluder. This scenario frequently occurs, for example, after an accelerometer-equipped utensil has been released and the hand that previously held the device moves away. In this case $(b^{acc}_t = 0)$, the similarity scores of all location proposals are updated to $S^{(i)}_t = \alpha S^{(i)}_{t-1}$ and,

as the ranking of hypotheses does not change, the estimated accelerometer location diverges from the true location. For this reason we extend our approach by detecting when an accelerometer is not moving, taking a snap-shot of the similarity scores, and re-initializing similarity scores once an accelerometer starts moving again.

A sample $b^{acc} = 0$ indicates constant velocity. As it is unlikely that motion induced by a human exhibits constant velocity over an extended period of time, it is likely that any sequence $(b_s^{acc}, \ldots, b_{t-j}^{acc}, \ldots, b_t^{acc})$ with $s \ll t$ and $b_{t-j}^{acc} = 0$ for all $j \in [0, t-s]$ is generated by a stationary device. At each time instant s with $b_s^{acc} = 0$ and $b_{s-1}^{acc} \neq 0$ a similarity map $\mathcal{M} : (S_s^{(i)}, \mathbf{x}_s^{(i)})$ consisting of similarity scores and associated location proposals is generated. Once the length of the interval $[s, t]$ in which an accelerometer continuously measures no significant acceleration exceeds a threshold τ_t, the location of that accelerometer is temporarily estimated as $\mathbf{x}_s^{(\hat{i})}$ until $b_{s+k}^{acc} = 1$ for some positive k. At time $s+k$ when an accelerometer measures significant acceleration after being stationary, the similarity scores of location proposals $\mathbf{x}_t^{(i)}$ are re-initialized. We found empirically that initializing $S_t^{(i)}$ to the similarity score in $\mathcal{M}$ corresponding to the nearest neighbor of $\mathbf{x}_t^{(i)}$ is most effective, compared to no re-initialization, cold start ($S_t^{(i)} = 0$ for all i) and kernel density estimation.

3.4. Object-generic relational histograms

Methods for modelling activities from local features usually follow a bag-of-words approach encoding the occurrence frequency of codewords, essentially discarding spatial relations between features. Spatio-temporal pyramids address this issue to some extent by coarsely encoding feature co-occurrence, but they are very limited in accurately capturing interactions that span across spatial segmentation boundaries (Laptev et al., 2008). Recognition of complex interactions based on tracking all objects of interest often relies on high-level reasoning methods which are computationally demanding and domain specific. This section presents object-generic relational histograms, a family of descriptors that captures relations between generic local features and *reference features* extracted from some objects. This feature representation adapts the bag-of-words model to scenarios in which some objects can be detected or tracked, and facilitates recognition of complex interactions with standard classification algorithms such as support vector machines (SVM). First, a formalization of the family of relational histograms is presented. Then, RETLETS – one instance of this family – is introduced. Subsequently, we use RETLETS to capture relational properties between dense tracklets and accelerometers' motion by encoding dense tracklets relative to reference tracklets acquired via accelerometer localization (Section 3.3).

3.4.1. Relational histograms

Consider a set of M local features $\{(\mathbf{f}_m, \mathbf{x}_m)\}_{m=1}^{M}$ consisting of feature descriptor $\mathbf{f}_m$ and location in the image $\mathbf{x}_m$, and a set of N reference features $\{(\mathbf{f}_n^{ref}, \mathbf{x}_n)\}_{n=1}^{N}$ extracted from N tracked objects. In order to encode interactions between local features and reference features we propose to construct N histograms H_n, one for each reference feature. Each histogram encodes pairwise relations $R(\mathbf{f}_n^{ref}, \mathbf{f}_m)$ between the descriptor of one reference feature and the descriptors of all local features using a codebook $\mathcal{C}$ of quantized pairwise relations. The codebook could, for example, encode feature co-occurrence, difference in appearance, relative location or relative motion. The contribution of each pairwise relation to a histogram is weighted by the likelihood of a meaningful interaction $w_{n,m}$. Given a quantization function $q(R(\mathbf{f}_n^{ref}, \mathbf{f}_m)) : \mathbb{R}^{|\mathbf{f}|} \times \mathbb{R}^{|\mathbf{f}|} \rightarrow [0,1]^{|\mathcal{C}|}$, weighted relational histograms are constructed using (8) and (9).

$$H_n = \sum_{m=1}^{M} w_{n,m} q(R(\mathbf{f}_n^{ref}, \mathbf{f}_m)) \tag{8}$$

All weighted histograms are individually L_1-normalized. Each histogram H_n provides a different representation of the set of local features by encoding their relations to one reference feature. Depending on the choice of relational codebook $\mathcal{C}$ and spatial weighting function this descriptor can encode meaningful interactions between a reference object and local visual features in its proximity. The presence of a meaningful interaction between a local feature and a reference feature is intuitively related to their spatial separation. We chose to weight the contribution of a feature $\mathbf{f}_m$ to a histogram H_n using a Gaussian function with Euclidean distance (9) for point features and with mean Euclidean distance along point trajectories (12), respectively.

$$w_{n,m} = \exp\left(-\frac{||\mathbf{x}_n^{ref} - \mathbf{x}_m||^2}{2\sigma^2}\right) \tag{9}$$

This formulation provides a generic model of relational histograms that can be used with a wide variety of local feature descriptors and pair-wise relations. Below, one instance of this family is described which we use for modelling interactions between dense tracklets and accelerometer-equipped objects.

3.4.2. Histograms of relative tracklets (RETLETS)

While feature co-occurrence may be a suitable second-order statistic for local appearance features, a *relative* description of local visual motion features better captures, in qualitative terms, interactions such as visual entities moving *towards*, *away from* and *around* each other (see Fig. 3). A descriptor encoding generic video tracklets relative to semantically meaningful reference tracklets acquired by tracking some objects is therefore more informative for complex interactions of multiple objects (see Fig. 4). This section proposes relational histograms using densely sampled fixed length point trajectories P_m as local features $\mathbf{f}_m$, using fixed length reference trajectories P_n^{ref} acquired through some form of object tracking as reference features $\mathbf{f}_n^{ref}$ and using relative tracklets R_m as pairwise relations R.

