Fine-Grained Kitchen Activity Recognition using RGB-D

Jinna Lei
University of Washington
jinna@cs.washington.edu

Xiaofeng Ren
Intel Labs
xiaofeng.ren@intel.com

Dieter Fox
University of Washington
fox@cs.washington.edu

ABSTRACT

We present a first study of using RGB-D (Kinect-style) cameras for fine-grained recognition of kitchen activities. Our prototype system combines depth (shape) and color (appearance) to solve a number of perception problems crucial for smart space applications: locating hands, identifying objects and their functionalities, recognizing actions and tracking object state changes through actions. Our proof-of-concept results demonstrate great potentials of RGB-D perception: without need for instrumentation, our system can robustly track and accurately recognize detailed steps through cooking activities, for instance how many spoons of sugar are in a cake mix, or how long it has been mixing. A robust RGB-D based solution to fine-grained activity recognition in real-world conditions will bring the intelligence of pervasive and interactive systems to the next level.

Author Keywords

Smart Spaces, Kitchen, Activity Tracking, Object Recognition, Action Recognition, RGB-D

ACM Classification Keywords

H.5.2 Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Future pervasive systems, if they are to seamlessly monitor and assist people in their daily activities, must have the capabilities to understand human activities in *fine-grain* details. For example, during cooking, a kitchen assistant system would want to know where the ingredients are and what states they are in, what actions a person is doing, and which step in the recipe he/she is at (see Figure 1).

How could a system automatically acquire such a large variety of information? One approach is to employ instrumentation, such as using RFID tags or markers, to simplify the perception problem [2]. In comparison, a camera-based approach is generic and requires minimal changes to the environment [5]. Combining the use of cameras and RFID tags leads to more robust solutions [11]. Computer vision, however, is computationally demanding and often brittle: despite a lot of recent progress in vision-based object recognition [7, 1] and action recognition [4, 5], fine-grained understanding of complex activities such as cooking [9] is still an open challenge in unconstrained conditions.

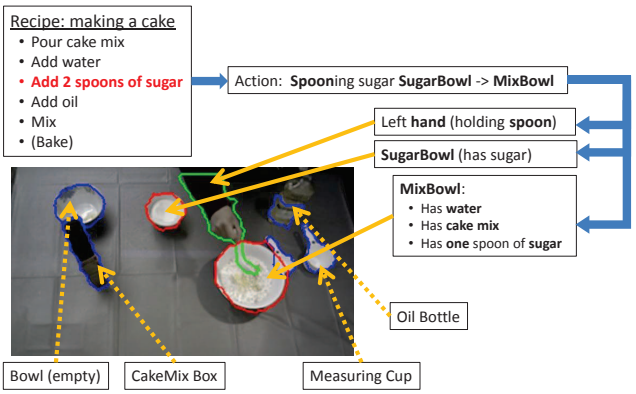

Figure 1. Fine-grained activity recognition requires identifying and tracking of objects and recognizing actions and how actions change the states of objects. For an automatic system to follow or assist a user through a recipe, the system needs to understand the spooning action, know about containers such as the SugarBowl which contains sugar, and keep track of how many spoons of sugar have been added. We demonstrate that RGB-D perception, using Kinect-style depth cameras, has great potentials in realizing fully automatic fine-grained activity recognition without instrumentation such as RFID tags or markers.

RGB-D cameras, affordable Kinect-style cameras that provide both color and depth, are changing the landscapes of vision research and applications. Using infrared projection, these cameras provide real-time 3D data that is largely independent of illumination, making visual perception much more robust (and efficient) than previously possible. RGB-D perception has greatly advanced the state of the art for many vision problems, including body pose [8], hand tracking [6], object recognition [3] and user interfaces [12].

In this work we demonstrate that RGB-D perception has great potentials for fully automatic recognition of activities at a very fine granularity. We use cooking in smart kitchens as a motivating example: we will examine objects that transform, such as flour to be mixed or vegetables to be chopped, which would be hard to instrument; we will also examine interactions between hands and objects, such as grasp/release and touch/untouch, which would be hard to detect if using a color-only camera. We build a prototype system where

UbiComp '12, Sep 5-Sep 8, 2012, Pittsburgh, USA.