Given a pair (P_m, P_n^{ref}), the relative trajectory P_m^{rel} is defined as the sequence of differences between point locations (10). The difference between a pair of point locations $(\mathbf{x}_0^{(m)}, \mathbf{x}_0^{ref})$ describes the location $\mathbf{x}_0^{(m)}$ relative to the location $\mathbf{x}_0^{ref}$, and the sequence of relative locations describes the motion of the visual entity tracked by P_m from the perspective of the reference feature P^{ref}. This relative motion is illustrated in Fig. 4(b).

$$P_m^{rel} = \left(\left(\mathbf{x}_0^{(m)} - \mathbf{x}_0^{ref}\right), \ldots, \left(\mathbf{x}_{L-1}^{(m)} - \mathbf{x}_{L-1}^{ref}\right)\right) \tag{10}$$

Similar to Eq. (1), the relative tracklet R_m is defined as the sequence of normalized displacements along the relative trajectory P^{rel} as in Eq. (11).

$$R_m = \frac{(\Delta\mathbf{x}_{m,0}^{rel}, \ldots, \Delta\mathbf{x}_{m,L-2}^{rel})}{\sum_{j=0}^{L-2} ||\Delta\mathbf{x}_{m,j}^{rel}||_2} \tag{11}$$

As tracklets are extracted along a sequence of points in the image, weights in Eq. (8) are determined based on the mean pair-wise distance between locations along the corresponding point trajectories (12).

$$w_{n,m} = \exp\left[-\frac{1}{2\sigma^2}\left(\frac{1}{L}\sum_{l=0}^{L-1} ||\mathbf{x}_l^{(m)} - \mathbf{x}_l^{ref}||\right)^2\right] \tag{12}$$

The relational codebook $\mathcal{C}$ is trained using k-means clustering on a training set of relative tracklets R_m. The Voronoi cells defined

Fig. 3. Point trajectories (green) in the left and right image have similar shape (best viewed in color). However, relative to the trajectory of the large spoon (red), points in the left image move *towards* whereas most points in the right image move *away from* this reference trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

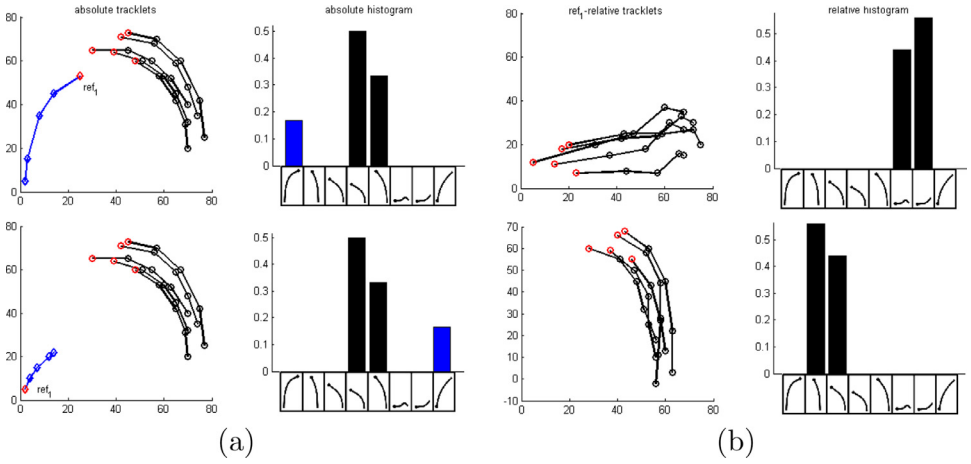

Fig. 4. Absolute tracklets and RETLETS. A toy example in which (a) histograms of absolute tracklets differ in only two bins whereas (b) tracklets relative to a reference tracklet (ref_1-relative) change their shape entirely and make the corresponding histogram representation more discriminative. Distance-based re-weighting of histogram entries adds further to discriminative power.

by the k cluster centers define the quantization function q. In every frame of a video, dense tracklets are extracted and reference tracklets are newly determined. For each reference tracklet, relative tracklets are constructed using all dense tracklets ending in the same frame using (10). All relative tracklets are then mapped onto the codebook $\mathcal{C}$ using quantization function q, and inserted to their respective histogram with their contribution weighted using (9) and (12).

Encoding tracklets relative to reference signals can be disadvantageous. The signal to noise ratio of an individual relative tracklet is usually lower compared to absolute tracklets as noise in trajectory estimates of a point feature and of a reference object are added. The potential negative impact this may have on classification is reduced in the RETLET descriptor by encoding each tracklet relative to multiple reference tracklets, which provides some noise averaging assuming estimation noise along reference tracklets are mutually independent. If reference object tracking fails, e.g. due to tracker uncertainty or objects leaving the cameras field of view, tracklets are encoded relative to a false reference. Note however, that each tracklet is also encoded relative to all other reference objects, and if at least one reference object is not in use, the relative description resembles the absolute tracklet with some added noise. Thereby, the RETLET encoding provides some robustness against tracking failure.

In this paper, we investigate visual accelerometer tracking as a method for generating reference tracklets from accelerometer-equipped objects. Analysis of relations between accelerometer motion and visual features in spatial proximity to the device allows for joint reasoning about *how* an accelerometer-equipped object moves and *what* it interacts with.

3.5. Online activity recognition

We recognize activities by classifying features extracted from temporal sliding windows. We refer to this as online recognition, because activities commonly extend beyond the end of a sliding window and a decision about the ongoing activity has to be made without inspecting data from the future. For each accelerometer-equipped object we create one reference tracklet from the most recent $L-1$ displacements of the point trajectory with highest similarity score. These reference tracklets are used to construct *RETLETS*. We also use dense tracklets, which are subsequently called *Absolute Tracklets*, as well as HOG, HOF and MBH descriptors along tracklets (Wang et al., 2011) for classification. Additionally, we extract features from raw accelerometer data (*Accelerometer Statistics*, *Object Use* and ECDF (Hammerla et al., 2013)) and from their respective visual trajectories (*Reference Tracklet Statistics*).

Accelerometer Statistics: Accelerometer data were encoded as features previously shown to give good performance on a recognition task involving food preparation activities (Pham and Oliver, 2009). Mean, standard deviation, energy and entropy were

extracted from each acceleration axis over the entire temporal window. Additionally, pitch and roll were estimated from four temporal sub-windows with 50% overlap. This set of features encodes each accelerometer's motion with a 20-dimensional feature vector.

Object use: As a baseline, we consider simple features that encode whether an accelerometer-equipped object is in use. Following argumentation in Section 3.3.4, an accelerometer is stationary if it measures no significant acceleration over an extended period of time. Assuming an accelerometer is in use if and only if it is moving, Object Use at time t is defined as in (13).

$$InUse(B^{acc}) = \sum_{j=0}^{\tau_t - 1} b_{t-j} > 0 \tag{13}$$

Reference Tracklet Statistics: From the most recent $L-1$ displacement vectors of the point trajectory that best matches an accelerometer's data, the mean, standard deviation, energy and entropy were estimated separately for displacements in x- and y-coordinates in the image.

Features extracted from temporal sliding windows were classified using one-vs-all, multi-class support vector machines (SVMs). For *Absolute Tracklets*, *RETLETS*, HOG, HOF and MBH we used the RBF-χ^2 kernel with $\gamma = \frac{1}{A}$, where A is the average distance between training histograms (Zhang et al., 2007). For *Accelerometer Statistics*, *Object Use* and *Reference Tracklet Statistics*, features extracted for each accelerometer were concatenated and feature vectors were compared using the squared Euclidean distance (Gaussian-RBF) after scaling all dimensions individually to $[-1, 1]$. γ was determined by cross-validation.

4. Evaluation

After introducing the 50 Salads dataset, this Section then presents empirical evaluations. Firstly, quantitative evaluation of the accelerometer location methods is reported (Section 4.2). Section 4.3 details the protocol used for activity recognition experiments and recognition results are then presented in Sections 4.4–4.6. These compare accelerometer features, visual features, the proposed RETLETS, and various combinations of features.

4.1. Scenario and data acquisition

The methods presented in this paper make several assumptions about the sensor setup: (i) the camera is equipped with a depth sensor which captures a surface that is perpendicular to the direction of gravity, (ii) some objects (or body parts) involved in interactions are equipped with accelerometers, and (iii) those objects are in the camera view when in use. To the best of our knowledge, none of the existing public datasets meets all of these criteria.

We have created and released annotated data of food preparation activities for evaluation purposes[1]. These are, to the best of our knowledge, the only readily available datasets combining RGBD-video and accelerometers attached to objects (as opposed to people). Fig. 5 shows an illustrative snapshot. More than 4 h of data were acquired and annotated, consisting of RGB-D video (30*Hz*) with a top-down view onto a work surface and readings from tri-axial accelerometers (50*Hz*) attached to kitchen objects[2]. The main data set which we call 50 Salads includes 50 sequences of people preparing a mixed salad with two sequences per subject. Preparing the salad involved mixing a dressing from salt, pepper, oil and balsamic vinegar, cutting cucumber, tomato, lettuce and cheese into pieces, mixing the ingredients, serving the salad onto a plate and dressing the salad. Accelerometers were attached to a knife, a peeler, a large spoon, a small spoon, a dressing glass, a pepper dispenser and an oil bottle. In addition to recruiting participants from different gender and a wide range of age, ethnicity and cooking experience, further variation was introduced by giving subjects a different task-ordering for each sequence, sampled from an activity model (Stein and McKenna, 2013).

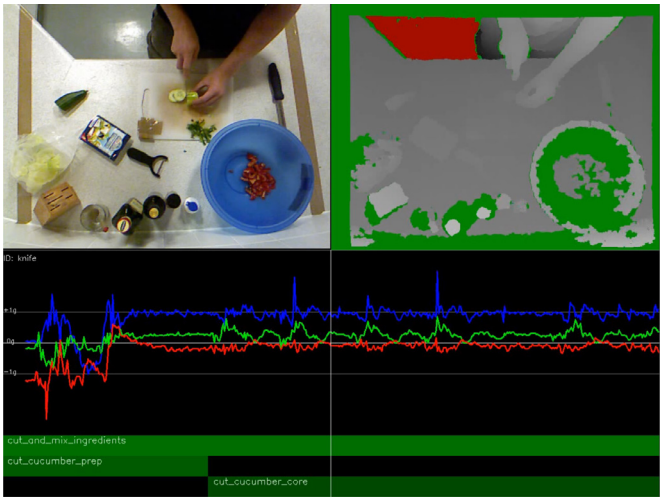

Fig. 5. A snapshot from the dataset showing RGB-D video (top), accelerometer data (middle) and activity annotations (bottom).