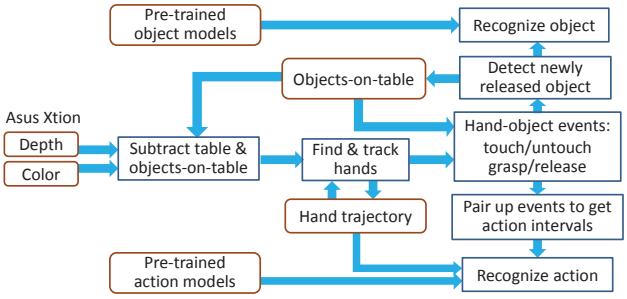

Figure 2. An overview of our approach. We use depth+color from an Asus Xtion camera to locate hands and detect events (grasp/release, touch/untouch). Objects are recognized when newly released, and tracked when on the table. Actions are classified using hand trajectories and pairs of hand events, which in turn update object states.

RGB-D data are used to solve several key problems in activity analysis: locating and tracking hands, recognizing and tracking objects, detecting hand-object interactions such as grasp and release, and recognizing actions using the intervals given by the hand events. Our system enables robust and efficient recognition of objects and actions, demonstrated through quantitative evaluations as well as examples of full-length cooking activities.

FINE-GRAINED RECOGNITION FOR SMART SPACES

In this work we build a prototype system to demonstrate that RGB-D cameras, modern depth cameras that provide synchronized color and depth information at high frame rates, can potentially enable a robust solution to fine-grained activity recognition. We consider a cooking scenario, where a user prepares food ingredients, such as mixing a cake, on a tabletop. We mount an Asus Xtion Pro Live camera over the tabletop, which provides color+depth data at 640x480 resolution and 30 fps. Figure 2 shows an overview of the algorithm steps and the data flow of our system. The main components of our system are:

1. **Hand and object tracking**, using depth to robustly track the positions of hands and objects, and detect when and where hands interact with the objects (e.g. grasp);
2. **Object and action recognition**, using both depth (shape) and color (appearance) to identify objects and to recognize the actions being performed on them.

Hand and Object Tracking

The depth channel of the Asus Xtion camera uses *active stereo* and is robust to lighting conditions (which are often poor in indoor spaces). We use depth to extract hands and objects from a tabletop and detect when hands touch and move objects. For separating hands from objects being grasped in hand, we resort to the use of skin color.

Table surface model and background subtraction. The system maintains and updates a background depth map, consisting of the expected depths of the table and all the static objects on the table. At each time instant, the background depth map (table and static objects) is subtracted from the incoming depth map, and the remainder is the active part of the scene: hands and objects held in hands. Similar to [12], active regions are classified as hands if they come from offscreen in the direction of the front of the table. Non-hand active regions are new objects that appear in the space; they are added to the system as static objects, their identities are recognized, and the background depth map is updated accordingly. The system also detects depth changes when a static object becomes active and moves.

Hand tracking. Hands are tracked frame-to-frame using the extracted hand masks. Left/right hands are differentiated by where they cross the offscreen boundary. The one-hand case is ambiguous as it could be left or right; our system retroactively labels them once it detects two hands. For more accurate hand positions, we need to separate the hands from the objects in the hands: we use a mixture-of-Gaussian model for hand color and compute connected components using double thresholds (as in Canny's edge detection). Hand color is calibrated by using a short training phase where the user shows only his/her hands. Comparing to the sophisticated Kinect-based hand tracking in [6], our task is much simpler: we only find and track the positions of the hands as a whole, which are already very useful for finding action spans and recognizing actions. The limited resolution of the hands from an overhead camera would make it difficult to track individual fingers.