4.2. Accelerometer tracking

In contrast to published work by other authors on accelerometer localisation (Maki et al., 2010; Shigeta et al., 2008) which evaluated localisation only qualitatively, we report quantitative evaluation. The locations of three accelerometers attached to a knife, a spoon and the rim of a bowl were annotated in every frame of a 13,263-frame sequence (31,346 accelerometer samples) by manually clicking on the image at the estimated location of the geometric center of the devices. We identified 16 sub-sequences during which at least one accelerometer measured strong acceleration. These are used for evaluation as they account for all the intervals in the sequence during which at least one object with an embedded accelerometer was in use. We compared point trajectories generated from dense optical flow (DOF) with sparse point tracking (KLT), estimating the distance from the camera from depth maps (variable) or with a manually defined constant depth (fixed). For a fair comparison we optimized the parameters of each point tracking method empirically. Dense trajectories were initialized on a grid with $d = 24$ pixels and terminated based on a threshold $\tau_d = 5$ pixels. For sparse point tracking, the maximum number of trajectories was set to $N_t = 96$ with a minimum distance at initialization of $d = 14$ pixels. The fixed depth was set to $\hat{z} = 0.9m$, which corresponds roughly to the operating height of the camera. In all experiments accelerations were estimated from Gaussian-filtered locations with $\sigma = \frac{0.3}{f_{vid}}$ and a temporal decay $\alpha = 0.9982$ was used.

As shown in Table 1, point trajectories from dense optical flow with fixed depth outperformed all other configurations on average. We suspect the substantial difference in performance to KLT to be due to the smoothness of the dense flow field, which significantly reduces the number of false feature correspondences, and the better coverage of low-texture regions obtained with uniform sampling. The depth maps produced by the camera are clearly not reliable enough for extending point trajectories to 3D. This might be

[1] All data created during this research are openly available from the University of Dundee Institutional Repository at http://doi.org/10.15132/10000120 (*50 Salads*) and at http://doi.org/10.15132/10000121 (*Accelerometer Localization*).

[2] Sensors used were a Kinect RGB-D camera and Axivity tri-axial wireless accelerometers.

Table 1
Average Euclidean distance of estimated accelerometer location from ground truth (in pixels) for different point tracking methods using measured (variable) and hard-coded (fixed) depth.

		KLT		DOF	
		Depth		Depth	
Seq. ID	#Frames	fixed	var.	fixed	var.
1	203	74	56	**32**	50
2	25	206	313	**92**	99
3	310	**23**	28	25	35
4	296	**19**	28	49	32
5	125	143	215	53	**46**
6	45	89	259	**67**	94
7	106	**67**	79	89	84
8	85	138	146	**48**	69
9	45	131	213	92	**74**
10	170	66	304	**36**	91
11	123	**65**	81	93	114
12	1133	60	184	**23**	179
13	375	**62**	63	97	105
14	95	81	87	**54**	76
15	798	135	265	**52**	104
16	233	**38**	338	74	85
Total	4167	76	167	**49**	106

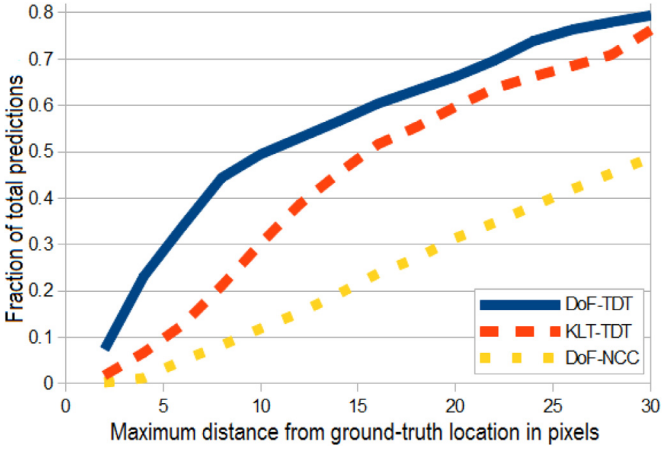

Fig. 6. Accelerometer localization accuracy.

due, in part, to noisy depth measurements and holes in the depth maps on areas that lie in the shadow of the structured light pattern.

We plotted cumulative distributions over the distances of predicted locations to ground-truth to gain understanding of the localization precision and to compare our temporal decay thresholding (TDT) method with normalized cross-correlation (NCC) as used by Maki et al. (2010) (Fig. 6). Specifically, we compared our method (DOF and KLT with fixed depth) to NCC using DOF trajectories. The plotted NCC result was obtained using a temporal window size of 150 frames; this gave the lowest average distance from ground-truth (114 pixels) of all the window sizes tried in the range 10 to 300 frames. Although the proposed method leaves room for improvement, it clearly outperforms NCC, increasing the fraction of predictions within a 10 pixel radius from 12% to 50% and withing a 30 pixel radius from 50% to 80%.

In order to compare the accuracy of long-term accelerometer tracking with different methods for re-initializing hypothesis scores after an accelerometer has been stationary, we evaluated the predicted accelerometer location in every frame of the entire video (Fig. 7). We compared no re-initialization and cold start (initialization to $S_t = 0$) with two methods for re-initialization based

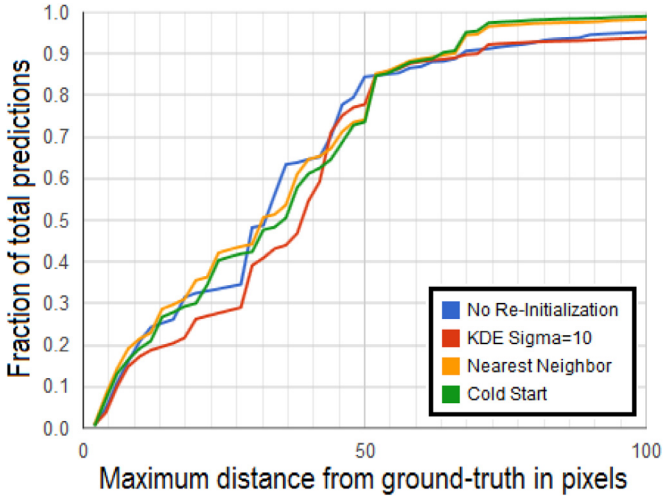

Fig. 7. Long-term accelerometer tracking accuracy (best viewed in color).

on the similarity map $\mathcal{M}$: (i) assigning the score of the nearest neighbor and (ii) kernel density estimation (KDE). KDE did not perform better than other methods with $\sigma = 1, \ldots, 10$. (For clarity we only plot KDE results for $\sigma = 10$). While none of the explored strategies clearly outperforms the others, re-initialization from the nearest neighbor in $\mathcal{M}$ and cold start show a significantly higher fraction of predictions in the range up to 25 pixels and approach saturation closer to the ground-truth location. As nearest-neighbor slightly outperforms cold start, we employ this method in all subsequent experiments. The shapes of the graphs in Fig. 7 give some indication for how frequently the tracker loses the target object. The roughly linear increase from 0 to 50 pixels and almost constant frequency between 50 and 100 pixels suggests that, on average, the target object is lost if the prediction is more than 50 pixels away from the ground-truth. Among all evaluated methods for long-term tracking, this occurs in 5–10% of frames.

4.3. Activity recognition evaluation protocol

In Sections 4.4–4.6, we report results on the task of classifying spatio-temporal windows into C activity classes, namely *add pepper*, *add oil*, *mix dressing*, *peel cucumber*, *cut ingredient*, *place ingredient into bowl*, *mix ingredients*, *serve salad onto plate*, *dress salad* and *NULL*, where *NULL* indicates that none of the other activities currently occurs. The *50 Salads* dataset was partitioned into five folds. Each test set consisted of two sequences of each of five participants; the corresponding training set consisted of two sequences of each of the remaining 20 participants. SVM parameters were determined via nested 5-fold cross-validation on each training set, using sequences from 16 participants for training and sequences from the remaining four participants for validation, followed by testing on the held-out set. Performance was measured as mean precision, mean recall and their harmonic mean (f-measure). For an unbiased estimate of recognition performance based on unbalanced test data, class precision and recall were weighted inverse proportionally to their occurrence in the test set when aggregated.

Features were extracted from temporal intervals of 154 video frames or 256 accelerometer samples (~ 3 s) at each video frame. A stratified sub-sample of all features extracted from training data was used. Codebooks with varied size k were learned from a sub-sample of 100 k tracklets extracted from training data. k-means was initialized 8 times and the codebook with minimal reconstruction error kept.

Table 2
Comparison of recognition performance observed with visual features, accelerometer features and sensor fusion methods on the 50 Salads dataset (in %).

Visual features ($k = 500$)	Precision	Recall	f-measure
Absolute Tracklets (AT)	42 ± 2	43 ± 4	43
HOG	50 ± 3	49 ± 3	49
HOF	48 ± 3	47 ± 4	47
MBH	54 ± 5	52 ± 5	53
Visual motion (VM)			
AT, HOF, MBH	55 ± 5	53 ± 6	54
Baseline vision (VIS)			
AT, HOG, HOF, MBH	**59 ± 4**	**58 ± 4**	**58**
Accelerometer Features	Precision	Recall	f-measure
Object Use	43 ± 3	50 ± 2	46
ECDF (Hammerla et al., 2013)	**60 ± 2**	**64 ± 5**	**62**
Baseline accelerometers (ACC)			
Accelerometer Statistics	**60 ± 2**	63 ± 6	**62**
Sensor fusion	Precision	Recall	f-measure
Ref. Tracklet Statistics (REF)	51 ± 3	50 ± 2	51
RETLETS ($k = 250, \sigma = 360$)	63 ± 3	62 ± 4	62
Baseline motion			
VM, ACC	70 ± 3	70 ± 3	70
Baseline fusion			
VIS, ACC	71 ± 3	71 ± 3	71
VIS, ACC, RETLETS	74 ± 3	74 ± 2	74
Proposed method			
VIS, ACC, REF, RETLETS	**76 ± 3**	**76 ± 2**	**76**

4.4. Visual features vs. accelerometers

By comparing recognition performance obtained with features extracted from embedded accelerometers and visual data independently we aim to further motivate our multi-modal approach, and to justify a sensor-rich environment for certain applications. We compared recognition performance obtained with *Absolute Tracklets*, HOG, HOF, MBH, *Object Use*, ECDF and *Accelerometer Statistics*. Features along tracklets were extracted using the same parameters as in Wang et al. (2011). The codebook size was set to $k = 500$ where performance saturated during cross-validation (see Section 4.5). The results, presented in Table 2 (top and middle), confirm that the problem under investigation is sufficiently challenging. The best performance of 60% precision at 64% recall was obtained with ECDF, with *Accelerometer Statistics* showing comparable performance. The best performance using visual features, observed when combining *Absolute Tracklets*, HOG, HOF, and MBH descriptors as proposed in Wang et al. (2011), was comparably lower at 59% precision and 58% recall. Comparing results observed with visual features, it is interesting to note that the combination of visual motion descriptors only showed marginal improvements over MBH, whereas adding local texture features (HOG) improved performance by about 5% compared to MBH. It may seem surprising that the simplest type of feature considered here, *Object Use*, compared favorably with *Absolute Tracklets*. This result indicates that, in this experimental scenario, the identity of objects involved in an activity is as discriminative as a generic description of motion in the scene. The comparable importance of the identity of objects involved in an activity and motion descriptors matches our intuition, considering that differences in visually observable motion across food preparation activities are very subtle, and knowledge about the involvement of specialized tools in an activity can significantly reduce the number of possibly occurring activities. Furthermore, the considerable margin between the results using *Accelerometer Statistics* and *Object Use* indicates that object involvement and motion characteristics are strongly complementary.

From a traditional computer vision perspective these results might suggest to use a method in which objects involved in activities of interest are detected and tracked over time, and activities are recognized by reasoning about these object's (relative) position and motion. Such an approach is problematic for reasons of scalability and reliability. Learning detectors for all objects requires substantial amounts of labeled training data for each object class, which is costly to obtain in practice. As significant portions of kitchen objects are usually occluded when in use, tracks obtained by visual object detection are expected to be highly unreliable and are therefore of limited value for motion analysis.