Hand-object interaction. When a hand approaches an object (distance below a threshold in 3D space), the system updates the object status to be *touched*. If the distance increases, it changes to *untouched*. If an object moves (i.e. its current depth map differs enough from the previously observed model), its status is changed to *grasped*. When a new non-hand region appears, and if it is similar enough to the grasped objects (size and appearance), it is not a new object; instead it is the grasped object being *released* to the scene. These events are useful in identifying when actions start and end (see Figure 3 for an example).

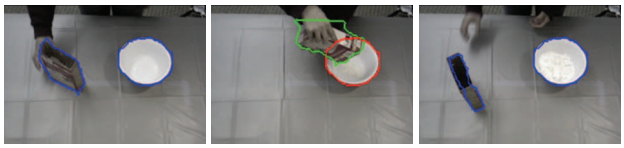

Figure 3. We use grasp and release events to determine spans of actions. (Left) Hand approaching the CakeMix box (inactive, in blue); action about to start. (Middle) Hand grasping the box and pouring on the MixBowl (active, in red). (Right) Box released on table, action ended.

Object and Action Recognition

The key to fine-grained activity analysis is the recognition of objects and their state changes through actions. We use our prototype tracking system to study both object and action recognition, in the context of kitchen activities, assuming objects are extracted from the scene and actions are localized in time through hand-object interaction events.

Scope. We consider 7 common actions: *place* (PL), putting an object into the smart space from offscreen; *move* (MV), moving an object inside the space; *chop* (CH), chopping vegetables and fruits; *mixing* (MX), stirring things and mixing them together; *pouring* (PR), pouring liquid or grain from

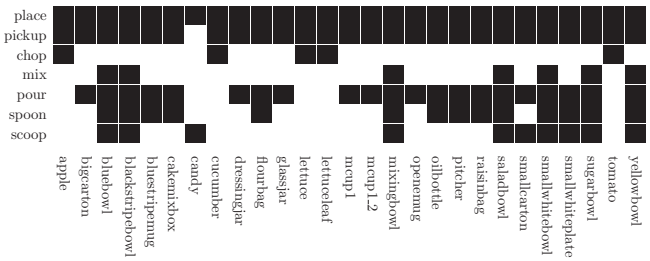

Figure 4. The action-object compatibility (compatible pairs in black). Each object has well-defined functions and can be used in a certain set of actions: bowl can be poured into, cucumber and tomato can be chopped, etc. Not all objects can be used in all actions.

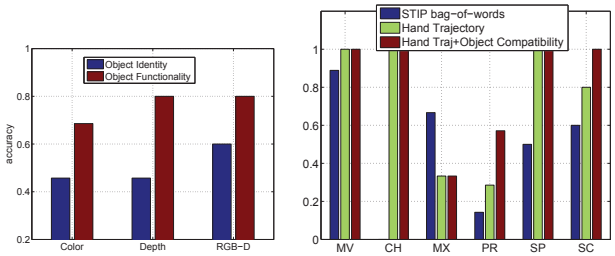

Figure 5. Object (left) and action (right) recognition results. Object recognition accuracies are in terms of both identity (which object) and functionality (what actions it can be used in), using color, depth, and color+depth. Action recognition accuracies are listed for our approach using hand trajectories (including the use of object-action compatibility), comparing to that using spatial-time interest points (STIP).

one container to another; *spooning* (SP), using a spoon to transport stuff between containers; and *scooping* (SC), moving piles of stuff using hands. We consider a set of 27 objects including bowls, mugs, boxes, bags as well as fruits (e.g. apple) and vegetables (e.g. lettuce).

One thing to note is that not all objects can appear in all actions. For example, chopping can only apply to a small number of objects such as cucumber and lettuce. On the other hand, more generic actions such as moving applies to all objects. In Figure 4 we list all the compatible pairs of actions and objects in our study. This semantic information will be used to facilitate action recognition.

Object recognition. For object recognition we apply the state-of-the-art recognition approach [1] using RGB-D kernel descriptors. We use the gradient kernel descriptors (SIFT-like) on both color and depth data with linear SVM classifiers. Note that there is no need to scan the scene for objects (which would be much harder): objects are found and cropped out using the tracking system.