4.5. Reference Tracklet Statistics and RETLETS

We comparatively evaluated the impact of codebook size k and number of training samples on recognition performance with *Absolute Tracklets*, *Reference Tracklet Statistics*, and *RETLETS*. As shown in Fig. 8, *RETLETS* significantly outperformed *Absolute Tracklets* and *Reference Tracklet Statistics*. Codebook size had less effect on recognition performance for *RETLETS* compared to *Absolute Tracklets*, and *RETLETS* with seven histograms of size $k = 100$ strongly outperformed *Absolute Tracklets* with equal feature dimensionality ($k = 700$). As with *RETLETS* each tracklet contributes to one bin in each reference object's histogram, these results support the hypothesis that *RETLETS* encode local motion using multiple complementary descriptors efficiently. Performance with *Absolute Tracklets* saturated at a codebook size of about $k = 500$, which is substantially smaller than $k = 2000$ as used in Wang et al. (2011). While larger codebooks better capture fine-grained nuances, a larger number of samples (tracklets) is required for a robust statistical estimate of the probability density function histograms approximate. We expect larger codebooks to be beneficial on longer temporal windows of video data with higher spatial resolution. Performance also saturated at about 10k training samples, corresponding to an expected overlap of $\sim$ 75% between temporal windows. Table 2 shows that *RETLETS* (bottom) considerably outperformed *Absolute Tracklets* (19% increase), HOG, HOF, MBH and their combination, and performed comparably to ECDF and *Accelerometer Statistics*.

The impact of applying spatial re-weighting to *RETLETS* on recognition performance was evaluated by constructing RETLETS with codebook size $k = 100$ and varied spatial weighting parameter σ. Average precision and average recall are plotted in Fig. 9. Performance rose sharply from $\sigma = 30$ to $\sigma = 360$. From that point onwards recognition performance was relatively unaffected, falling a little. A possible explanation is that tracklets that were very close to the reference tracklet were likely to exhibit motion similar to the reference tracklet. A relative description of such motion is therefore uninformative. At the other extreme, tracklets that were very far away from the reference tracklet were less likely to interact with the reference object, justifying an intermediate spatial weighting of $\sigma = 360$ used here to discount the contribution of far away tracklets.

4.6. Feature concatenation

This section investigates recognition performance using concatenations of various feature types. Table 2 (bottom) shows recognition results obtained by concatenating visual motion features with *Accelerometer Statistics* (baseline motion), all visual features with *Accelerometer Statistics* (baseline fusion), and features used for baseline fusion with *RETLETS* and *Reference Tracklet Statistics*. Concatenations of features from accelerometers and video consistently showed a significant performance increase compared to features from individual modalities. Concatenating features extracted from both sensor modalities independently (baseline fusion) showed a performance increase of 8% and 12% compared to accelerometer features and visual features, respectively. The best performance

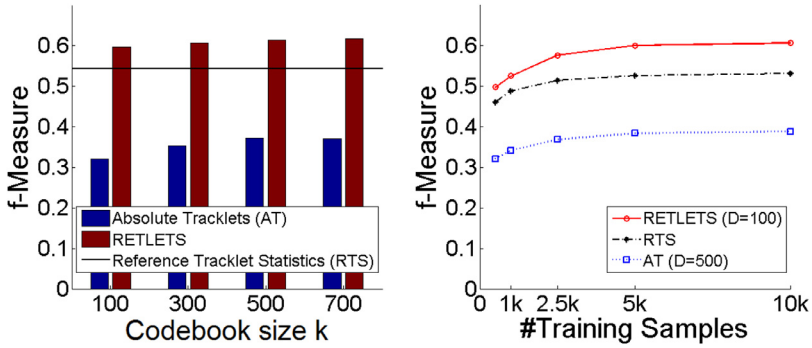

Fig. 8. Recognition results using Reference Tracklet Statistics, Absolute Tracklets and RETLETS with variation in codebook size and number of training samples.

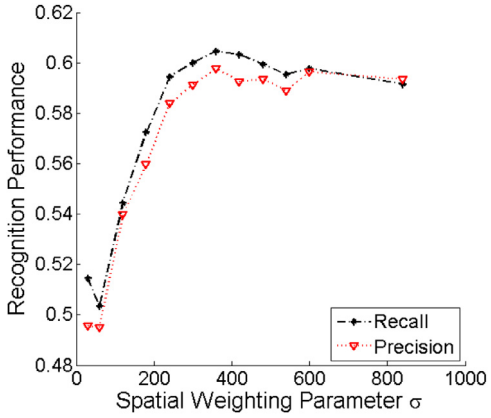

Fig. 9. Average precision and average recall using RETLETS with varied spatial weighting parameter σ.

was achieved by concatenating baseline vision features, *Accelerometer Statistics*, *Reference Tracklet Statistics* and *RETLETS*. Our approach of fusing information from video and accelerometers clearly outperformed the concatenation of features extracted from each sensor type independently (baseline motion and baseline fusion in Table 2) by 5% and 6%, respectively.

We would argue that the cost of extracting *Accelerometer Statistics* is negligible and the additional cost of extracting HOG, HOF, MBH and *Reference Tracklet Statistics* is small compared to the cost of extracting RETLETS. Therefore, there is little to be gained by not extracting these features in addition to RETLETS. The computational time complexities of accelerometer tracking and RETLET encoding in each frame are $O(MN)$ and $O(NMk)$, respectively, where M is the number of tracked points, N is the number of reference objects, and k is the size of the codebook. In practice, the computation time required for accelerometer tracking and RETLET encoding is relatively small compared to estimation of dense optical flow.

The results presented in this section make a strong case for combining vision with accelerometers for activity recognition and confirm our hypothesis that careful combination of data from these sensors can significantly improve activity recognition performance: the proposed method outperformed visual features by 18% and accelerometer features by 14%.

5. Conclusion & future work

We developed a relational histogram model that encodes relations between local visual descriptors and properties of a small fixed number of tracked objects, where quantized relations are learned using bag-of-words. By distinguishing between generic features and features from reference objects, this model facilitates development of hybrids between generic and object-based recognition models. We proposed one such hybrid model in this paper using accelerometer tracking and RETLETS to capture interactions between accelerometer-equipped objects and visual entities. We presented an accelerometer localization algorithm that outperforms previous methods and extended it to enable long-term tracking across multiple episodes in which accelerometer-equipped objects are used. We thus proposed a novel approach to multi-modal activity recognition combining information from video and accelerometer data through relative motion descriptors.

RETLETS showed considerably better activity recognition performance compared to dense tracklets, HOG, HOF, MBH, and their combination on the 50 Salads dataset. Our comparative evaluation of features from accelerometers and video highlighted a performance gap between visual and accelerometer-based motion features and showed a substantial performance gain when combining features from these sensor modalities. A considerable further performance gain was observed in combination with RETLETS and Reference Tracklet Statistics as proposed in this paper. These results justify a multi-modal approach and indicate the importance of developing methods for effective modality fusion.

For future work, evaluating the proposed method in different scenarios such as surgery, assembly tasks, repair tasks, sports and social interactions would be desirable to further support the effectiveness of our method. Currently, there is a strong unmet need for multi-modal activity recognition datasets. This is partly due to the substantial effort necessary for careful planning, data acquisition and annotation.

The 50 Salads dataset has richer annotation than used here. Specifically, activities were split into preparation, core and post-phases, and these phases were annotated as temporal intervals. Each activity annotation also includes the ingredient acted upon (e.g., *cut tomato into pieces*) and is associated hierarchically with more broadly defined activities. These detailed annotations may be used in future work to investigate the main sources of confusion errors between activities and for evaluating methods that simultaneously reason about motion and objects acted upon (Aksoy et al., 2011; Yang et al., 2013) and hierarchical activities (Summers-Stay et al., 2012).

There is potential for improvement in accelerometer localization accuracy through, e.g., probabilistic formulations such as particle filtering methods and explicit pose estimation. We expect that more reliable estimation of accelerometer trajectories would translate into higher recognition performance using RETLETS.

Occasionally, activities of interest are performed (partially) outside the camera view. While visual recognition may fail in these instances, features extracted from accelerometer data capture useful information if at least one accelerometer-equipped object is in use. A set of conditional representations and a method for opportunistic switching depending on visibility could help in this situation.

In this paper, reference tracklets were determined by localizing accelerometer-equipped objects. However, the proposed feature representation can be used with reference tracklets obtained in any way. These could, for example, be prominent point trajectories, or trajectories of visually tracked objects. Future work could evaluate RETLETS with reference tracklets from other sources such as visual object tracking. Accelerometer tracking could provide a useful method for bootstrapping visual object tracker training with the local neighborhood around localised accelerometer-equipped objects serving as noisy object region annotations. It would also be useful to extend this model to incorporate uncertainty about localization of tracked objects.

Acknowledgements

The authors would like to thank Jianguo Zhang and Ruixuan Wang for valuable feedback on drafts of this paper. This research was funded by RCUK grants EP/G066019/1 and EP/K037293/1.