Action Recognition. We use hand movements as the basis for action recognition. Our system can robustly track hands through complex activities. We use two types of features; a global feature using PCA on (the gradients of) 3D hand trajectories, and a local feature using bag-of-words of snippets of trajectory gradients (useful for locally distinctive actions such as chopping). We also use the duration of the action. Note we do not need to search over the time intervals of potential actions: they are provided by pairs of hand events such as the grasp and release of the same object.

Evaluation. To train object models, we place each object at a small number of locations (3-4) on the table and record their RGB-D appearance. For action models, we collect a small number of clean, non-cluttered training sequences (3 per action). For testing, we use a total of 10 sequences, each consisting of multiple actions. They include a total of 35 object instances, and 28 action instances. We assume the start and end points of the actions are given.

Object and action recognition results are shown in Figure 5. The object recognition accuracy for 27 objects is 60%. This is lower than reported in [1], partly because of the inherent ambiguities of some of the objects (bowls look round and white), partly the overhead view of the camera, and partly the sparsity of the training data. On the other hand, if we only require knowing the functions of the object, i.e. what actions they can be used in, the accuracy is 80%. Note that depth (shape) alone gives the same accuracy for function recognition as depth+color, showing that the functions of an object are mostly based on its shape. On the other hand, accurately recognizing object identity would require using both depth and color.

For action recognition, we compare to a state-of-the-art vision approach (RGB only), spatial-temporal interest points (STIP) [4] in a bag-of-words model. STIP does not require hand tracking (difficult to do in general) and is the basis of many recent works on action recognition. We find that the accuracy of STIP is 54%, while our trajectory-based approach, which is much simpler in nature and faster to compute, achieves a higher accuracy of 75%. This shows the advantage of explicitly tracking hands, even though hand positions are not always accurate (in particular the separation of hands from objects-in-hand using color). One interesting note is the benefit of combining trajectory-based action recognition with object recognition. By feeding object recognition results (through the compatibility matrix in Figure 4) into action recognition, where scores are linearly combined, the accuracy is further increased to 82%.

Activities and Interactive Recipes

Combining hand-object tracking and object-action recognition, our system is capable of identifying and recognizing objects and actions in real-world activities. We demonstrate this capability on a cake baking video which consists of 7 objects and 17 actions over 6000 frames. A number of semantic constraints are incorporated into recognition as heuristics: there are no chopping or scooping actions; there is a long mixing action; and the SugarBowl contains sugar.

Figure 6 summarizes our results. All the actions are correctly recognized (except for a *move* action on the MixBowl, which is combined with mixing in the computed result because its start point is not detected). Most of action start and end events are accurately identified in time: over 90% of the endpoints are detected within a margin of $1/3$ second. As a result, our system keeps track of object states: it knows whether and when oil or sugar is added to the MixBowl, or how long it has been mixing.

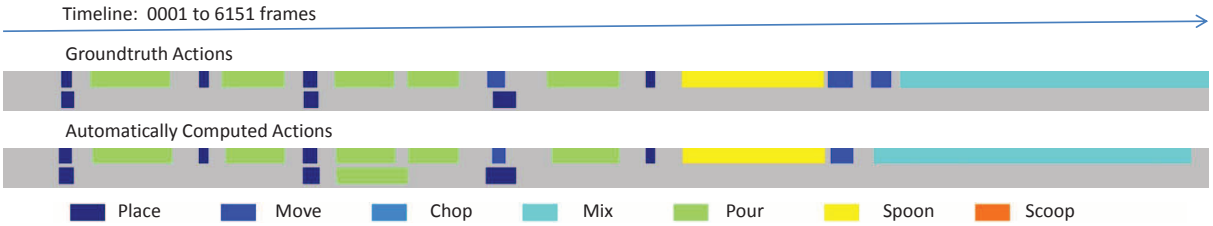

Figure 6. A timeline visualization of our action recognition results for a full-length activity (making a simple cake). The first row is the groundtruth and the second is our automatic recognition. Each row contains two lines, as two actions may happen at the same time. The timeline is shortened where no action takes place and for the mixing action in the end. Our system works very well both in tracking objects and determining the start and end of actions; it also does action recognition almost perfectly. The parsed video is included as supplemental material.

DISCUSSION

A robust RGB-D solution to fine-grained activity recognition can have substantial impact on future intelligent systems. Figure 7 illustrates what our system can enable in the kitchen domain: with fine-grained recognition, an intelligent kitchen might highlight next steps, notify of a misstep, or guide the timing of actions. Fine-grained activity recognition can enable previously Wizard-of-Oz systems such as the Cook's Collage [10] to be fully autonomous, or be used in many other scenarios such as interactive play, workbench operations, personal health, or elder care.

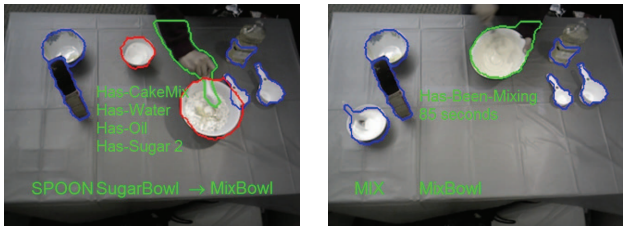

Figure 7. Automatic fine-grained activity recognition is crucial for enabling intelligent environments. Recognizing actions and tracking the states of objects leads to interesting scenarios. (Left) Our system knows the ingredients the MixBowl currently has. Adding more sugar could trigger an alarm if it is against the recipe. (Right) Has the user done enough mixing? The system knows the length of the mixing action, and it can potentially recognize the appearance of a well-mixed pile.

It will undoubtedly be a long journey toward realizing this goal in a real deployed system. Our study is only a proof of concept and is limited in its scope: a larger set of objects and actions, along with variations across people and physical environments, will present many challenges not revealed in our work. On the other hand, many of these challenges are high-level ones that any approach will have to face, regardless of its sensor choices. Another limitation of our current system is that it performs offline analysis; an interactive system will require more complex online inference. There are many possible ways to reduce the complexity of the problem, such as enforcing constraints in a recipe, adapting to individual users, or combining multiple sensors including wearable cameras and infrastructure sensors. We have sufficient evidence that low-level hand and object tracking can be made robust using RGB-D cameras in the real world, and this will make a fundamental difference when a system needs to pick up subtle details in human activities under unconstrained conditions.

Acknowledgment

This work was funded by the Intel Science and Technology Center for Pervasive Computing (ISTC-PC).

REFERENCES

1. L. Bo, X. Ren, and D. Fox. Depth Kernel Descriptors for Object Recognition. In *IROS*, pages 821–826, 2011.
2. M. Buettner, R. Prasad, M. Philipose, and D. Wetherall. Recognizing daily activities with RFID-based sensors. In *Ubicomp*, pages 51–60, 2009.
3. K. Lai, L. Bo, X. Ren, and D. Fox. A scalable tree-based approach for joint object and pose recognition. In *AAAI*, 2011.
4. I. Laptev. On space-time interest points. *Int'l. J. Comp. Vision*, 64(2):107–123, 2005.
5. R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, pages 104–111. IEEE, 2009.
6. I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, 2011.
7. X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, pages 3137–3144. IEEE, 2010.
8. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, volume 2, page 3, 2011.
9. E. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *First Workshop on Egocentric Vision*, 2009.
10. Q. Tran, G. Calcaterra, and E. Mynatt. Cook's collage. *Home-Oriented Informatics and Telematics*, 2005.
11. J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg. A scalable approach to activity recognition based on object use. In *ICCV*, pages 1–8, 2007.
12. R. Ziola, S. Grampurohit, N. Landes, J. Fogarty, and B. Harrison. Examining interaction with general-purpose object recognition in LEGO OASIS. In *Visual Languages and Human-Centric Computing*, pages 65–68, 2011.