References

Aksoy, E.E., Abramov, A., Dörr, J., Ning, K., Dellen, B., Wörgötter, F., 2011. Learning the semantics of object-action relations by observation. Int. J. Rob. Res. 30 (10), 1229–1249.
Albanese, M., Chellappa, R., Cuntoor, N., Moscato, V., Picariello, A., Subrahmanian, V.S., Udrea, O., 2010. PADS: a probabilistic activity detection framework for video data. IEEE Trans. Pattern Anal. Mach. Intell. 32 (12), 2246–2261.
Behera, A., Hogg, D.C., Cohn, A.G., 2012. Egocentric activity monitoring and recovery. In: Asian Conference on Computer Vision.
Bilinski, P., Corvee, E., an F. Bremond, S.B., 2013. Relative dense tracklets for human action recognition. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition.
Bouguet, J.-Y., 1999. Pyramidal implementation of the Lucas Kanade feature tracker. In: Proceedings of USENIX Annual Technical Conference.
Brown, D.C., 1966. Decentering distortion of lenses. Photometric Eng. 32 (3), 444–462.
Chen, C., Jafari, R., Kehtarnavaz, N., 2015. A survey of depth and inertial sensor fusion for human action recognition. Multimedia Tools Appl. 1–21.
Farnebäck, G., 2003. Two-frame motion estimation based on polynomial expansion. In: Proceedings of Scandinavian Conference on Image Analysis, pp. 363–370.
Fathi, A., Farhadi, A., Rehg, J.M., 2011a. Understanding egocentric activities. In: Proceedings of International Conference on Computer Vision, pp. 407–414.
Fathi, A., Ren, X., Rehg, J.M., 2011b. Learning to recognize objects in egocentric activities. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, Colorado, USA, pp. 3281–3288.
Figo, D., Diniz, P.C., Ferreira, D.R., 2010. Preprocessing techniques for context recognition from accelerometer data. Pers. Ubiquitous Comput. 14 (7), 645–662.
Gupta, A., Kembhavi, A., Davis, L.S., 2009. Observing human-object interactions: using spatial and functional compatibility for recognition. IEEE Trans. Pattern Anal. Mach. Intell. 31 (10), 1775–1789.
Hammerla, N., Kirkham, R., Andras, P., Plotz, T., 2013. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In: Proceedings of International Symposium on Wearable Computers, pp. 65–68.
Henderson, S., Feiner, S., 2011. Exploring the benefits of augmented reality documentation for maintenance and repair. IEEE Trans. Vis. Comput. Graph. 17 (10), 1355–1368.
Hoey, J., Ploetz, T., Jackson, D., Monk, A., Pham, C., Oliver, P., 2010a. Rapid specification and automated generation of prompting systems to assist people with dementia. Pervasive Mobile Comput. 7 (3), 299–318.
Hoey, J., Poupart, P., v. Bertoldi, A., Craig, T., Boutilier, C., Mihailidis, A., 2010b. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. Comput. Vis. Image Understand. 114 (5), 503–519.
Hsu, C.-H., Yu, C.-H., 2009. An accelerometer based approach for indoor localization. In: Proceedings of Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, Washington, DC, USA, pp. 223–227.
Huynh, T., Fritz, M., Schiele, B., 2008. Discovery of activity patterns using topic models. In: Proceedings of International Conference on Ubiquitous Computing.
Laptev, I., 2005. On space-time interest points. Int. J. Comput. Vis. 64 (2/3), 107–123.
Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B., 2008. Learning realistic human actions from movies. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.
Laptev, I., Prez, P., 2007. Retrieving actions in movies. In: Proceedings of IEEE Conference on Computer Vision.
Lei, J., Ren, X., Fox, D., 2012. Fine-grained kitchen activity recognition using RGB-D. In: Proceedings of International Conference on Ubiquitous Computing, pp. 208–211.
Liu, J., Luo, J., Shah, M., 2009. Recognizing realistic actions from videos “in the wild”. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.
Maki, Y., Kagami, S., Hashimoto, K., 2010. Accelerometer detection in a camera view based on feature point tracking. In: Proceedings of IEEE/SICE International Symposium on System Integration.
Marszałek, M., Laptev, I., Schmid, C., 2009. Actions in context. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.
Matikainen, P., Hebert, M., Sukthankar, R., 2009. Trajectons: action recognition through the motion analysis of tracked features. In: Proceedings of International Conference on Computer Vision.
Messing, R., Pal, C., Kautz, H., 2009. Activity recognition using the velocity histories of tracked keypoints. In: Proceedings of International Conference on Computer Vision.
Pham, C., Oliver, P., 2009. Slice&Dice: recognizing food preparation activities using embedded accelerometers. Ambient Intell. LNCS 5859, 34–43.
Plötz, T., Hammerla, N.Y., Olivier, P., 2012. Feature learning for activity recognition in ubiquitous computing. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 1729–1734.
Rao, C., Yilmaz, A., Shah, M., 2002. View-invariant representation and recognition of actions. Int. J. Comput. Vis. 50 (2), 203–226.
Rhienmora, P., Haddawy, P., Suebnukarn, S., Dailey, M.N., 2009. Intelligent dental training simulator with objective skill assessment and feedback. Artif. Intell. Med. 52 (2), 115–121.
Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., Doppler, J., Holzmann, C., Kurz, M., Holl, G., Chavarriaga, R., Creatura, M., del R. Milln, J., 2010. Collecting complex activity data sets in highly rich networked sensor environments. In: Proceedings of International Conference on Networked Sensing Systems.
Rohrbach, M., Rohrbach, A., Regneri, M., Amin, S., Andriluka, M., Pinkal, M., Schiele, B., 2015. Recognizing fine-grained and composite activities using hand–centric features and script data. Int. J. Comput. Vis. 1–28.
Ryoo, M.S., Aggarwal, J.K., 2007. Hierarchical recognition of human activities interacting with objects. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition.
Ryoo, M.S., Aggarwal, J.K., 2009. Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In: Proceedings of IEEE Conference on Computer Vision.
Schuldt, C., Laptev, I., Caputo, B., 2004. Recognizing human actions: a local SVM approach. In: Proceedings of International Conference on Pattern Recognition.
Shigeta, O., Kagami, S., Hashimoto, K., 2008. Identifying a moving object with an accelerometer in a camera view. In: Proceedings IEEE International. Conference on Intelligent Robots and Systems.
Stein, S., McKenna, S.J., 2012. Accelerometer localization in the view of a stationary camera. In: Proceedings of Conference on Computer and Robot Vision, pp. 109–116.
Stein, S., McKenna, S.J., 2013. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of ACM International Joint Conference on Pervasive and Ubiquitous Computing.
Summers-Stay, D., Teo, C.L., Yang, Y., Fermüller, C., Aloimonos, Y., 2012. Using a minimal action grammar for activity understanding in the real world. In: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamourna, Portugal, pp. 4104–4111.
Teixeira, T., Jung, D., Savvides, A., 2010. Tasking networked CCTV cameras and mobile phones to identify and localize multiple people. In: Proceedings ACM International Conference on Ubiquitous computing.
Tenorth, M., Bandouch, J., Beetz, M., 2009. The TUM kitchen data set of everyday manipulation activities for motion tracking and action recognition. IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences.
Tomasi, C., Kanade, T., 1991. Detection and tracking of point features. Carnegie Mellon University Technical Report CMU-CS-91-132, April.
de la Torre, F., Hodgins, J., Montano, J., Valcarcel, S., Forcada, R., Macey, J., 2009. Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database. Technical Report. Robotics Institute, Carnegie Mellon University.
Wang, H., Klaser, A., Schmid, C., Liu, C.-L., 2011. Action recognition by dense trajectories. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition.
Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C., 2009. Evaluation of local spatio-temporal features for action recognition. In: Proceedings of British Machine Vision Conference, pp. 124.1–124.11.
Wilson, A.D., Benko, H., 2014. Crossmotion: fusing device and image motion for user identification, tracking and device association. In: Proceedings of International Conference on Multimodal Interaction.
Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M., 2007. A scalable approach to activity recognition based on object use. In: Proceedings of IEEE International Conference on Computer Vision, pp. 1–8.
Yang, Y., Fermüller, C., Aloimonos, Y., 2013. Detection of manipulation action consequences (MAC). In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, Oregon, USA, pp. 2563–2570.
Zappi, P., Lombriser, C., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., Tröster, G., 2008. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In: Proceedings of European Conference on Wireless Sensor Networks.
Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C., 2007. Local features and kernels for classification of texture and object categories: a comprehensive study. Int. J. Comput. Vis. 73 (2), 213–238.
Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. 22 (11), 1330–1334.